

Nagravision SA
IMS- SI
Generic download procedure
Specifications
V 1.2.0

This document contains confidential and privileged information.
The reproduction of any part of the document is strictly prohibited without the prior
written consent of Kudelski SA.

Table Of Contents

Introduction	3
1.1. Purpose.....	3
1.2. Intended Audience	3
1.3. Trademarks.....	3
1.4. Acronyms and Abbreviations.....	3
1.5. References.....	3
Overview	4
1.7. Broadcasting information.....	4
1.8. Notifying the STB.....	4
1.9. Checking the software.....	4
1.10. Finding the software.....	4
Specifications	5
1.11. Nagra_download_descriptor.....	5
1.12. Download service	7
1.13. In the PAT	7
1.14. In the PMT.....	7
Examples	8
1.15. Example 1.....	8
1.16. Example 2.....	8
1.17. Example 3.....	8
1.18. Example 4.....	9
1.19. Example 5.....	9

List Of Tables

Table 1 - Acronyms	3
Table 2 – Nagra_download_descriptor.....	5
Table 3 – Hardware_type definitions	6
Table 4 – Hardware_type definitions	6
Table 5 – Usage_type definition.....	6
Table 6 – Object_type definition.....	6
Table 7 – Download_type definition	7

Introduction

1.1. Purpose

The purpose of this document is to describe a generic way of downloading a Set Top Box with the Nagravision signalization.

1.2. Intended Audience

This document is intended for staff with privileges according to the security level of the present document.

Creation Date: April 3, 2001
Last Modification Date: April 3, 2001
Print Date: July 15, 2003
Reference: ImsDwnSpe010200.Pub.doc
Author(s): Christophe Korber
Approval(s):

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 Abbreviations

Acronym Abbreviation	Definition
ATSC	Advanced Television Systems Committee
DVB	Digital Video Broadcasting
DSM-CC	Digital Storage Media - Command & Control
EPG	Electronic Program Guide
IMS	Information Management System
MPEG	Moving Pictures Expert Group
PSI	Program Specific Information
SI	Service Information
STB	Set Top Box

Table 1 - Acronyms

1.5. References

- [1] ISO/IEC 13818-1: Information technology - Generic coding of moving pictures and associated audio information – part 1: Systems. November 13, 1994.
- [2] ETS 300-468 v1.3.1: Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems. February 1998.
- [3] ETR-211: Digital Video Broadcasting (DVB); Guidelines on implementation an usage of DVB service information. Final Draft, May 23, 1996.
- [4] ETR-162: Allocation of Service Information (SI) codes for Digital Video Broadcasting (DVB) systems; Working draft, 26th March 1999
- [5] NAGRAVISION, DVB / SI descriptors V2.3, No date.

Overview

In order to be able to change the software contained in Set Top Boxes, some procedures must be implemented. Currently each manufacturer has his own way of downloading STBs, sometime more than one method. But all methods are quite the same with the same kind of information broadcast. The idea is to create a generic way of downloading new code in STBs for manufacturer who haven't already defined.

Different phases are needed to implement and use a new download:

- Broadcasting information of the actual versions of software available on air
- Eventually notifying a STB or a group of STBs that new software is available for them.
- The STB has to check if the software is relevant for itself.
- Finding the correct data in the network and downloading it.

1.7. Broadcasting information

To broadcast information of what is on air, the SI are used (independently of the standard used). The information needed on air is:

- Identifications of the release of software on the air (manufacturer, version of software, kind of software). A linkage_descriptor with a private linkage_type and private data is used for that.
- Information on the way that the software is broadcast (PIDs...). This information is included in one or more PMT.
- Information to connect all these things together. This information is dependant of the standard used but, for DVB the NIT contains the linkage_descriptor that points to the service, (the service contains eventually the CA information for the download if it is scrambled). And that value of service is used to identify the PMT.

1.8. Notifying the STB

It's possible to notify the STB that a download has to be done. This is done by an IRD command (EMMs that could trigger some behavior in the STB). This is used mainly to force the download immediately. The description of this command is not in the scope of this document.

1.9. Checking the software

A linkage descriptor located in the NIT (or the network piece of information for a non-DVB system) contains a generic description of the software. The STB has to check all version of on air software to identify the correct one. Some information such as manufacturer, version of software, version of hardware supported, are presented to identify uniquely all versions of software present. This linkage descriptor with a private type is called nagra_download_descriptor further in this document.

1.10. Finding the software

When the correct software is identified, the linkage descriptor points on a service uniquely identified by its service_id (= program_number in MPEG-2), its transport_stream_id and its original_network_id. Then the STB checks in the SDT (for DVB) if the customer has the right to download the new software, then the PMT for that program is read and the PID is extracted. Then the STB is able to download the software.

Specifications

In this chapter you will find the information needed for that download. The description is done according to the DVB standard. If another standard is used, the exact mapping of the DVB fields has to be done.

1.11. Nagra_download_descriptor

In the DVB standard, the nagra_download_descriptor is located in the first loop of the NIT. This descriptor is in fact a linkage_descriptor with a private linkage_type. For that reason, this descriptor is preceded by a private_data_specifier_descriptor with the Nagra private_data_specifier value of 'NAG\0' (This value is to be confirmed). Be careful, this descriptor could be extended in the future. To avoid any problem in the future, check the loop_length field to jump from one loop to the other, even if you read all the actual fields. The private_data field is reserved for the Nagra use only. This descriptor points to the service that contains the broadcast software.

The definition of this descriptor is the following:

Syntax	Nb. of bits	Identifier
Nagra_download_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
service_id	16	uimsbf
linkage_type	8	uimsbf
for (I=1 ; I<N ; I++)		
{		
loop_length	8	uimsbf
manufacturer_id	32	uimsbf
hardware_type	32	uimsbf
hardware_version	32	uimsbf
usage_type	8	uimsbf
object_type	8	uimsbf
object_version_major	32	uimsbf
object_version_minor	32	uimsbf
download_type	8	uimsbf
component_tag	8	uimsbf
object_id	8	uimsbf
for (I=0;I<N;I++)		
{		
private_data	8	uimsbf
}		
}		
}		

Table 2 – Nagra_download_descriptor

The first fields of the nagra_download_descriptor are defined according to the DVB standards for the linkage_descriptor. The descriptor contains at least one loop to be sure to identify the STB manufacturer.

Linkage_type: This is a 8-bit field specifying the type of linkage. For this specific usage, the linkage_type is 0xD0. This linkage_type is a private Nagra value.

Loop_length: This is a 8-bit field specifying the length of the following data included the private_data loop. The private_data field is used to extend the descriptor in the future, then the loop_length value will be used

to identify the version of the linkage descriptor. Take care to check the length of the descriptor to discard all the new fields of the future extension of this descriptor.

Manufacturer_id: This is a 32-bit field identifying the manufacturer who broadcast the current software. This field must be unique within all the Nagra system. For that reason, this value is defined by Nagra. If the manufacturer has a private_data_specifier (allocated by DVB), this value will be used (if available).

Manufacturer_id	STB manufacturer
0x00000000	Undefined
0x4D445320 ('MDS ')	Mentor
0xFFFFFFFF00 – 0xFFFFFFFF	Temporary used

Table 3 – Hardware_type definitions

Hardware_type: This is an 32-bit field specifying the hardware.

Hardware_type	Description
0x00000000	Undefined or not used
0x00000001	Satellite
0x00000002	Cable
0x00000003	Terrestrial
0x00000004 – 0x0000000F	Reserved for future use
0x00000010 – 0xFFFFFFFF	Manufacturer defined

Table 4 – Hardware_type definitions

Hardware_version: This is an 32-bit field specifying the current version of hardware_type.

Usage_type: This is an 8-bit field specifying a group of STB unidentified for a specific usage.

Usage_type	Description
0x00	Undefined or not used
0x01	Development
0x02	Alpha
0x03	Beta
0x04	Production
0x05 – 0x0F	Reserved for future use
0x10 – 0xFF	Manufacturer defined

Table 5 – Usage_type definition

Object_type: This is an 8-bit field specifying the type of software.

Object_type	Description
0x00	Undefined or not used
0x01	Loader
0x02	Firmware
0x03	Application
0x04	Data
0x05 – 0x0F	Reserved for future use
0x10 – 0xFF	Manufacturer defined

Table 6 – Object_type definition

Object_version_major: This is an 32-bit field specifying the major version number.

Object_version_minor: This is an 32-bit field specifying the minor version number.

Download_type: This is an 8-bit field specifying the way to download the new software.

Download_type	Description
---------------	-------------

Download_type	Description
0x00	Undefined or not used
0x01	Immediate
0x02	After power on
0x03	After reboot
0x04	According to object versions
0x05	Force the download
0x05 – 0x0F	Reserved for future use
0x10 – 0xFF	Manufacturer defined

Table 7 – Download_type definition

Component_tag: This is an 8-bit field specifying the component where the software is broadcast. This component is a part of the service pointed by the nagra_download_descriptor. If the value is set to 0, no stream_identifier_descriptor appear in the PMT. The value is defined by the user.

Object_id: This is an 8-bit field specifying this object in the stream. In the MPEG-2 standard, this field may contain the table_id of the MPEG-2 private section used to carry the object.

Private_data: This succession of 8-bit fields specifies the future extension of this descriptor. This is reserved for the use of Nagra only. To avoid any problem in the future, check the loop_length field to jump from one loop to the other, even if you read all the actual fields. For backward compatibility, the actual fields will never be removed from this descriptor during future updates.

1.12. Download service

The download service is a service with a private type that contains only the mandatory information for a service. Usually this kind of service is not scrambled. If some STB support that kind of behavior, all the Nagra CA information related to the service will be added to that service (namely the NASP_CA_descriptor).

This service is usually not displayed in the guide. The type of this private service is 0xD0.

1.13. In the PAT

This service is listed in the PAT as all the other programs. The program number of the download service equal its service ID

1.14. In the PMT

A PMT for the download service is created. The program contains one or more streams of type 0x05. The stream_identifier_descriptor is added with the value identified in the nagra_download_descriptor. If the service is scrambled, all circuits of the service are scrambled together.

Examples

Each of the following examples are introducing one more concept (identified by a field in the `nagra_download_descriptor`). In the following examples the STB manufacturer has a **manufacturer_id** assigned to 'TST\0' (= 0x54535400). Nagra assigns this value; if the manufacturer has a DVB private_data_specifier, it will be used.

1.15. Example 1

In this example, we have the STB manufacturer 'TST\0' have one kind of satellite boxes on a network. The actual version of software is 2.4 (**object_version_major** = 0x00000002 and **object_version_minor** = 0x00000004).

The software must be updated to 2.5 to fix a minor bug:

- 1) A file containing an image of the new software is created and packetized in MPEG-2 packets
- 2) A new download service is created in the stream
- 3) A `nagra_linkage_descriptor` is created with the **manufacturer_id** = 'TST\0', the **object_version_major** = 0x00000002 and **object_version_minor** = 0x00000005, all other parameters are set to 0x00. This descriptor appears in the NIT. Because of these changes, the version of the NIT is upgraded.
- 4) The STB (that runs the software 2.4) detects the new version of the NIT and acquires the `nagra_download_descriptor`. Now, the STB knows that new software exists and behaves in an hardcoded way. For example, because of it's a minor release, the STB hasn't download the soft since the next reboot of the box.
- 5) At the next reboot of the box, it will detect the new version of software, and because we are in a boot phase, it will download the soft. Now the STB have the soft version 2.5.
- 6) At the next reboot, the version of software on air will be the same as the actual one, the STB won't download it again.
- 7) New STBs are integrated to the network with the same hardware, but the last release of software (version 2.6). If an older version of the software is detected when the new STB boots, then it does nothing. The version of software stays at 2.6.

1.16. Example 2

Now, we decided to upgrade the box with software that manages the way of downloading the new soft (field **download_type**). The new version of software will be the 3.0. This is a major release. The actual hardcoded way of downloading the new software is to download it immediately when available.

- 1) Remove the old `nagra_download_descriptor`
- 2) Create a new one with the **object_version_major** = 0x00000003 and **object_version_minor** = 0x00000000
- 3) All the boxes with software version 2.5 and 2.6 detect a new major release of the software, the download the new software immediately. The new version of software is actually 3.0

Now the STB manufacturer is able to define if a new software must be downloaded and when. If the STB manufacturer wants to keep the actual way to download new software, he will use the `download_type` = 0x04.

1.17. Example 3

Now, the STB manufacturer wants to introduce a new version of satellite STB in the network. For that he has to identify the actual one with the **hardware_version** = 0x00000001. Then, the next version of software to download is the 3.1 and contains that information. The new software must be downloaded immediately in all the STB to be sure that the old hardware is identified. The following `nagra_download_descriptor` will be generated with the following values:

- **Manufacturer_id** = 'TST\0'
- **Object_version_major** = 0x00000003
- **Object_version_minor** = 0x00000001
- **Download_type** = 0x05 (force a download without checking the versions)

Now all the boxes in the network run the version 3.1 of the software.

1.18. Example 4

The new STB (with the **hardware_version** = 0x00000002) is running three software:

- The Loader, that manages the download of new software
- The Firmware, that manages the box at the hardware level
- The Application, that is the user interface of the STB (version 1.1).

If the STB manufacturer wants to change the user's interface of the box, he must update the Application software of the new box and the full software of the old one. It's now mandatory to download the new software immediately, then he decides that the new software will be downloaded next time that the customer switches on its box. He has to broadcast to two versions of the software on the same PID, then create a **nagra_download_descriptor** with two loops. The first one with the following values for the new STB:

- **Manufacturer_id** = 'TST\0'
- **Hardware_type** = 0x00000001 (satellite boxes)
- **Hardware_version** = 0x00000002
- **Object_type** = 0x03 (Application)
- **Object_version_major** = 0x00000002
- **Object_version_minor** = 0x00000000
- **Download_type** = 0x02

For the old hardware, the download is not possible when the box is switched on, the download will be 'according to the software version', and the linkage descriptor second loop will be:

- **Manufacturer_id** = 'TST\0'
- **Hardware_type** = 0x00000001 (satellite boxes)
- **Hardware_version** = 0x00000001
- **Object_type** = 0x00 (not used)
- **Object_version_major** = 0x00000004
- **Object_version_minor** = 0x00000000
- **Download_type** = 0x04

An other solution for the manufacturer is to broadcast the two software on two different PIDs; then, he has to define the **component_tag** for each **nagra_download_descriptor**. For example: **component_tag** = 0x0a for the new hardware and **component_tag** = 0x0b for the old one. The result will be that a **stream_identifier_descriptor** will be added on the two streams defined in the PMT for the download service. The STB has to check for the stream with the correct **stream_identifier_descriptor** to know the PID where the correct software is broadcast.

For the new box the **nagra_download_descriptor** will look like:

- **Manufacturer_id** = 'TST\0'
- **Hardware_type** = 0x00000001 (satellite boxes)
- **Hardware_version** = 0x00000002
- **Object_type** = 0x03 (Application)
- **Object_version_major** = 0x00000002
- **Object_version_minor** = 0x00000000
- **Download_type** = 0x02
- **Component_tag** = 0x0a

Instead of managing the **component_tag**, the STB manufacturer has also the possibility to create two download services, each one with one **nagra_download_descriptor** containing only one PID.

1.19. Example 5

The last example adds a new notion: the **usage_type**. The STB manufacturer may want to identify what the STB is used for (development, beta tests, production...). For example, an alpha test box will be used internally with the real satellite signal; the soft will be upgraded often to test it. A beta box will be used by a little number of known customers who agree to check first the future release of software. A beta box is useful

to test a sensitive upgrade of software. In the previous example, the application software will be tested on beta boxes first, then, the nagra_download_descriptor will look like:

- Manufacturer_id = 'TST\0'
- Hardware_type = 0x00000001 (satellite boxes)
- Hardware_version = 0x00000002
- Usage_type = 0x03 (Beta boxes)
- Object_type = 0x03 (Application)
- Object_version_major = 0x00000002
- Object_version_minor = 0x00000000
- Download_type = 0x02
- Component_tag = 0x0a

In this last example, only the beta boxes will be upgraded, and, if no problem occurred, it will be possible to upgrade all the production boxes.