Doc. A/55
03 Jan 96

PROGRAM GUIDE FOR DIGITAL TELEVISION

ATSC STANDARD

ADVANCED TELEVISION SYSTEMS COMMITTEE
James C. McKinney, Chairman
Dr. Robert Hopkins, Executive Director

Blank Page

PROGRAM GUIDE FOR DIGITAL TELEVISION

ATSC STANDARD

Table Of Contents

List of Figures

List of Tables

Blank Page

PROGRAM GUIDE FOR DIGITAL TELEVISION

ATSC STANDARD

1. SCOPE

This document specifies a compact, easily acquired, and easily updated program guide data base. It provides a standardized format for transmitting data about current and future programs using private data sections in a transport stream as defined in ISO/IEC 13818-1.

This data can be acquired and processed by suitably equipped receivers to provide the user with an interactive electronic program guide.

The syntax and semantics of this program guide is flexible and can address varying needs of broadcasters: those controlling one single terrestrial channel may choose to describe the programs carried on their specific physical transmission channel only, and those controlling closed networks (e.g., cable-telco TV systems, direct broadcast satellites) may choose to provide a coordinated program guide for the entire group of physical transmission channels.

In the case where each broadcaster sends program data only about his/her own programs, a suitably equipped receiver would collect the data from each broadcast channel available at its location and assemble it into a data base that would support an interactive program guide. In the case of a closed network, this program guide structure allows the program guide information to be distributed over the various physical transmission channels as configured by the network administrator. In this case, a common master program guide is provided on all physical transmission channels, and it contains the program guide map that directs the receivers to the physical transmission channels and packet identifiers (PIDs) that carry additional program guide information.

Obviously, if two or more terrestrial broadcasters wish to coordinate and combine their program guides so that each physical transmission channel carries data about programs in two or more multiplexes, it is possible for them to do so. The only requirement is that if a master program guide is transmitted on a given physical transmission channel, this master program guide must at least include information about all programs that are transmitted in the transport multiplex of that physical transmission channel.

NOTE: The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights. By publication of this standard, no position is taken with respect to the validity of this claim, or of any patent rights in connection therewith. The patent holder has, however, filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. Details may be obtained from the publisher.

2. NORMATIVE REFERENCES

The following Recommendations and International standards contain provisions which, through reference in this text constitute provisions of this standard.

ATSC Standard A/53 (1995), ATSC Digital Television Standard.

ANSI/EIA-608-94 (1994), Recommended Practice for Line 21 Data Service.

ISO/IEC 13818-1, International Standard (1994), *MPEG-2 Systems*.

3. COMPLIANCE NOTATION

As used in this document, *"shall"* or *"will"* denotes a mandatory provision of the standard. *"Should"* denotes a provision that is recommended but not mandatory. *"May"* denotes a feature whose presence does not preclude compliance, that may or may not be present at the option of the implementer.

This document contains symbolic references to syntactic elements. These references are typographically distinguished by the use of a different font (e.g., restricted), may contain the underscore character (e.g., sequence_end_code) and may consist of character strings that are not English words (e.g., dynrng).

4. INTRODUCTION

This Program Guide consists of all the data necessary to tune channels and display available program information. The Program Guide data is conveyed using the private section format (stream_type=0x05) of the "ISO/IEC 13818-1 Information Technology — Generic Coding of Moving Pictures and Associated Audio," MPEG 2 system level transport. The Program Guide also consists of system configuration and control information, such as a system time, channel PIDs, and physical transmission channel numbers.

This Program Guide consists of a Master Program Guide (MPG), several optional special program guides (SPG) and several optional information streams (Description Information Parcels: DIPs and Private Information Parcels: PIPs), which are configurable by the broadcaster. The contents of the private data streams, that are defined in this Program Guide are identified using the registration_descriptor (Section 2.6.8 of ISO/IEC 13818-1) and the private_data_indicator descriptor (Section 2.6.29 of ISO/IEC 13818-1) within the MPG. When the Program Guide is transmitted within ATSC Digital Television bit stream, the registration authority for the private_data_indicator is ATSC. Figure 4.1 illustrates the general structure of this electronic program guide.

The MPG indicates all current programming and the presence of further descriptive material within the network. In essence, the MPG is the only required stream for the Program Guide. If one Program Guide describes programs in more than one physical transmission channel, the same MPG has to be transmitted on all related physical transmission channels.

Any special program guides, SPGs, other than the MPG are optional and may provide additional slot information for future events, alternate channel information, or other guide information as configured by the service provider.

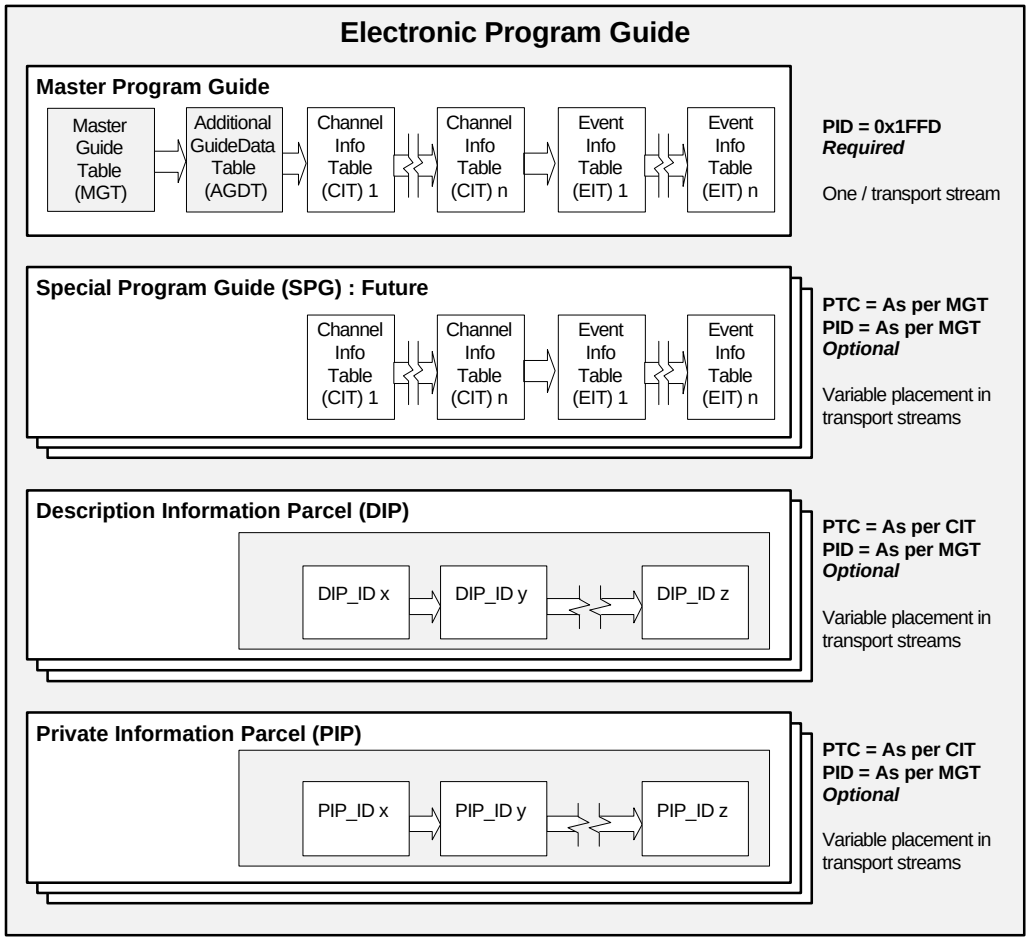

Figure 4.1. General structure of the Electronic Program Guide. (PID: Packet Identifier, PTC: Physical Transmission Channel).

The time data is sent as GMT, this allows the program guide to cover networks that may span time zones. The local time is derived from this GMT time information. The guide also provides provision for day light savings adjustments in every locality. The time critical data resides in the Master Program Guide (MPG) which is sent on each data multiplex at a maximum rate of 100kbps. The amount of this time critical data never exceeds 64 kbytes.

The data structures and transmission requirements of the Program Guide described in this document are specified to optimize both the transmission bandwidth requirements, as well as the complexity of implementation at the receiver. The structures are listed in binary form, most significant bit first (to the left). Integers and Double Length Integer quantities should be put into these byte formats with the most significant byte first, in the

lowest address. The left byte of any quantity should be placed as the lowest byte address of that quantity.

The document is organized as follows: The MPG is described in Section 5, and SPG is described in Section 6. Section 7 describes the Description Information Parcels (DIP) and Private Information Parcels (PIP). Section 8 discusses the transmission requirements of the program guide. Annex A explains the daylight savings time control. The service paradigm list and the default labels are provided in Annexes B and C, respectively.

5. MASTER PROGRAM GUIDE

The Master Program Guide (MPG) describes all of the current programming information and reference information for all of the other optional guides. The MPG is the only required stream for the Program Guide. If one Program Guide describes programs in more than one physical transmission channel, the same MPG has to be transmitted on all related physical transmission channels.

The Master Program Guide is defined in four major tables: Master Guide Table (MGT); Additional Guide Data Table (AGDT); Channel Information Table (CIT); and Event Information Table (EIT).

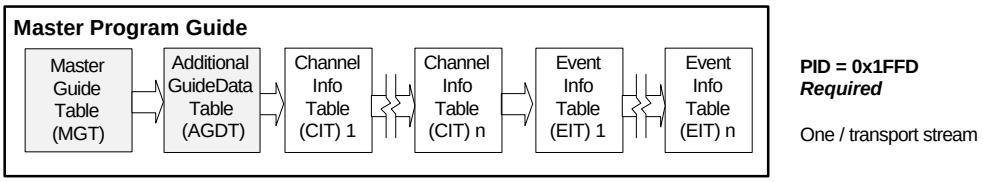

Figure 5.1. General structure of the Master Program Guide.

The MGT carries the registration_descriptor and private_data_indicator_descriptor that identifies the contents of the private data streams defined in this program guide. In addition, the MGT carries time information, the PID list, the channel grouping list, and information regarding the size of the MPG.

The AGDT carries the time base list for the available physical transmission channels, the Program Guide Map, and Default Override Records.

The CIT contains the channel information that does not change with time during the life of MPG. This includes the PIDs that the streams are carried on, the virtual channel number, physical transmission channels, etc. The CIT contains the information needed to determine the video, audio, and data PIDs, physical transmission channel number, and channel information of any available channel within the program guide.

The EIT contains the channel event information (programming event titles, start times, etc.) for programming events on defined channels.

The table IDs of these tables are listed in Table 5.1. The MPG tables are carried in the PID 0x1FFD as defined in this document. This PID shall be exclusively reserved for MPG stream only. Within these tables several other data structures are defined. The following subsections contain a description of each of the data structures.

5.1 Master Guide Table (MGT)

The Master Guide Table (MGT) carries the registration_descriptor and the private_data_indicator_descriptor that identify the contents of the private data streams defined in this program guide. In addition, the MGT carries time information, the PID list, the

channel grouping list, and information regarding the size of the MPG. The Master Guide Table only appears in the MPG.

Table 5.1 Program Guide Structures and Table IDs

Structure	Acronym	Table ID	PID
Master Guide Table	MGT	0xE0	0x1FFD
Additional Guide Data Table	AGDT	0xE1	0x1FFD
Channel Information Table	CIT	0xE2	depends on guide
Event Information Table	EIT	0xE3	depends on guide
Description Information Parcel	DIP	0xE4	As per MGT
Private Information Parcel	PIP	0xE5	As per MGT
ATSC reserved		0xE6-0xEF	ATSC reserved

In order to make data acquisition for large Program Guides covering large number of virtual channels easy, the concept of Channel Groupings is defined. Channel Groupings help to partition the channels in a Program Guide into non-overlapping, contiguous groups. There can be a maximum of 16 channel groupings (0-15) with a maximum of 64 channels per Channel Grouping. Each Channel Grouping has its own Channel Information Table (CIT) section. The events that are scheduled to be presented on those channels are listed in Event Information Table sections.

Once the channel groupings are defined in the MGT, all guides, MPG as well as SPGs have to obey this grouping structure. Each channel can be described only in the EIT and CIT sections corresponding to the channel grouping whose range includes the virtual channel number of that specific channel. For example, if Channel Grouping 5 covers channel numbers in the range 300 to 600 (not more than 64 of them are actually present), channel 500, if it exists, has to be defined in channel grouping 5, and cannot be defined in any other channel grouping.

In the case of one broadcaster, who is describing a single data multiplex on one single physical channel, the concept of channel grouping is not important. In this case the broadcaster may define only one group consisting of all the virtual channels in his data multiplex, as long as there are less than 65 virtual channels in this multiplex. To describe more than 64 virtual channels, at least two channel groupings would be necessary.

The Master Guide Table is carried in a single Transport Packet with PID 0x1FFD, and obeys the syntax and semantics of the Private Section as described in Section 2.4.4.10 and 2.4.4.11 of ISO/IEC 13818-1. The following constraints apply to the Transport packet carrying the MGT:

- PID shall have the value 0x1FFD
- transport_scrambling_control bits shall have the value ‘00’
- adaptation_field_control bits shall have the value ‘01’
- payload_unit_start_indicator shall be 1
- pointer_field shall have the value 0x00

- the remainder of the transport packet after the MGT section shall be padded with stuffing bytes of value 0xFF.

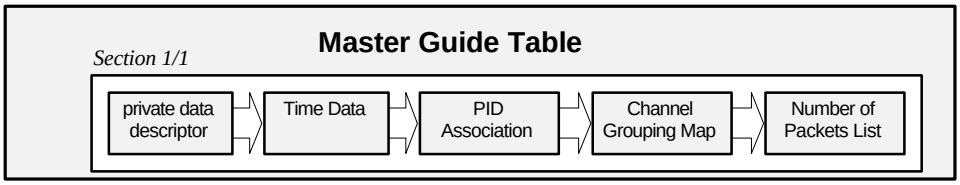

Figure 5.2. General structure of the Master Guide Table.

Table 5.2 Bit Stream Syntax for the Master Guide Table

Syntax	Bits	Format
master_guide_table_section () {		
table_id	8	0xE0
section_syntax_indicator	1	'1'
private_indicator	1	'1'
reserved	2	'11'
private_section_length	12	uimsbf
table_id_extension	16	0x0000
reserved	2	'11'
version_number	5	uimsbf
current_next_indicator	1	'1'
section_number	8	0x00
last_section_number	8	0x00
registration_descriptor () {		
descriptor_tag	8	0x05
descriptor_length	8	uimsbf
format_identifier	32	uimsbf
for (I = 0;I< N;I++)		
additional_identification_info	8	bslbf
}		
private_data_indicator_descriptor () {		
descriptor_tag	8	0x0F
descriptor_length	8	0x04
private_data_indicator	32	uimsbf
}		
daylight_savings	16	bslbf
life_time	16	uimsbf
app_time () {		
app_time_day	8	uimsbf
app_time_slot	8	uimsbf
}		
actual_GMT_time () {		
actual_time_day_of_week	3	uimsbf
actual_time_days_in_month	5	uimsbf
actual_time_year	8	uimsbf
actual_time_month	8	uimsbf
actual_time_day	8	uimsbf
actual_time_hour	8	uimsbf

Syntax	Bits	Format
actual_time_minute	8	uimsbf
actual_time_second	8	uimsbf
}		
PIDs () {		
zero_bits	3	'000'
UP_PID	13	uimsbf
zero_bits	3	'000'
PIP_PID	13	uimsbf
zero_bits	3	'000'
DIP_PID	13	uimsbf
}		
width_in_slots	8	uimsbf
number_channel_groupings	8	uimsbf
channel_grouping_list () {		
for (I = 0;I< number_channel_groupings;I++)		
zero_bits	6	'000000'
start_channel [I]	10	uimsbf
}		
Npkts_list () {		
Num_pkts_MPG	8	uimsbf
Num_pkts_AGDT	8	uimsbf
for (I = 0;I< number_channel_groupings;I++) {		
Num_pkts_CIT[I]	8	uimsbf
Num_pkts_EIT[I]	8	uimsbf
}		
}		
first_link	8	uimsbf
CRC_32	32	rpchof
}		

table_id — This is an 8-bit field, which shall be set to 0xE0, identifying this table as the Master Guide Table.

section_syntax_indicator — This 1-bit field shall be set to '1'. It denotes that the section follows the generic section syntax beyond section length field.

private_indicator— This 1-bit field shall be set to '1'.

reserved — Reserved bits that shall be set to '11'.

section_length — 12-bit field specifying the number of remaining bytes in this section immediately following the section_length field up to the end of the section. The value of the section_length shall be smaller than 181. The whole MGT section shall be transmitted in one Transport Packet.

table_id_extension— This 16-bit field shall be set to 0x0000.

version_number — This 5-bit field is the version number of the whole program guide. The version number shall be incremented by 1 modulo 32 when a field in either the MPG or any SPG changes with the exception of the life_time and actual_time fields in the Master Guide Table.

current_next_indicator — This 1-bit indicator is always set to '1' for program guide sections; the program guide sent is always currently applicable.

section_number — The value of this 8-bit field shall always be 0x00 (this table is only one section long).

last_section_number — The value of this 8-bit field shall always be 0x00.

registration_descriptor — This ISO/IEC 13818-1 descriptor identifies the registration authority, controlling the private_data_indicator field within the private_data_indicator_descriptor:

descriptor_tag — This 8-bit unsigned integer field shall have the value 5, identifying this descriptor as registration_descriptor.

descriptor_length — This 8-bit unsigned integer field specifies the number of bytes of the registration_descriptor immediately following descriptor_length field.

format_identifier — This 32-bit field identifies the registration authority that assigns the private_data_indicator field within the private_data_indicator_descriptor. Within ATSC Digital TV bit stream, the value of the format_identifier is 0x41545343 (ATSC in ASCII) defining the registration authority to be ATSC.

private_data_indicator_descriptor — This ISO/IEC 13818-1 descriptor, in conjunction with the registration_descriptor, uniquely identifies the constructor of the user private data streams (UP_PID and PIP_PID), and definition of user defined SType and SPI_indicator values (Annex B) that are defined within this Program Guide. This allows a receiver to determine whether it can process the user defined fields (known registration authority and private_data_indicator), and if it can, to determine what the intended purpose of the user defined fields are:

descriptor_tag — This 8-bit unsigned integer field shall have the value 15, identifying this descriptor as private_data_indicator_descriptor.

descriptor_length — This 8-bit unsigned integer field shall have the value 4, specifying the number of bytes of the private_data_indicator field.

private_data_indicator — The value of the private_data_indicator is assigned to a user by the registration authority identified in the registration_descriptor. If neither the UP_PID nor PIP_PID are used, and there are no user-defined SType and SPI_indicator values, there is no need to identify the user; hence private_data_indicator can be set to 0.

daylight_savings — Daylight Savings Time Control bytes. Refer to Annex A for the use of these two bytes.

life_time — This 16-bit unsigned integer field indicates the time in seconds to the next change in any program guide field, except the life_time and actual_time fields in the Master Guide Table. Although this field represents the intended life time of a guide, changes in the program guide shall be reflected through the version_number.

app_time — These two bytes indicate the day and time slot number (in terms of 30 minute slots, where each day is divided into 48 time slots each covering 30 minutes) of the left column slot time (beginning of this MPG).

app_time_day— 8-bit unsigned integer field indicating the day (1-31) of the left slot.

app_time_slot — 8-bit unsigned integer field indicating the slot number (0-47) of the left slot.

Example: A Master Program Guide describing a program starting at 3:00 a.m. on the 10th day of the month would have app_time_day = 10 and app_time_slot = 6 (6 slots of 30 minutes each, offset from the beginning of the day to come to 3:00 a.m.).

actual_GMT_time — Current actual time (in GMT) of arrival of this field in the receiver. In addition to the actual_time offset and daylight_savings fields, there are two additional pieces of data necessary for the receiver to derive the correct local time: time offset (for time zone correction, indicating the difference between GMT and local time) and an indicator whether daylight savings correction is used in the locality. Both of these data may be supplied through Conditional Access system or through user interface.

The values of the fields indicate:

actual_time_day_of_week— Day of the week (1=Sunday - 7 = Saturday).

actual_time_days_in_month— Number of days in current month (28-31).

actual_time_year— Year minus 1900.

actual_time_month— Month of the year (1-12) (Jan - Dec.).

actual_time_day— Day of the month (1-31).

actual_time_hour— Current hour (0-23).

actual_time_minute— Current minute (0-59).

actual_time_second— Current second (0-59).

PIDs — The PIDs are 13-bit unique packet identifiers identifying each transport packet belonging to an elementary stream within an MPEG Transport stream:

UP_PID — User Private PID. This PID shall be exclusively reserved for the user private stream, and exists in all physical transmission channels the program guide covers. The content of the UP_PID is defined by the user as identified by the registration_descriptor and private_data_indicator fields. If not used,UP_PID value shall be set to 0x1FFF.

PIP_PID — Private Information Parcel PID. This PID shall be exclusively reserved for the PIP stream only. The content of the private_information field in the PIP_PID is defined by the user as identified by registration_descriptor and the private_data_indicator fields. If not used, PIP_PID value shall be set to 0x1FFF.

DIP_PID — Description Information Parcel PID. This PID shall be exclusively reserved for the DIP stream only. If not used,DIP_PID value shall be set to 0x1FFF.

width_in_slots— This is a byte that gives the width of the MPG in 30-minute slots.

Example: value of 10 means that the MPG covers five hours of programming.

number_channel_groupings — This is a byte that gives the number of Channel Groupings being used (1 to 16, inclusive).

channel_grouping_list — Variable length list (number_channel_groupings * 2 bytes) describing how the list of channels is partitioned into Channel Groupings. The sequentially increasing, by channel number, list of all channels in use is partitioned into non-overlapping segments. The top channel number in the last grouping is assumed to be 999.

start_channel[i] — The first channel number (lowest channel number) in the i-th Channel Grouping (CG). A channel number shall be between 1 and 999, inclusive. The channel number corresponds to the program number in MPEG System PSI sections.

Example: If a program guide describes 100 virtual channels, with channel numbers 100 to 119, 200 to 209, 300 to 319 and 950 to 999, one possible channel grouping of the Program Guide can be constructed such that channels 100 to 319 are in CG 0 (50 actual channels in CG 0) and channels 950 to 999 are in CG 1 (50 channels in CG 1). Hence, number_channel_groupings = 2, start_channel[0] = 100 and start_channel[1] = 950. Note that with these values, CG 0 covers all channels between 100 and 949 (only 64 of them can exist at the same time) and CG 1 covers all channels between 950 and 999. Therefore, if the service provider wants to send information for a new channel with channel number 910, this channel has to be defined within CG 0.

An alternative grouping structure could be to assign channels 100-119 to CG 0, 200 - 209 to CG 1, 300 - 319 to CG 2 and channels 950 to 999 to CG 3.

In this case, number_channel_groupings = 4, start_channel[0] = 100, start_channel[1] = 200, start_channel[2] = 300, and start_channel[3] = 950. CG 0 covers 100 - 199, CG 1 covers 200 - 299, CG 2 covers 300 - 949 and CG 3 covers 950 - 999. In this case, channel number 910 would have to be defined in CG 2.

Npkts_list — Variable length list (number_channel_groupings * 2 + 2 bytes) indicating the number of Transport packets in MPG, AGDT, CITs and EITs:

Num_pkts_MPG — This 8-bit unsigned integer field indicates the total number of transport packets carrying the whole MPG in PID 0x1FFD; i.e., total number of transport packets with PID 0x1FFD carrying the MGT, AGDT, all CITs and all EITs (as indicated by the number_channel_groupings value).

Num_pkts_AGDT — This 8-bit unsigned integer field indicates the total number of transport packets carrying the Additional Guide Data Table section in PID 0x1FFD.

Num_pkts_CIT[i] — This 8-bit unsigned integer field indicates the total number of transport packets carrying the Channel Information Table section for the i-th Channel Grouping in PID 0x1FFD.

Num_pkts_EIT[i] — This 8-bit unsigned integer field indicates the total number of transport packets carrying all Event Information Table sections for all channels of the i-th Channel Grouping in PID 0x1FFD.

first_link — This 8-bit unsigned integer field indicates the number of the first linked guide (0 to 15 inclusive). Guide data streams other than the MPG are termed Special Program Guides (SPGs). SPGs are numbered from 1 to 15 indicating the index in the Program Guide Map in the Additional Guide Data Table (Section 5.2.2). SPGs can be forward linked from the MPG using this field. This field is set to zero for a guide that is not linked.

Example: If MPG covers programs from the present through the next 5 hours, and SPG 1 covers programs from +5 to +24 hours, then SPG 1 would be linked to MPG using a value of 1 in the first_link field.

CRC_32 — This is a 32-bit field that contains the CRC value that ensures a zero output from the registers in the decoder defined in Annex B of ISO/IEC 13818-1 "MPEG-2 Systems" after processing the entire Transport Stream Master Guide Table section.

5.2 Additional Guide Data Table (AGDT)

The Additional Guide Data Table (AGDT) carries the time_base list for the available physical transmission channels, the Program Guide Map, and the Default Override Records. The Additional Guide Data Table only appears in the MPG.

The Additional Guide Data Table is carried in a single private section with table ID 0xE1 in PID 0x1FFD, and obeys the syntax and semantics of the Private Section as described in Section 2.4.4.10 and 2.4.4.11 of ISO/IEC 13818-1. The following constraints apply to the Transport packets carrying the AGDT:

- PID shall have the value 0x1FFD
- transport_scrambling_control bits shall have the value '00'
- adaptation_field_control bits shall have the value '01'
- payload_unit_start_indicator of the transport packet carrying the table_id field of the AGDT section shall be 1 (first transport packet of the section)
- pointer_field of the transport packet carrying the table_id field of the AGDT section shall have the value 0x00 (first transport packet of the section)
- the remainder of the transport packet carrying the CRC_32 field of the AGDT section shall be padded with stuffing bytes of value 0xFF (last transport packet of the section)

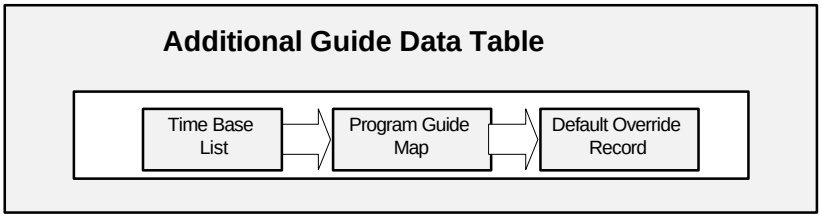

Figure 5.3. General structure of the Additional Guide Data Table.

table_id — This is an 8-bit field, which shall be set to 0xE1, identifying this table as the Additional Guide Data Table.

section_syntax_indicator — This 1-bit field shall be set to '1'. It denotes that the section follows the generic section syntax beyond section length field.

private_indicator— This 1-bit field shall be set to '1'.

Table 5.3 Bit Stream Syntax for the Additional Guide Data

Syntax	Bits	Format
additional_guide_data_section () {		
table_id	8	0xE1
section_syntax_indicator	1	'1'
private_indicator	1	'1'
reserved	2	'11'
private_section_length	12	uimsbf
table_id_extension	16	0x0000
reserved	2	'11'
version_number	5	uimsbf
current_next_indicator	1	'1'
section_number	8	0x00
last_section_number	8	0x00
time_base_list ()	var	
program_guide_map ()	var	
number_DOR	8	uimsbf
for (I = 0;I< number_DOR;I++) {		
default_override_record (I)	var	
}		
CRC_32	32	rpchof
}		

reserved — Reserved bits that shall be set to '11'.

section_length — 12-bit field specifying the number of remaining bytes in this section immediately following the section_length field up to the end of the section, including CRC_32 field. The value of this field shall not exceed 4093.

table_id_extension— This 16-bit field shall be set to 0x0000.

version_number — This 5-bit field is the version number of the whole program guide. The version number shall be incremented by 1 modulo 32 when a field in either the MPG or any SPG changes with the exception of the life_time and actual_time fields in the Master Guide Table.

current_next_indicator — This 1-bit indicator is always set to '1' for program guide sections; the program guide sent is always currently applicable.

section_number — The value of this 8-bit field shall always be 0x00 (this table is only one section long).

last_section_number— The value of this 8-bit field shall always be 0x00.

time_base_list — The time_base_list gives a list of available physical transmission channels, and one PCR_PID number for each physical transmission channel (time_base_list is described in Section 5.2.1).

program_guide_map — The program_guide_map defines the optional SPGs in the program guide (program_guide_map is described in Sec. 5.2.2).

number_DOR— Number of default_override_records that follow (0-230).

default_override_record(l) — The default_override_record allows for the names of program ratings and theme category labels to be redefined (default_override_record is described in Section 5.2.3).

CRC_32 — This is a 32-bit field that contains the CRC value that ensures a zero output from the registers in the decoder defined in Annex B of ISO/IEC 13818-1 "MPEG-2 Systems" after processing the entire Transport Stream Additional Guide Data section.

5.2.1 The time_base_list

The time_base_list provides the number of physical channels available and one PID number on each physical transmission channel that carries the PCR fields for the time base. The purpose of this list is to point to at least one PID on each physical channel that has time base information. If the transport stream carries any video or audio programs, there needs to be at least one, or possibly multiple PCR_PIDs (one for each program) in the transport stream that carry PCR fields. The time_base_PID in this list can be any of those PCR carrying PIDs.

The purpose of the time_base_list is to have a pointer in the receiver that makes access to a time base possible in cases where no PCR is associated with a program definition for private streams. Note that the time_base_PID in this list may not be valid for a specific program on that transport stream. The valid time base for each time-related program is defined in the Program Map Table section for that specific program through PCR_PID definition (ISO/IEC 13818-1, Section 2.4.4.9).

Table 5.4 Bit Stream Syntax for the time_base_list

Syntax	Bits	Format
time_base_list () {		
number_PTCs	8	uimsbf
for (I = 0;I< number_PTCs;I++) {		
zero_bits	3	'000'
time_base_PID [I]	13	uimsbf
}		
}		

number_PTCs — This 8-bit unsigned integer field indicates the number of physical transmission channels accessible with this MPG (the value would be 1 for a single physical channel).

time_base_PID — One PCR_PID on the corresponding physical transmission channel. Set to 0x1FFF if the physical transmission channel does not have any PCR_PID. The first PID corresponds to physical transmission channel 0, and the last PID corresponds to the physical transmission channel given by (number_PTCs - 1).

5.2.2 The Program Guide Map (PGM)

The Program Guide Map defines up to 15 optional special program guides (SPGs) in the Program Guide. Multiple program guides are allowed in the system. These guides

may be linked or independent. All guides must be defined in the PGM. Program Guide number 0 is reserved for the MPG. One entry must be created for each SPG being transmitted. The first entry corresponds to SPG 1, the second to SPG 2, etc., in ascending sequential order.

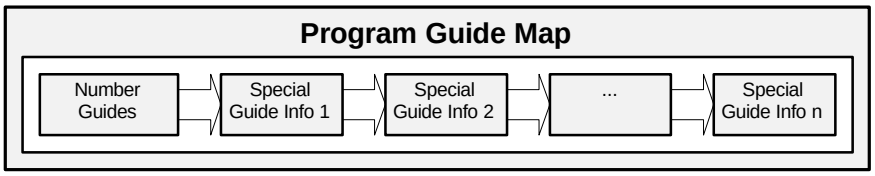

Figure 5.4. General structure of the Program Guide Map.

Table 5.5 Bit Stream Syntax for the Program Guide Map

Syntax	Bits	Format
program_guide_map () {		
number_guides	4	uimsbf
program_guide_map_size	12	uimsbf
for (I = 1; I< number_guides+1;I++) SPG_map(I) {		
next	4	uimsbf
previous	4	uimsbf
SPG_PTC	8	uimsbf
PG_CSS	32	bslbf
SPG_indirect	1	bslbf
reserved	3	'111'
left_column_month	4	uimsbf
reserved	3	'111'
left_column_day	5	uimsbf
reserved	2	'11'
left_column_slot	6	uimsbf
width_in_slots	16	uimsbf
if(SPG_indirect=='0') {		
zero_bits	3	'000'
SPG_PID	13	uimsbf
} else {		
SPG_ID	16	uimsbf
}		
Npkts_list_SPG (I) {		
for (i = 0;i< number_channel_groupings;i++) {		
Num_pkts_SPG[I]_CIT[i]	8	uimsbf
Num_pkts_SPG[I]_EIT[i]	8	uimsbf
}		
}		
SPG_name_size	8	uimsbf
SPG_name	8*n	ISO-LATIN-1
}		
}		

number_guides — This 4-bit unsigned integer field specifies the number of special program guides (SPGs) 0 - 15 inclusive (if 0, only MPG is transmitted, no SPGs defined).

program_guide_map_size — This 12-bit unsigned integer field specifies the number of bytes of the program_guide_map, starting immediately following the program_guide_map_size field up to the end of the program_guide_map.

next — This 4-bit unsigned integer field specifies the number of next special program guide (0 -15 inclusive) in sequence, if linked. A value of zero terminates the next link.

previous — This 4-bit unsigned integer field specifies the number of previous special program guides (0 -15 inclusive) in the sequence, if linked. A value of zero terminates the previous link and links to the MPG as the previous guide.

Example: If MPG covers programs from the present through the next 5 hours, and there are two SPGs defined such that SPG 1 covers programs from +5 to +12 hours, and SPG 2 covers programs from +12 to +24 hours, SPG 1 would have values next = 2 and previous =0, and SPG 2 would have values next = 0 and previous = 1.

SPG_PTC — This 8-bit unsigned integer field specifies the Physical Transmission Channel Number where the SPG is found (0 to 255).

PG_CSS — This 32-bit field specifies the Customer Service Segment field for this SPG. A value of 0x00000000 indicates that the SPG is visible to all customers. Other values are defined by the service provider to regulate the visibility of this SPG through Conditional Access.

SPG_indirect — This is a 1-bit flag. A value of '0' indicates that SPG is transmitted on the SPG_PTC. A value of '1' indicates that the SPG_ID is used as a source ID into the Virtual Channel Table (VCT).

left_column_month — This 4-bit field indicates the month of the left column time (start time) for this SPG (1: January - 12: December).

left_column_day — This 5-bit field indicates the day of month of the left column time (start time) for this guide (1-31 inclusive).

left_column_slot — This 6-bit field indicates the slot number of the left hand column time (start time) for this guide in terms of 30 minute slots, where each day is divided into 48 time slots each covering 30 minutes. (0-47 inclusive).

width_in_slots — This 16-bit unsigned integer field indicates the number of 30 minute time slots spanned by this SPG.

SPG_PID — The SPG_PID under which this program guide is transmitted in SPG_PTC. This PID shall be exclusively reserved for this SPG stream only.

SPG_ID — This 16-bit unsigned integer field is the application ID for the service carrying the SPG via indirection through the Virtual Channel Table (VCT).

Num_pkts_SPG[l]_CIT[i] — This 8-bit unsigned integer field indicates the total number of transport packets carrying the Channel Information Table section for the i-th Channel Grouping for SPG[l].

Num_pkts_SPG[l]_EIT[i] — This 8-bit unsigned integer field indicates the total number of transport packets carrying all Event Information Table sections for all channels of the i-th Channel Grouping for SPG[l].

SPG_name_size — This 8-bit unsigned integer field indicates the total number of bytes in the SPG_name field. The range of numbers is 0 to 20 inclusive (0 meaning no SPG_name is present).

SPG_name — Variable length field carrying ISO-LATIN-1 codes describing the program guide name for this SPG (e.g., +5 Hour, or Tuesday Morning). The name shall not have more than 20 characters.

5.2.3 Default Override Record (DOR)

Information regarding program ratings and theme category/subcategories are provided in the Event Information Table Sections for each programming event through subcategory, category, rating, content_advisory, violence, language and sex rating fields (Section 5.4). The values of these fields are interpreted by referring to default names defined in Annex C.

The Default Override Record structure allows the system provider to override these default names of program ratings and theme category/subcategory labels defined in Annex C. If no DOR are received in AGDT section, then the receiver should use the default labels defined in Annex C.

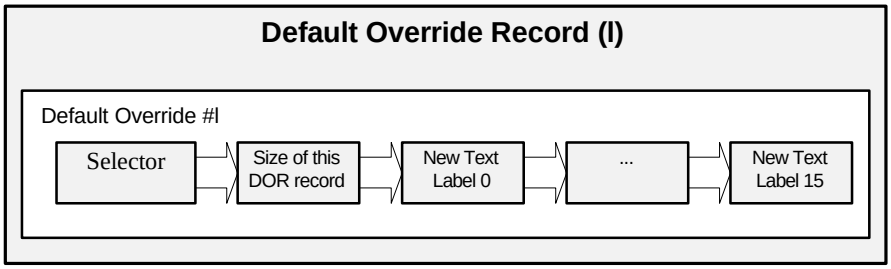

Figure 5.5. General structure of the Default Override Record.

selector — This 6-bit field indicates the type of default override record information conveyed:

selector = 0: Subcategories

selector = 1: Categories

selector = 2: Ratings

selector = 3: Violence rating

selector = 4: Language rating

selector = 5: Sex rating

All other values are reserved for future extensions.

Table 5.6 Bit Stream Syntax for the Default Override Record

Syntax	Bits	Format
default_override _record(l) {		
selector	6	bslbf
DOR_size	10	uimsbf
if (selector==0x0) {		
super_category	4	uimsbf
reserved	4	'1111'
}		
for (i = 0; i<16;i++) {		
text[i] () {		
multiple_compressed_strings ()	var	
}		
}		
}		

DOR_size — This 10-bit unsigned integer field specifies the number of bytes of the default_override_record(l), starting immediately following the DOR_size field up to the end of the default_override_record(l).

super_category — This 4-bit field exists only if the value of selector field is 0 (DOR defines new subcategory labels). In this case, the super_category field indicates the category number for which each field in this DOR is a subcategory.

text[i] — Multiple Compressed string corresponding to the i-th rating or category assignment. (See Section 5.5 for definition of multiple compressed strings).

5.3 Master Program Guide Channel Information Table (CIT)

The Channel Information Table (CIT) contains channel information that does not change with time during the life of this program guide. Any changes in the channel structure shall be conveyed with a new program guide, with a new version number. This channel information includes the PIDs that the streams are carried on, the virtual channel number, physical transmission channels, etc. The MPG CIT shall contain the information needed to determine the video, audio, and data PIDs, physical transmission channel number, and channel information of all available channels within the time span of the MPG.

The channel information table may be divided into up to 16 sections (0 to 15), each of which corresponds to one channel grouping consisting of a maximum of 64 channels. Each channel grouping must have its own CIT section. Every channel that is referenced has to be included in the corresponding channel grouping and in the corresponding CIT section. Channels from channel grouping k must be listed in section number k in ascending order, beginning with the lowest channel number, as specified by start_channel(k) field in the MGT. The entire CIT table is simply a concatenation of all the sections in order.

Once the groupings are defined in the MGT, all guides, the MPG as well as the SPGs, if present, shall obey the same grouping structure. However, SPGs may cover

different channels than the MPG. The concept of channel grouping is illustrated in Figure 5.6 with an example.

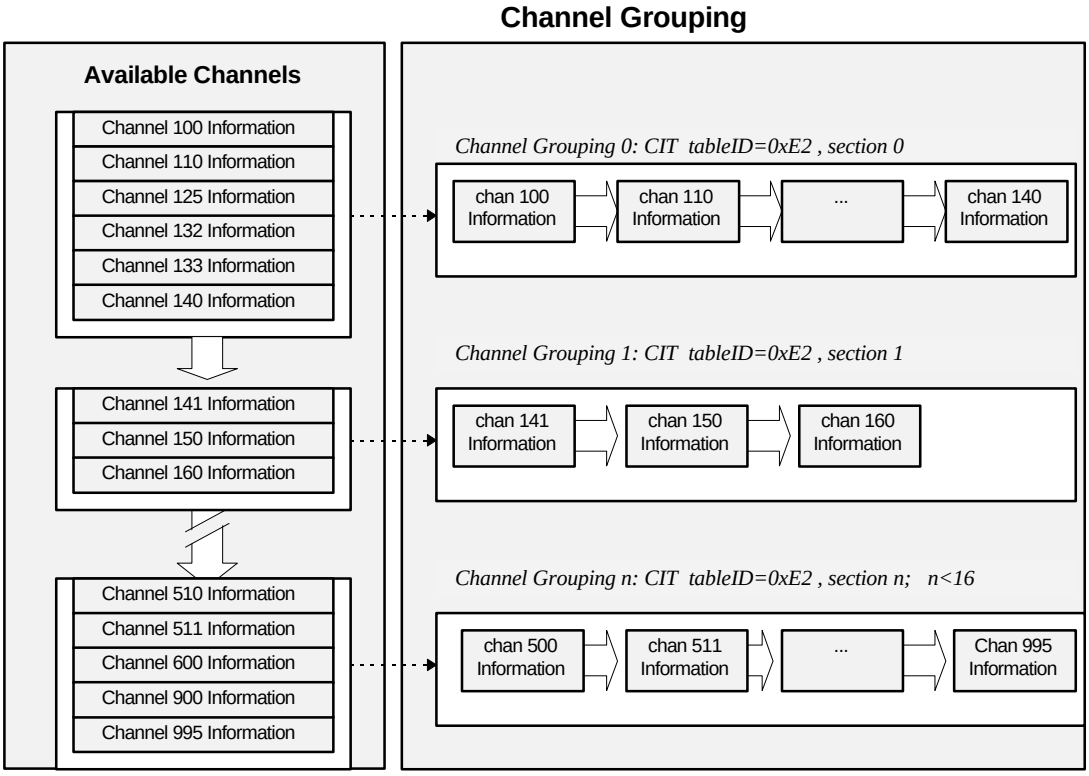

Figure 5.6. Example of channel grouping and how these channel groupings are mapped to CIT sections.

The Channel Information Table within the MPG is carried in private sections with table ID 0xE2 in PID 0x1FFD, and obeys the syntax and semantics of the Private Section as described in Section 2.4.4.10 and 2.4.4.11 of ISO/IEC 13818-1. The following constraints apply to the Transport packets carrying the CIT sections in MPG:

- PID shall have the value 0x1FFD
- transport_scrambling_control bits shall have the value '00'
- adaptation_field_control bits shall have the value '01'
- payload_unit_start_indicator of the transport packet carrying the table_id field of each CIT section shall be 1 (first transport packet of each section)
- pointer_field of the transport packet carrying the table_id field of each CIT section shall have the value 0x00 (first transport packet of each section)
- the remainder of the transport packet carrying the CRC_32 field of the CIT section shall be padded with stuffing bytes of value 0xFF (last transport packet of each section)

Table 5.7 Syntax of channel_information_table Sections

Syntax	Bits	Format
channel_information_table_section () {		
table_id	8	0xE2
section_syntax_indicator	1	'1'
private_indicator	1	'1'
reserved	2	'11'
section_length	12	uimsbf
table_id_extension	16	uimsbf
reserved	2	'11'
version_number	5	uimsbf
current_next_indicator	1	'1
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved	2	'11'
number_channels	6	uimsbf
for(l=0; l<number_channels;l++) channel_info(l){		
reserved	2	'11'
channel_number	10	uimsbf
channel_info_length	12	uimsbf
channel_PTC	8	uimsbf
short_name	8*4	ISO-LATIN-1
CI_CSS	32	bslbf
reserved	4	'1111'
number_PIDs	4	uimsbf
SPI_indicator	8	bslbf
flags (l) {		
channel_PIP_flag	2	bslbf
channel_DIP_flag	2	bslbf
indirect_mapping	1	bslbf
promo_flag	1	bslbf
provider_index_flag	1	bslbf
name_flag	1	bslbf
}		
if (SPI_indicator == 0x00) {		
for (i = 1;i<= number_PIDs;i++) {		
zero_bits	3	'000'
PID[i]	13	uimsbf
SType[i]	8	uimsbf
}		
} else {		
zero_bits	3	'000'
base_PID	13	uimsbf
}		
PIP_PTC	8	uimsbf
DIP_PTC	8	uimsbf
if (channel_PIP_flag == 0x2)		
channel_PIP_message_number	16	uimsbf
if (channel_DIP_flag == 0x2)		
channel_DIP_message_number	16	uimsbf
if (indirect_mapping == '1') {		
PIP_Application_ID	16	uimsbf
DIP_Application_ID	16	uimsbf

Syntax	Bits	Format
source_ID	16	uimsbf
}		
if (provider_index_flag == '1')		
provider_index	16	uimsbf
if (name_flag == '1')		
long_channel_name	8*n	ISO-LATIN-1
}		
CRC_32	32	rpchof
}		

table_id — This is an 8-bit field, which shall be set to 0xE2, identifying this section as belonging to the Channel Information Table.

section_syntax_indicator — This 1-bit field shall be set to '1'. It denotes that the section follows the generic section syntax beyond section length field.

private_indicator— This 1-bit field shall be set to '1'.

reserved — Reserved bits that shall be set to '11'.

section_length — 12-bit field specifying the number of remaining bytes in this section immediately following the section_length field up to the end of the section, including CRC_32 field. The value of this field shall not exceed 4093.

table_id_extension— This 16-bit field shall be set to 0x0000.

version_number — This 5-bit field is the version number of the whole program guide. The version number shall be incremented by 1 modulo 32 when a field in either the MPG or any SPG changes with the exception of the life_time and actual_time fields in the Master Guide Table.

current_next_indicator — This 1-bit indicator is always set to '1' for program guide sections: the program guide sent is always currently applicable.

section_number — The value of this 8-bit field gives the number of this section. This number shall reflect the Channel Grouping number (0-15) this section corresponds to.

last_section_number — The value of this 8-bit field shall reflect the number of Channel Grouping defined, and is equal to the value of the number_channel_groupings field given by (MGT - 1).

number_channels — This 6-bit unsigned integer field indicates the number of channels defined in this CIT section (0-63). The actual channel number is (number_channels + 1).

channel_number — This 10-bit unsigned integer field specifies the channel number that the user presses to directly access this channel (1 through 999). The channel number corresponds to the program number in MPEG system PSI sections.

channel_info_length — This 12-bit field specifies the number of remaining bytes in channel_info(l), immediately following the channel_info_length field up to the end of the channel_info(l).

channel_PTC — This 8-bit field indicates the Physical transmission channel number where service data for this channel is transmitted. Physical transmission channel number is 0 through 255.

short_name — A four character channel designator (8-bit ISO LATIN-1). For longer names, name_flag and long_channel_name fields can be used.

CI_CSS — This 32-bit field specifies the Customer Service Segment field for this channel. A value of 0x00000000 indicates that the channel information is visible to all customers. Other values are defined by the service provider to regulate the visibility of this channel information through Conditional Access.

number_PIDs — This 4-bit unsigned integer indicates the number of PIDs used for this channel (number of all elementary streams comprising this channel: video, audio, data, etc.)

SPI_indicator — An 8-bit indicator called the Service Paradigm Indicator. If SPI = 0x0 then an explicit PID list is given within this channel_info. For non zero values, Service Paradigms listed in Annex B apply with the base_PID given in this channel_info.

channel_PIP_flag — 2 bits indicating how the PIP_ID for this channel is constructed (see Section 7.3.2).

channel_PIP_flag = 0x0: there is no PIP for this channel.

channel_PIP_flag = 0x1: the PIP_ID for this channel is constructed implicitly.

channel_PIP_flag = 0x2: the PIP_ID for this channel is constructed explicitly using the channel_PIP_message_number field.

channel_PIP_flag = 0x3: reserved.

The content of the private_information field in the PIP_PID is defined by the user as identified by registration_descriptor and the private_data_indicator fields in the MGT.

channel_DIP_flag — 2 bits indicating how the DIP_ID for this channel is constructed (see Section 7.3.1).

channel_DIP_flag = 0x0: there is no DIP for this channel.

channel_DIP_flag = 0x1: the DIP_ID for this channel is constructed implicitly.

channel_DIP_flag = 0x2: the DIP_ID for this channel is constructed explicitly using the channel_DIP_message_number field.

channel_DIP_flag = 0x3: reserved.

indirect_mapping— 1-bit flag, indicating channel indirection.

indirect_mapping = 1: indirection (channel description is done through pointing to a virtual channel table).

indirect_mapping = 0: direct channel definition.

promo_flag — 1-bit flag. Value ‘1’ indicates that this is a promotional channel (Coming Attractions).

provider_index_flag— 1-bit flag. If set to '1', then the provider_index field exists.

name_flag — 1-bit flag. If set to '1', then long_channel_name field exists.

PID [i] — Packet Identifier for the i-th stream of this channel.

SType [i] — This 8-bit unsigned integer field is the type of service provided by the i-th data stream. If a SType defines an audio service, the SType also defines the language for the audio. For more information on the data types of explicit list service items see Annex B.

base_PID — The base PID for service paradigm implicit PID list as defined in Annex B.

PIP_PTC — PIP_PTC indicates the physical transmission channel number where PIPs for this channel are transmitted. Together with the PIP_PID number defined in the MGT, this field uniquely identifies the PIP stream for this channel. The physical transmission channel number is 0 through 255. The content of the private_information field in the PIP_PID is defined by the user as identified by registration_descriptor and the private_data_indicator fields in the MGT.

DIP_PTC — Physical transmission channel number where DIPs for this channel are transmitted. Together with the DIP_PID number defined in the MGT, this field uniquely identifies the DIP stream for this channel. The physical transmission channel number is 0 through 255.

channel_PIP_message_number — Explicit PIP message number for this channel. If channel_PIP_flag = 0 x 2 then this 16-bit field is used to construct the PIP_ID for this channel (see Section 7.3.2).

channel_DIP_message_number — Explicit DIP message number for this channel. If channel_DIP_flag = 0 x 2 then this 16-bit field is used to construct the DIP_ID for this channel (see Section 7.3.1).

PIP_application_ID — This 16-bit unsigned integer field indicates the application_ID used as a reference into a virtual channel table (VCT) to locate a service carrying PIPs for this channel.

DIP_application_ID — This 16-bit unsigned integer field indicates the application_ID used as a reference into a virtual channel table (VCT) to locate a service carrying DIPs for this channel.

source_ID — This 16-bit field indicates the source_ID to be used as index into virtual channel table (VCT).

provider_index — This 16-bit unsigned integer field is the index of the provider. The registration authority for provider_index is ATSC.

long_channel_name — ISO-LATIN-1 codes describing the channel name. The name shall be not more than 10 characters.

CRC_32 — This is a 32-bit field that contains the CRC value that ensures a zero output from the registers in the decoder defined in Annex B of ISO-13818-1 "MPEG-2 Systems" after processing the entire Transport Stream Channel Information Table section.

5.4 Master Program Guide Event Information Table (EIT)

The Event Information Table (EIT) contains the channel event information (programming event titles, start times, etc.) for programming events on defined channels. An event is a specific segment of programming content with a start and stop time.

The EIT is partitioned into a Private Section as described in Section 2.4.4.10 and 2.4.4.11 of ISO/IEC 13818-1. The table ID for EIT sections is 0xE3. Each EIT section carries information on programming events in one channel.

As discussed in Section 5.3, a variable number of adjacent logical channel numbers are grouped together in Channel Groupings, and the information regarding these channels is conveyed through CIT sections, each section corresponding to one channel grouping. The EIT sections are tied to the CIT sections and channel numbers through table_id_extension field.

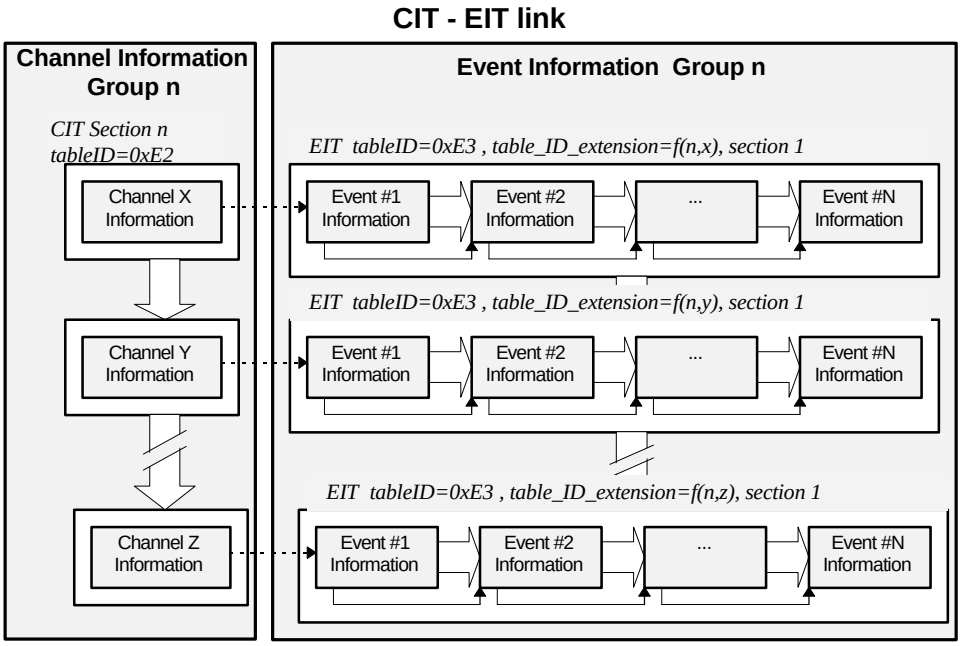

Figure 5.7. An example of channel grouping/CIT section link to the EIT sections. The channel grouping number and channel number are reflected in the table_ID_extension field.

When an event is in progress during the current time slot (i.e. the time slot in which it started is no longer in the guide), information about that event must continue to be sent in the current time slot. Since the amount of data needed to describe an event varies, a pointer to the next event in that channel is available for easy traversal of the section (event_info_length field).

The title of the programming event must be sent for each event. The program event title shall be no longer than 80 characters.

The following constraints apply to the Transport packets carrying the EIT sections in the MPG:

- PID shall have the value 0x1FFD
- transport_scrambling_control bits shall have the value '00'
- adaptation_field_control bits shall have the value '01'
- payload_unit_start_indicator of the transport packet carrying the table_id field of the EIT section shall be 1 (first transport packet of every section)
- pointer_field of the transport packet carrying the table_id field of each EIT section shall have the value 0x00 (first transport packet of every section)
- the remainder of the transport packet carrying the CRC_32 field of the EIT section shall be padded with stuffing bytes of value 0xFF (last transport packet of each section)

Table 5.8 Bit Stream Syntax for the Event Information Table Sections

Syntax	Bits	Format
event_information_table_section () {		
table_id	8	0xE3
section_syntax_indicator	1	'1'
private_indicator	1	'1'
reserved	2	'11'
section_length	12	uimsbf
table_id_extension	16	uimsbf
reserved	2	'11'
version_number	5	uimsbf
current_next_indicator	1	'1'
section_number	8	uimsbf
last_section_number	8	uimsbf
number_events	16	uimsbf
for (I = 1; I<= to number_events;I++) event_info(I){		
reserved	4	'1111'
event_info_length	12	uimsbf
event_repetition_flag	1	bslbf
taping_flag	1	bslbf
closed_captioning_flag	1	bslbf
program_event_ID_flag	1	bslbf
event_PIP_flag	2	bslbf
event_DIP_flag	2	bslbf
category	4	uimsbf
subcategory	4	uimsbf
reserved	1	'1'
start_month	4	uimsbf
start_day_of_month	5	uimsbf
start_slot	6	uimsbf
start_minute_offset	8	uimsbf
length_in_slots	8	uimsbf
end_minute_offset	8	uimsbf
reserved	4	'1111'

Syntax	Bits	Format
rating	4	uimsbf
content_advisory	4	uimsbf
violence_rating	4	uimsbf
language_rating	4	uimsbf
sex_rating	4	uimsbf
if (event_repetition_flag == '1') start_list(){		
number_start_times	8	uimsbf
for (i = 1; i<= number_start_times;i++) {		
start_slot [i]	8	uimsbf
start_minute_offset [i]	8	uimsbf
length_in_slots [i]	4	uimsbf
repeat_event_PIP_flag[i]	2	bslbf
reserved	2	'11'
if (repeat_event_PIP_flag [i] == '10') {		
repeat_event_PIP_message_number [i]	16	uimsbf
}		
}		
}		
if (program_event_ID_flag=='1') {		
program_event_ID	24	uimsbf
}		
if (event_PIP_flag == '10') {		
event_PIP_message_number	16	uimsbf
}		
if (event_DIP_flag == '10') {		
event_DIP_message_number	16	uimsbf
}		
compressed_title () {		
multiple_compressed_strings ()	var	
}		
}		
CRC_32	32	rpchof
}		

table_id — This is an 8-bit field, which shall be set to 0xE3, identifying this section as belonging to the Event Information Table.

section_syntax_indicator — This 1-bit field shall be set to '1'. It denotes that the section follows the generic section syntax beyond the section length field.

private_indicator— This 1-bit field shall be set to '1'.

reserved — Reserved bits that shall be set to '11'.

section_length — 12-bit field specifying the number of remaining bytes in this section immediately following the section_length field up to the end of the section, including CRC_32 field. The value of this field shall not exceed 4093.

table_id_extension — This 16-bit field identifies the Channel Grouping and channel number of this channel. This field is constructed as follows. The 4 most significant bits represent the Channel Grouping number (which is also equal to the CIT section number of the CIT section in which this channel is described). Of the remaining 12 least significant bits, the

two left bits are set to ‘00’, and the remaining 10 least significant bits represent the channel number (10-bit unsigned integer field specifies the channel number, 1 through 999, that the user presses to directly access this channel).

program_event_ID_flag — This 1-bit flag signals if this event has a program_event_ID field. Value ‘1’ indicates that this event has a program_event_ID field, value ‘0’ indicates that this event does not have a program_event_ID field. If this bit is set to ‘1’, the value of the provider_index_flag in the CIT section of this channel shall be set also to ‘1’ and a provider_index field shall occur in the corresponding CIT section.

version_number — This 5-bit field is the version number of the whole program guide. The version number shall be incremented by 1 modulo 32 when a field in either the MPG or any SPG changes with the exception of the life_time and actual_time fields in the Master Guide Table.

current_next_indicator — This 1-bit indicator is always set to ‘1’ for program guide sections; the program guide sent is always currently applicable.

section_number — The value of this 8-bit field shall always be 0x00 (EIT info for one channel in MPG is always conveyed in one section).

last_section_number — The value of this 8-bit field shall always be 0x00 (EIT info for one channel in MPG is always conveyed in one section).

number_events — Indicates the number of programming events in this EIT section for this channel.

event_info_length — 12-bit field specifying the number of remaining bytes in this event info field immediately following the event_info_length field up to the end of the event information.

event_repetition_flag — This 1-bit flag signals the repetitive transmission of the same event in this channel. A value of ‘1’ indicates that this event is transmitted repeatedly in the same channel and has multiple start times as indicated in the start_list field. A value of ‘0’ indicates there is no repetition.

taping_flag — This 1-bit flag signals the taping permission (for information purposes only): Value of ‘1’ indicates no taping is allowed, value ‘0’ indicates that taping is allowed.

closed_captioning_flag — This 1-bit flag signals if this event has closed captioning. Value ‘1’ indicates that this event is closed captioned, value ‘0’ indicates that there is no closed captioning.

event_PIP_flag — 2 bits indicating how the PIP_ID for this event is constructed (see Section 7.3.4).

If event_PIP_flag = 0x0 then there is no PIP for this event.

If event_PIP_flag = 0x1 then thePIP_ID for this event is constructed implicitly.

If event_PIP_flag = 0x2 then thePIP_ID for this event is constructed explicitly.

event_PIP_flag = 0x3 is reserved.

The content of the private_information field in the PIP_PID is defined by the user as identified by registration_descriptor and the private_data_indicator fields in the MGT.

event_DIP_flag — 2 bits indicating how the DIP_ID for this event is constructed (see Section 7.3.1).

If event_DIP_flag = 0x0 then there is no DIP for this event.

If event_DIP_flag = 0x1 then the DIP_ID for this event is constructed implicitly.

If event_DIP_flag = 0x2 then the DIP_ID for this event is constructed explicitly.

event_DIP_flag = 0x3 is reserved.

start_month — This 4-bit unsigned integer field indicates the month of start of this event (1: January - 12: December).

start_day_of_month — This 5-bit unsigned integer field indicates the absolute day of month of start of this event.(1-31).

start_slot — This 6-bit unsigned integer field (0-47 inclusive) indicates the slot number associated with start of this event. Each day is divided into 48 half hour slots.

start_minute_offset — This 8-bit unsigned integer field indicates the offset in minutes of actual start of this event from Start Slot time. The exact start time of an event can be calculated with one minute precision by adding the start_minute_offset value to the start time as indicated by the start_slot value * 30 minutes.

length_in_slots — This 8-bit unsigned integer field indicates the number of 30 minute slots this event occupies (0-47 inclusive). Refer to the description of the end_minute_offset field for the use of this field.

end_minute_offset — This 8-bit unsigned integer field indicates the offset in minutes of the actual end of the event from the end slot. The exact end time of this event can be calculated with one minute precision by adding the end_minute_offset value to (start_slot value + length_in_slots value)* 30 minutes.

category — This 4-bit unsigned integer field (0-15 inclusive) indicates the theme category level of this event. Default categories are defined in Annex C. If default category labels defined in Annex C are not used, new labels may be defined in DOR records.

subcategory — This 4-bit unsigned integer field (0-15 inclusive) indicates the theme subcategory of theme category for this event. Defaults are defined in Annex C. If default subcategory labels defined in Annex C are not used, new labels may be defined in DOR records.

rating — This 4-bit unsigned integer field (0-15 inclusive) indicates the MPAA rating level of this event. Default rating levels are defined in Annex C. If default rating labels defined in Annex C are not used, new rating labels may be defined in DOR (Section 5.2.3).

content_advisory — This 4-bit unsigned integer field (0-15 inclusive) indicates the EIA-608 content advisory of this event (Annex C). If default content advisory labels as defined in Annex C are not used, new labels may be defined in DOR (Section 5.2.3).

violence_rating — This 4-bit unsigned integer field (0-15 inclusive) indicates the EIA-608 violence rating of this event (Annex C). If default violence rating labels as defined in Annex C are not used, new labels may be defined in DOR (Section 5.2.3).

language_rating — This 4-bit unsigned integer field (0-15 inclusive) indicates the EIA-608 language rating of this event (Annex C). If default language rating labels as defined in Annex C are not used, new labels may be defined in DOR (Section 5.2.3).

sex_rating — This 4-bit unsigned integer field (0-15 inclusive) indicates the EIA-608 sex rating of this event (Annex C). If default sex rating labels as defined in Annex C are not used, new labels may be defined in DOR (Section 5.2.3).

number_start_times — This 8-bit unsigned integer field indicates the number of starting times for this event listed in the start_list.

start_slot[i] — This 8-bit unsigned integer field indicates the starting position relative to the start_day field in terms of 30 minute slots. This number may be greater than 47.

start_minute_offset[i] — This 8-bit unsigned integer field indicates the offset of actual start of the repeated event in minutes from the start slot position indicated in the previous field.

length_in_slots[i] — This 4-bit unsigned integer field indicates the number of 30 minute slots this repeated event occupies (0-15 inclusive).

repeat_event_PIP_flag[i] — 2 bits which indicate how to construct the PIP_ID for this repeating event (see Section 7.3.4).

If repeat_event_PIP_flag = 0x0 then there is no PIP for this repeated event.

If repeat_event_PIP_flag = 0x1 then the PIP_ID for this repeated event is constructed implicitly using the actual start time as described in Section 7.3.4.

If repeat_event_PIP_flag = 0x2 then the PIP_ID for this repeated event is constructed explicitly using the repeat_event_PIP_message_number[i] field below.

If repeat_event_PIP_flag = 0x3 then the PIP_ID for this repeated event is constructed explicitly using the event_PIP_message_number below.

repeat_event_PIP_message_number[i] — Explicit PIP message number. If value of repeat_event_PIP_flag = 0x2 then this 16-bit field is used to construct the PIP_ID for this repeated event (see Section 7.3.4).

program_event_ID — This 24-bit unsigned integer field is assigned by the program provider as indicated by the provider_index field in the CIT. The provider_index and the program_event_ID can be used in conjunction with the program_event_descriptor described in the standard or recommended practice developed from ATSC document T3/384.

event_PIP_message_number — Explicit PIP message number. If event_PIP_flag = 0x2 then this 16-bit field is used to construct the PIP_ID for this event (see Section 7.3.4).

event_DIP_message_number — Explicit DIP message number. If event_DIP_flag = 0x2 then this 16-bit field is used to construct the PIP_ID for this event (see Section 7.3.3).

compressed_title() — This is a multiple compressed string which represents the title for the programming event in possibly multiple languages. The event title shall be less than 81 characters before compression (see Section 5.5). If necessary, the title may be truncated to fit allocated display space.

CRC_32 — This is a 32-bit field that contains the CRC value that ensures a zero output from the registers in the decoder defined in Annex B of ISO-13818-1 "MPEG-2 Systems" after processing the entire Transport Stream EIT section.

5.5 Multiple Compressed Strings

This is a general construction that is used for text such as the DOR texts in AGDT, the programming event names in the EITs and the description text in the DIPs.

number_strings — This 8-bit unsigned integer field identifies the number of strings in the following data.

number_bytes — This 8-bit unsigned integer field identifies the number of bytes in the code for string i.

Table 5.9 Syntax for Multiple Compressed Strings

Syntax	Bits	Format
multiple_compressed_strings () {		
number_strings	8	uimsbf
for (i = 0;i< number_strings;i++) {		
number_bytes	8	uimsbf
compression_type	8	bslbf
language_indicator	8	bslbf
for (j = 0;j<number_bytes;j++) {		
compressed_string_byte [j]	8	bslbf
}		
}		
}		

compression_type— This 8-bit field identifies the compression type for string i.

compression_type =0x0: no compression, just ISO Latin-1 codes.

All other values are reserved for future use.

language_indicator — Language indicator for string i. The Language code is the SType for audio for the language (Annex B).

compressed_string_byte[j]— j-th byte of the i-th string

6. SPECIAL PROGRAM GUIDE (SPG)

In addition to the required Master Program Guide (MPG), the program guide can support 15 optional Special Program Guides (SPG). Each additional special guide is a single separate data stream appearing on one exclusive SPG_PID, as described in the Program Guide Map in the MPG.

The channel groupings that apply to the MPG also apply to the special guides. In other words, the channel groupings are defined in the Master Guide Table of the MPG and apply to all guides. However, SPGs may have a different number of channels than the MPG. Each channel must be described within the specific channel grouping whose range includes the channel number.

Some SPGs may be extensions to the MPG and are automatically referenced using the "next" field in the Program Guide Map. In this case, SPGs may be used to describe programming events in the future, beyond the coverage of the MPG. Multiple SPGs can be linked through the "next" and "previous" fields in the Program Guide Map. For example, if the MPG covers the programming events for the next 6 hours, SPG 1 may be used to cover program guide information for programs in the period +6 to +20 hours and SPG 2 may cover programs from +20 hours to +2 days.

However, SPGs do not necessarily have to be linked to the MPG. They may as well have the same time span as the MPG. In any case, SPGs are defined in the Program Guide Map of the MPG and have to obey the channel grouping structure defined in Master Guide Table of the MPG.

The formatting of these data streams is similar to the MPG guide stream, except that the SPGs consist of only CITs and EITs. The Master Guide Table and Additional Guide Data Table are not allowed in SPGs.

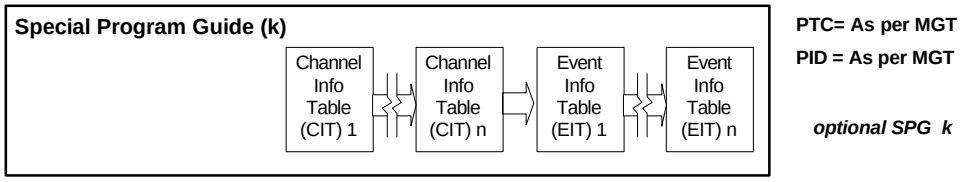

Figure 6.1. General structure of the Special Program Guide.

6.1 Special Program Guide Channel Information Tables

The SPG CIT sections have the same syntax and semantics as the MPG CIT sections defined in Section 5.3. The SPG CIT sections are carried on physical channel SPG_PTC with PID values SPG_PID as defined in the Program Guide Map. Each SPG_PID is exclusively reserved for one specific SPG stream.

The following constraints apply to the Transport packets carrying the CIT sections in SPGs:

- PID shall have the value SPG_PID as defined in the PGM.
- payload_unit_start_indicator of the transport packet carrying the table_id field of each CIT section shall be 1 (first transport packet of each section)
- pointer_field of the transport packet carrying the table_id field of each CIT section shall have the value 0x00 (first transport packet of each section)
- the remainder of the transport packet carrying the CRC_32 field of the CIT section shall be padded with stuffing bytes of value 0xFF (last transport packet of the section)

6.2 Special Program Guide Event Information Tables

The SPG EIT sections have same syntax as the MPG EIT sections defined in Section 5.4. The SPG EIT sections are carried on physical channel SPG_PTC with PID values SPG_PID as defined in the Program Guide Map.

The following constraints apply to the Transport packets carrying the EIT sections in SPG:

- PID shall have the value SPG_PID as defined in the PGM
- payload_unit_start_indicator of the transport packet carrying the table_id field of the EIT section shall be 1 (first transport packet of every section)
- pointer_field of the transport packet carrying the table_id field of each EIT section shall have the value 0x00 (first transport packet of every section)
- the remainder of the transport packet carrying the CRC_32 field of the EIT section shall be padded with stuffing bytes of value 0xFF (last transport packet of the section)

The semantics of SPG EIT sections are the same as MPG EIT sections with the following exceptions:

section_number — This 8-bit field gives the number of this SPG CIT section. The section number of the first section in the SPG EIT section shall be 0x00. The section_number shall be incremented by 1 modulo 256 with each additional section (EIT info for one channel in SPG may be conveyed in more than one section).

last_section_number — This 8-bit field specifies the number of the last section of the SPG EIT info for this channel (EIT info for one channel in SPG may be conveyed in more than one section).

7. INFORMATION PARCEL STREAMS

Information Parcels are a means of transmitting large volumes of extra data information which is not easily acquired in table format. Information Parcels are typically used to transmit data about content details and/or purchases.

7.1 Description Information Parcel (DIP)

The Description Information Parcel (DIP) streams are optional and are used to provide detailed descriptions of channels and program events. A DIP is a multiple compressed string data structure (Section 5.5), and so, may represent a description in several different languages (each string corresponding to one language). If necessary, the description may be truncated to fit allocated display space.

Within a transport stream, the Description Information Parcel is carried on a private section with table ID 0xE4. Each event description is distinguished by its unique 32-bit DIP_ID at the beginning of the section. This allows the receiver to search for a single description quickly without having to parse into the payload of a large table.

The DIP sections for a channel or for programming events are carried on the physical channel DIP_PTC (as identified in CIT section) with PID values DIP_PID as defined in the Master Guide Table of the MPG. This specific PID is exclusively reserved for the DIP stream.

The following constraints apply to the Transport packets carrying the DIP sections:

- PID shall have the value DIP_PID as defined in the MGT
- payload_unit_start_indicator of the transport packet carrying the table_id field of each DIP section shall be 1 (first transport packet of each section)
- pointer_field of the transport packet carrying the table_id field of each DIP section shall have the value 0x00 (first transport packet of each section)

Table 7.1 Bit Stream Syntax for the Description Information Parcel

Syntax	Bits	Format
description_information_parcel_section () {		
table_id	8	0xE4
section_syntax_indicator	1	'0'
private_indicator	1	'1'
reserved	2	'11'
private_section_length	12	uimsbf
DIP_ID	32	bslbf
description () {		
multiple_compressed_strings ()		
}		
}		

table_id — Identifies this section as belonging to a Description Information Parcel. (0xE4)

section_syntax_indicator — This 1-bit flag shall be set to '0', denoting that DIP data bytes immediately follow the private_section_length field.

private_indicator— This 1-bit flag shall be set to '1'.

private_section_length — 12-bit field specifying the number of remaining bytes in the section immediately following the section_length field up to the end of the section.

DIP_ID — Unique 32-bit identifier of this description information parcel. The construction of this 32-bit identifier is described in Section 7.3.

description() — This is multiple compressed strings representing a detailed description of an event or channel possibly in multiple languages.

7.2 Private Information Parcel (PIP)

The PIP data stream is used to transmit Private Information Parcels (PIPs) that contain user private information for channels or events. The packetization and acquisition is similar to the description stream.

Within a transport stream, the Private Information Parcel is carried in a private section with table ID 0xE5. Each event description is distinguished by its unique 32-bit PIP_ID at the beginning of the section. This allows the receiver to search for a single private descriptor quickly. The content of the private_information field in the PIP_PID is defined by the user as identified by registration_descriptor and the private_data_indicator fields in the MGT.

The PIP sections for a channel or events on a channel are carried on physical channel PIP_PTC (as identified in the CIT section) with PID values PIP_PID as defined in the Master Guide Table of the MPG. This specific PID is exclusively reserved for the PIP stream.

The following constraints apply to the Transport packets carrying the PIP sections:

- PID shall have the value PIP_PID as defined in the MGT
- payload_unit_start_indicator of the transport packet carrying the table_id field of each PIP section shall be 1 (first transport packet of each section).
- pointer_field of the transport packet carrying the table_id field of each PIP section shall have the value 0x00 (first transport packet of each section).

table_id — Identifies this section as belonging to a Private Information Table (0xE5)

section_syntax_indicator — This 1-bit flag shall be set to '0', denoting that PIP data bytes immediately follow the private_section_length field.

private_indicator— This 1-bit flag shall be set to '1'.

private_section_length — 12-bit field specifying the number of remaining bytes in the section immediately following the section_length field up to the end of the section.

PIP_ID — A unique 32-bit identifier of this private information parcel. The construction of this 32-bit identifier is described in Section 7.3.

private_information() — Contains user private information that is not defined by this document. The content and structure of the private_information field is defined by the user as identified by registration_descriptor and the private_data_indicator fields in the MGT.

Table 7.2 Bit Stream Syntax for the Private Information Parcel

Syntax	Bits	Format
private_information_parcel_section () {		
table_id	8	0xE5
section_syntax_indicator	1	'0'
private_indicator	1	'1'
reserved	2	'11'
private_section_length	12	uimsbf
PIP_ID	32	bslbf
private_information()		
}		

7.3 Information Parcel Identifiers (PIP_ID or DIP_ID)

As explained above, Private Information Parcels (PIPs) and Description Information Parcels (DIPs) are acquired using 32-bit unique ID's. These 32-bit unique IDs consist of several data fields and may either be constructed from an explicit message number, or constructed implicitly from the channel number and event start time. The choice of implicit or explicit construction of the ID is indicated by the value of the channel_DIP_flag or channel_PIP_flag in the CIT section for a channel and by the value of the event_DIP_flag, event_PIP_flag or repeat_event_PIP_flag in the EIT section for an event.

Implicitly constructed ID's are used whenever a program has unique private and/or description information. Explicitly constructed ID's are needed if multiple programs share the same private and/or description information (NVOD applications, time shifted presentation of the same event in different channels, etc.). Note that the PIP_ID fields are necessary to identify the Information Parcel, but do not define the contents of the Private Information. The content and use of the private_information field is defined by the user as identified by the private_data_indicator field in the MGT.

7.3.1 Channel DIP_ID

parcel_type — This 1-bit flag indicates the type of data in the parcel; It shall be set to '1', to indicate the parcel as DIP information.

channel_number — The value of this 10-bit field represents the number of the channel represented (1 through 999 inclusive).

message_number— 16-bit channel_DIP_message_number (defined in the CIT section).

Table 7.3 Syntax of the 32-Bit Channel DIP_ID

Syntax	Bits	Format
channel_DIP_parcel_identifier () {		
reserved	1	'1'
parcel_type	1	bslbf
if (channel_DIP_flag=='01') implicit(){		
channel_number	10	uimsbf
zero_bits	4	'0000'
zero_bits	16	0x00
}		
else if (channel_DIP_flag=='10') explicit(){		
zero_bits	10	0x00
zero_bits	4	'0000'
message_number	16	uimsbf
}		
}		

7.3.2 Channel PIP_ID

Table 7.4 Syntax of the 32-Bit Channel PIP_ID

Syntax	Bits	Format
channel_PIP_parcel_identifier () {		
reserved	1	'1'
parcel_type	1	bslbf
if (channel_PIP_flag=='01') implicit(){		
channel_number	10	uimsbf
zero_bits	4	'0000'
zero_bits	16	0x00
}		
else if (channel_PIP_flag=='10') explicit(){		
zero_bits	10	0x00
zero_bits	4	'0000'
message_number	16	uimsbf
}		
}		

parcel_type — This 1-bit flag indicates the type of data in the parcel; It shall be set to '0', to indicate the parcel as PIP information.

channel_number — The value of this 10-bit field represents the number of the channel represented (1 through 999 inclusive).

message_number— 16-bit channel_PIP_message_number (defined in CIT section).

7.3.3 Event DIP_ID

parcel_type — This 1-bit flag indicates the type of data in the parcel; It shall be set to '1', to indicate the parcel as DIP information.

Table 7.5 Syntax of the 32-Bit Event_DIP_ID

Syntax	Bits	Format
event_DIP_parcel_identifier () {		
reserved	1	'1'
parcel_type	1	bslbf
if (event_DIP_flag=='01') implicit(){		
channel_number	10	uimsbf
start_month	4	uimsbf
start_day_of_month	5	uimsbf
start_hour	5	uimsbf
start_minute	6	uimsbf
}		
else if (event_DIP_flag=='10') explicit(){		
zero_bits	10	0x00
zero_bits	4	'0000'
message_number	16	uimsbf
}		
}		

channel_number — The value of this 10-bit field represents the number of the channel represented (1 through 999 inclusive).

start_month — This 4-bit unsigned integer field (1: January-12: December) indicates the month of start of this programming event (defined in the EIT section).

start_day_of_month — This 5-bit unsigned integer field (1-31) indicates the absolute day of month of start of program event (defined in the EIT section).

start_hour — The value of this 5-bit field is calculated as follows: start time is computed by multiplying the start_slot field (in the EIT section) value by 30 and adding the start_minute_offset field (in the EIT section) value to get minutes since midnight. This value is divided by 60 to get value of start_hour field.

start_minute — The value of this 6-bit field is calculated as follows: start time is computed by multiplying the start_slot field (in the EIT section) value by 30 and adding the start_minute_offset field (in the EIT section) value to get minutes since midnight. This value is divided by 60 and the remainder is the value of the start_minute field.

message_number— 16-bit event_DIP_message_number (defined in the EIT section).

7.3.4 Event PIP_ID

parcel_type — This 1-bit flag indicates the type of data in the parcel; It shall be set to '0', to indicate the parcel as PIP information.

channel_number — The value of this 10-bit field represents the number of the channel represented (1 through 999 inclusive).

start_month — This 4-bit unsigned integer field (1: January-12: December) indicates the month of start of this programming event (defined in the EIT section).

Table 7.6 Syntax of the 32-Bit Event_PIP_ID

Syntax	Bits	Format
event_PIP_parcel_identifier () {		
reserved	1	'1'
parcel_type	1	bslbf
if (event_PIP_flag=='01') implicit(){		
channel_number	10	uimsbf
start_month	4	uimsbf
start_day_of_month	5	uimsbf
start_hour	5	uimsbf
start_minute	6	uimsbf
}		
else if (event_PIP_flag=='10') explicit(){		
zero_bits	10	0x00
zero_bits	4	'0000'
message_number	16	uimsbf
}		
}		

start_day_of_month — This 5-bit unsigned integer field (1-31) indicates the absolute day of month of start of program event (defined in the EIT section).

start_hour — The value of this 5-bit field is calculated as follows: start time is computed by multiplying the start_slot field (in the EIT section) value by 30 and adding the start_minute_offset field (in the EIT section) value to get minutes since midnight. This value is divided by 60 to get value of start_hours field.

start_minute — The value of this 6-bit field is calculated as follows: start time is computed by multiplying the start_slot field (in the EIT section) value by 30 and adding the start_minute_offset field (in the EIT section) value to get minutes since midnight. This value is divided by 60 and the remainder is the value of the start_minute field.

For repeated events, as described in the EIT section with event_repetition_flag value 1, the start_hour and start_minute fields are calculated using the start_slot[i] and start_minute_offset[i] fields.

message_number — 16-bit event_PIP_message_number (defined in the EIT section). For repeated events, as described in the EIT section with event_repetition_flag value 1, either the event_PIP_message_number or the repeat_event_PIP_message_number[i] is used as determined through the value of the repeat_event_PIP_flag[i].

8. TRANSMISSION REQUIREMENTS

This Program Guide is a collection of separate data streams that are conveyed using the private section format (stream_type=0x05) of the "ISO/IEC 13818-1 Information Technology — Generic Coding of Moving Pictures and Associated Audio", MPEG-2 system level transport packets.

This section defines the constraints imposed on the program guide structure sizes, as well as transmission rates of these structures.

8.1 Size of MPG and SPG Structures

This section defines the constraints imposed on the program guide structure sizes. The goal is to define the resources needed to implement the program guide in a receiver, and guarantee fast enough cycle time for the guide structures in the transport stream.

For over-the-air broadcast applications the following constraints apply:

- **For MPG**: The total size of the MPG stream, as indicated by the Num_pkts_MPG field in the Master Guide Table, depends on the number of distinct data multiplexes (physical channels) contained in its description, and shall be equal or less than the values defined in Table 8.1.

Table 8.1 Maximum Values for Num_pkts_MPG for MPG

Number of data multiplexes	Value of Num_pkts_MPG
1	35
2	70
3	105
4	140
5	175
6	210
7	245
8	280
9	315
10+	350

- **For any SPG**: The total number of the transport packets carrying the MGT and the AGDT, and all CIT sections of one SPG and all EIT sections of one channel grouping in that SPG shall be equal to or less than 100. In terms of Program Guide fields:

  1+ Num_pkts_AGDT + Num_pkts_SPG[l]_EIT[i] +
  Sum(Num_pkts_SPG[l]_CIT[j]: j=1 to number_channel_groupings) <=100, for all i and l

  where Num_pkts_AGDT is defined in Master Guide Table, and
  Num_pkts_SPG[l]_EIT[i] and Num_pkts_SPG[l]_CIT[j] are defined in Program Guide Map.

Note that the amount of data sent, and the rate at which the data is sent determines the cycle time of the data, i.e., how often can a user capture whole data structure starting from scratch. For example, if a broadcaster has only one physical transmission channel, the maximum amount of MPG data that can be sent is (from num_pkts_MPG=35):

==>35 * 1472 (bits/packet payload)= 51,520 bits

Therefore, the cycle time of the MPG would be 51,520/Rate. If it is decided to send the MPG at the maximum allowed data rate of 100,000 bps, the cycle time for the MPG would be 516 ms.

The same considerations should be applied to the DIP and PIP streams. Although each DIP or PIP section cannot be longer than 4096 bytes, the number of different DIP or PIP sections and their distribution to different channels (if possible) affects the cycle time of the DIP and PIP streams.

8.2 STD Model for the Program Guide Streams

The program guide data streams use the private section format (stream_type =0x05) of the "ISO/IEC 13818-1 Information Technology — Generic Coding of Moving Pictures and Associated Audio", MPEG-2 System.

For over-the-air broadcast applications the following constraints apply:

- In over-the-air broadcast applications, each of the Program Guide elementary streams as identified by transport packets in PIDs 0x1FFD (MPG), SPG_PID (for each separate SPG), DIP_PID (for each DIP) and PIP_PID (for each PIP), shall adhere independently to an STD model that can be described by an MPEG-2 Smoothing buffer descriptor (Section 2.6.30 in ISO/IEC 13818-1) with the following parameters:

  sb_leak_rate shall be 250 (indicating a leak rate of 100,000 bps)

  sb_size shall be 1024 (indicating a smoothing buffer size of 1024 bytes)

8.3 Program Guide Extensibility

This Program Guide standard can be extended by defining additional tables. Table_IDs 0XE6 through 0xEF are reserved by ATSC for this purpose.

ANNEX A

(Normative)

DAYLIGHT SAVINGS TIME CONTROL

In order to convert Program Guide time (GMT) into local time, the receiver needs to store in local memory a time offset (from GMT to local time) and an indicator as to whether daylight savings is observed. These two quantities can be obtained from a user interface (indicating time zone and daylight savings observance) or from the conditional access system and stored in non-volatile receiver memory.

Since there is a common time (GMT) transmitted in the Program Guide, there needs to be mechanism to indicate when the receiver should switch into (or out of) daylight savings time at the appropriate local time. Once all the receivers have transitioned at their local times, the entire system can be shifted into daylight savings time. This is accomplished by appropriate setting of the two DS bytes in the MGT.

Table 1 Structure of Daylight Savings Time Control

Syntax	Bits	Format
daylight_savings () {		
DS_status	1	bslbf
reserved	2	'11'
DS_day_of_month	5	uimsbf
DS_hour	8	uimsbf
}		

DS_status — This bit indicate the status of daylight savings.

DS_status = '0': Not in daylight savings time.

DS_status = '1': In daylight savings time.

DS_day_of_month — This 5-bit unsigned integer field indicates the local day of month number on which transition into or out of daylight savings time is to occur (1-31).

DS_hour — This 8-bit unsigned integer field indicates the local hour at which transition into or out of daylight savings time is to occur (0-18). This usually occurs at 2 AM.

Table 2 Basic Use of Daylight Savings Fields Through the Year

Conditions	DS status	DS_day of_month	DS_hour
At the beginning of the year (January) daylight savings is off. This is the status of the fields until:	0	0	0
When transition into daylight savings time is within less than one month, DS_day_of_month field takes the value day_in , and DS_hour field takes the value hour_in. DS_status bit is 0 indicating it is not daylight savings time yet. (Transition is to occur on day_in day of the month at hour=hour_in, for example, if the transition were on April 15 at 2 AM, then day_in=15 and hour_in=2)	0	day_in	hour_in
After all time zone daylight transitions (within network span) have occurred, DS_status bit takes value 1, indicating daylight savings time is on. DS_day_of_month field and DS_hour field takes the values 0. (This transition has to occur no later than 7 PM Pacific Time on the day day_in). This is the status of the fields until:	1	0	0
When transition out of daylight savings time is within less than one month, DS_day_of_month field takes the value day_out , and DS_hour field takes the value hour_out. DS_status bit is 1 indicating it is still daylight savings time. (Transition is to occur on day_out day of the month at hour=hour_out, for example, if the transition were on October 27 at 2 AM, then day_out=27 and hour_out=2)	1	day_out	hour_out
After all time zones (within network span) have shifted out of daylight savings time, DS_status bit takes value 0, indicating daylight savings time is off. DS_day_of_month field and DS_hour field takes the values 0. (This transition has to occur no later than 7 PM Pacific Time on the day day_out). This finishes the cycle.	0	0	0

ANNEX B

(Normative)

SERVICE PARADIGM LIST

The Service Paradigm Indicator (SPI) field in the CIT determines an implicit PID list according to the following standard formats.

This standard format of the PID list allows the service to use a range of 15 PIDs. The description of the SPI number determines which PIDs are in use and the type of data within each stream that is in use.

The STypes field (8-bit unsigned integer) enters this equation and the following version of STypes is used:

Table 1 List of STypes

SType	Label	Function
0	English	audio
1	Spanish	audio
2	French	audio
3	Portuguese	audio
4	German	audio
5	Italian	audio
6	Japanese	audio
7	Korean	audio
8	Chinese (Mandarin)	audio
9	ATSC reserved	data
10	ATSC reserved	data
11	ATSC reserved	data
12	session data	data
13	continuous data	data
14	high speed data	data
15	video	video
16-63	ATSC reserved	ATSC reserved
64-255	User Private	User Private

Table 2 List of Service Paradigms

SPI_indicator	General	Specific PID list
0x0	Explicit PID list.	Use the explicit PID list that appears in the CIT section
0x1	ATSC Program Paradigm, main audio in English	PID+1 = video (Stype=15), PID+4 = audio (Stype=0), PID+10 = data (Stype=12)
0x2	ATSC Program Paradigm, main audio in Spanish	PID+1 = video (Stype=15), PID+4 = audio (Stype=1), PID+10 = data (Stype=12)
0x3	ATSC Program Paradigm, main audio in French	PID+1 = video (Stype=15), PID+4 = audio (Stype=2), PID+10 = data (Stype=12)
0x4	ATSC Program Paradigm, main audio in Portuguese	PID+1 = video (Stype=15), PID+4 = audio (Stype=3), PID+10 = data (Stype=12)
0x5	ATSC Program Paradigm, main audio in German	PID+1 = video (Stype=15), PID+4 = audio (Stype=4), PID+10 = data (Stype=12)
0x6	ATSC Program Paradigm, main audio in Italian	PID+1 = video (Stype=15), PID+4 = audio (Stype=5), PID+10 = data (Stype=12)
0x7	ATSC Program Paradigm, main audio in Japanese	PID+1 = video (Stype=15), PID+4 = audio (Stype=6), PID+10 = data (Stype=12)
0x8	ATSC Program Paradigm, main audio in Korean	PID+1 = video (Stype=15), PID+4 = audio (Stype=7), PID+10 = data (Stype=12)
0x9	ATSC Program Paradigm, main audio in Chinese	PID+1 = video (Stype=15), PID+4 = audio (Stype=8), PID+10 = data (Stype=12)
0xA	General (SType = 0) television. Video with maybe more than 1 audio	PID = video (Stype=15), PID+1 = audio (Stype=0), PID+2 = audio (Stype=0),...,PID+NPID-1 (Stype=0).
0xB	Language Preference TV. Video with alternate languages.	PID= video(Stype=15), PID+1=audio (Stype=0), PID+2=audio(Stype=1), PID+3 = audio(SType=2) ..., PID+NPID-1 = audio (Stype=NPID-1). Default order is English, Spanish, French, Portuguese, German, Italian...
0xC	Language Preference TV. Video with alternate languages (permutation 1)	PID= video(Stype=15), PID+1=audio (Stype=1), PID+2=audio(Stype=0), PID+3 = audio(SType=2) ..., PID+NPID-1 = audio (Stype=NPID-1). Default order is Spanish, English, French, Portuguese, German, Italian...
0xD	Language Preference TV. Video with alternate languages (permutation 2).	PID= video(Stype=15), PID+1=audio (Stype=2), PID+2=audio(Stype=0), PID+3 = audio(SType=1) ..., PID+NPID-1 = audio (Stype=NPID-1). Default order is French, English, Spanish, Portuguese, German, Italian...
0xE	Language Preference TV. Video with alternate languages (permutation 3).	PID= video(Stype=15), PID+1=audio (Stype=3), PID+2=audio(Stype=0), PID+3 = audio(SType=1) ..., PID+NPID-1 = audio (Stype=NPID-1). Default order is Portuguese, English, Spanish, French, German, Italian...
0xF	Language Preference TV. Video with alternate languages (permutation 4).	PID= video(Stype=15), PID+1=audio (Stype=4), PID+2=audio(Stype=0), PID+3 = audio(SType=1) ..., PID+NPID-1 = audio (Stype=NPID-1). Default order is German, English, Spanish, Portuguese, French, Italian...
0x10	General (SType = 0) television. Video with maybe more than 1 audio and Continuous Data	PID = video (Stype=15), PID+1 = audio (Stype=0), PID+2 = audio (Stype=0),...,PID+NPID-2 (Stype=0) ,PID+NPID-1 (Stype=13).
0x11	Language Preference TV. Video with alternate languages, and Continuous Data.	PID= video(Stype=15), PID+1=audio (Stype=0), PID+2=audio(Stype=1), PID+3 = audio(SType=2) ..., PID+NPID-1 = data (Stype=13). Default order is English, Spanish, French, Portuguese, German, Italian...

SPI_indicator	General	Specific PID list
0x12	Language Preference TV. Video with alternate languages (permutation 1), and Continuous Data.	PID= video(Stype=15), PID+1=audio (Stype=1), PID+2=audio(Stype=0), PID+3 = audio(SType=2) ..., PID+NPID-1 = data (Stype=13). Default order is Spanish, English, French, Portuguese, German, Italian...
0x13	Language Preference TV. Video with alternate languages (permutation 2), and Continuous Data.	PID= video(Stype=15), PID+1=audio (Stype=2), PID+2=audio(Stype=0), PID+3 = audio(SType=1) ..., PID+NPID-1 = data (Stype=13). Default order is French, English, Spanish, Portuguese, German, Italian...
0x14	Language Preference TV. Video with alternate languages (permutation 3) and Continuous Data.	PID= video(Stype=15), PID+1=audio (Stype=3), PID+2=audio(Stype=0), PID+3 = audio(SType=1) ..., PID+NPID-1 = 13 (Stype=13). Default order is Portuguese, English, Spanish, French, German, Italian...
0x15	Subscription Audio.	PID, PID+1, ..., PID+NPID-1, all Stype=0.
0x16	Subscription Continuous Data.	PID,PID+1,... , PID+NPID-1, all Stype=13.
0x17	Subscription Session Data.	PID, PID+1, ... , PID+NPID-1, all Stype=12.
0x18	High Speed data	PID(Stype=14) only one PID
0x19-0x3F	ATSC reserved	ATSC reserved
0x40-0xFF	User Private	User Private

ANNEX C

(Normative)

DEFAULT LABELS

The following program content ratings are based on the MPAA and the XDS Program Rating Specification of ANSI/EIA-608-94. Some space between categories is provided for future additions.

Table 1 Default Ratings Labels

level	rating
0	N/A
1	reserved
2	G
3	reserved
4	PG
5	reserved
6	PG-13
7	reserved
8	R
9	reserved
10	NC-17
11	reserved
12	X
13	reserved
14	Not Rated
15	reserved

Table 2 Default Content Advisory Labels

level	content advisory
0	Does not apply
1-15	reserved

Table 3 Default Violence Labels

level	violence
0	No violent content
1-15	reserved

Table 4 Default Language Labels

level	language
0	No adult language
1-15	reserved

Table 5 Default Sexual Content Labels

level	sex
0	No sexual content
1-15	reserved

Table 6 Default Category Labels

level	label
0	Movies
1	Sports
2	Specials
3	Series
4	News
5	Shopping
6	reserved
7	reserved
8	reserved
9	reserved
10	reserved
11	reserved
12	reserved
13	reserved
14	reserved
15	reserved

Table 7 Default Subcategory Labels

Level	Movies Labels	Sports Labels	Specials Labels	Series Labels	News Labels	Shopping Labels
0	Action/ Adv	Baseball	Children	Children	Business	Home Shopping
1	Children	Basketball	Comedy	Comedy	Current	Product Info.
2	Comedy	Boxing	Cultural Arts	Cultural Arts	Gen. Interest	reserved
3	Drama	Football	Drama	Drama	Sports	reserved
4	Fantasy	Golf	Educational	Educational	Weather	reserved
5	Horror	Hockey	Gen. Interest	Gen. Interest	reserved	reserved
6	Musical	Racing	How-To	How-To	reserved	reserved
7	Romance	Skiing	Mature	Mature	reserved	reserved
8	Sci-Fi	Soccer	Music	Music	reserved	reserved
9	Western	Tennis	Religious	Religious	reserved	reserved
10	Other	Wrestling	Soap Opera	Soap Opera	reserved	reserved
11	reserved	Other	Talk	Talk	reserved	reserved
12	reserved	reserved	reserved	reserved	reserved	reserved
13	reserved	reserved	reserved	reserved	reserved	reserved
14	reserved	reserved	reserved	reserved	reserved	reserved
15	reserved	reserved	reserved	reserved	reserved	reserved

Blank Page