

Additional information may be carried by descriptors which may be placed in the descriptor loop after the basic information.

The Virtual Channel Table may be segmented into as many as 256 sections. One section may contain information for several virtual channels, but the information for one virtual channel shall not be segmented and put into two or more sections. Thus for each section, the first field after `protocol_version` shall be `num_channels_in_section`.

6.3.1 Terrestrial Virtual Channel Table

The Terrestrial Virtual Channel Table is carried in private sections with table ID 0xC8, 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 Stream packets carrying the VCT sections:

- PID for Terrestrial VCT shall have the value 0x1FFB (`base_PID`)
- `transport_scrambling_control` bits shall have the value '00'
- `adaptation_field_control` bits shall have the value '01'

The bit stream syntax for the Terrestrial Virtual Channel Table is shown in Table 6.4.

table_id — An 8-bit unsigned integer number that indicates the type of table section being defined here. For the `terrestrial_virtual_channel_table_section()`, the `table_id` shall be 0xC8.

section_syntax_indicator — The `section_syntax_indicator` is a one-bit field which shall be set to '1' for the `terrestrial_virtual_channel_table_section()`.

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

section_length — This is a twelve bit field, the first two bits of which shall be '00'. It specifies the number of bytes of the section, starting immediately following the `section_length` field, and including the CRC. The value in this field shall not exceed 1021.

transport_stream_id — The 16-bit MPEG-2 Transport Stream ID, as it appears in the Program Association Table (PAT) identified by a PID value of zero for this multiplex. The `transport_stream_id` distinguishes this Terrestrial Virtual Channel Table from others that may be broadcast in different PTCs.

Table 6.4 Bit Stream Syntax for the Terrestrial Virtual Channel Table

Syntax	Bits	Format
terrestrial_virtual_channel_table_section () {		
table_id	8	0xC8
section_syntax_indicator	1	'1'
private_indicator	1	'1'
zero	2	'00'
section_length	12	uimsbf
transport_stream_id	16	uimsbf
reserved	2	'11'
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
protocol_version	8	uimsbf
num_channels_in_section	8	uimsbf
for(i=0; i<num_channels_in_section;i++) {		
short_name	7*16	unicode™ BMP
reserved	4	'1111'
major_channel_number	10	uimsbf
minor_channel_number	10	uimsbf
modulation_mode	8	uimsbf
carrier_frequency	32	uimsbf
channel_TSID	16	uimsbf
program_number	16	uimsbf
ETM_location	2	uimsbf
access_controlled	1	bslbf
hidden	1	bslbf
reserved	6	'111111'
service_type	6	uimsbf
source_id	16	uimsbf
reserved	6	'111111'
descriptors_length	10	uimsbf
for (i=0;i<N;i++) {		
descriptors()		
}		
}		
reserved	6	'111111'
additional_descriptors_length	10	uimsbf
for(j=0; j<N;j++) {		
additional_descriptors()		
}		
CRC_32	32	rpchof
}		

version_number— This 5 bit field is the version number of the Virtual Channel Table. For the current VCT (current_next_indicator = 1), the version number shall be incremented by 1 whenever the definition of the current VCT changes. Upon reaching the value 31, it wraps around to 0. For the next VCT (current_next_indicator = 0), the version number shall be one unit more than that of the current VCT (also in modulo 32 arithmetic). In any case, the value of the version_number shall be identical to that of the corresponding entries in the MGT.

current_next_indicator— A one-bit indicator, which when set to '1' indicates that the Virtual Channel Table sent is currently applicable. When the bit is set to '0', it indicates that the table sent is not yet applicable and shall be the next table to become valid.

section_number— This 8 bit field gives the number of this section. The **section_number** of the first section in the Terrestrial Virtual Channel Table shall be 0x00. It shall be incremented by one with each additional section in the Terrestrial Virtual Channel Table.

last_section_number— This 8 bit field specifies the number of the last section (that is, the section with the highest **section_number**) of the complete Terrestrial Virtual Channel Table.

protocol_version — An 8-bit unsigned integer field whose function is to allow, in the future, this table type to carry parameters that may be structured differently than those defined in the current protocol. At present, the only valid value for **protocol_version** is zero. Non-zero values of **protocol_version** may only be processed by decoders designed to accommodate the later versions as they become standardized.

num_channels_in_section— This 8 bit field specifies the number of virtual channels in this VCT section. The number is limited by the section length.

short_name— The name of the virtual channel, represented as a sequence of one to seven 16-bit character codes coded in accordance with the Basic Multilingual Plane (BMP) of Unicode™, as specified in ISO 10646-1. If the name of the virtual channel is shorter than seven Unicode™ characters, one or more instances of the null character value 0x0000 shall be used to pad the string to its fixed 14-byte length.

major_channel_number— A 10-bit number that represents the “major” channel number associated with the virtual channel being defined in this iteration of the “for” loop. Each virtual channel must be associated with a major and a minor channel number. The major channel number, along with the minor channel number, act as the user’s reference number for the virtual channel. The **major_channel_number** shall be between 1 and 99. For **major_channel_number** assignments in the U.S., refer to Annex B.

minor_channel_number— A 10-bit number in the range 0 to 999 that represents the “minor” or “sub-” channel number. This field, together with **major_channel_number**, performs as a two-part channel number, where **minor_channel_number** represents the second or right-hand part of the number. When the **service_type** is analog television, **minor_channel_number** shall be set to 0. Services whose **service_type** is either **ATSC_digital_television** or **ATSC_audio_only** shall use minor numbers between 1 and 99. For other types of services, such as data broadcasting, valid minor virtual channel numbers are between 1 and 999

modulation_mode — An 8-bit unsigned integer number that indicates the modulation mode for the transmitted carrier associated with this virtual channel. Values of **modulation_mode** are defined by this standard in Table 6.5. For digital signals, the standard values for modulation mode (values below 0x80) indicate transport framing structure, channel coding, interleaving, channel modulation, forward error correction, symbol rate, and other transmission-related parameters, by means of a reference to an appropriate standard. Values of **modulation_mode** 0x80 and above are outside the scope of ATSC. These may be used to specify non-standard modulation modes in

private systems. A value of 0x80 for `modulation_mode` indicates that modulation parameters are specified in a private descriptor.

Table 6.5 Modulation Modes

<code>modulation_mode</code>	meaning	terrestrial broadcast	cable
0x00	[Reserved]		
0x01	analog — The virtual channel is modulated using standard analog methods for analog television.		
0x02	SCTE_mode_1 — The virtual channel has a symbol rate of 5.057 Msps, transmitted in accordance with <i>Digital Transmission Standard for Cable Television</i> , Ref. [12] (Mode 1). Typically, mode 1 will be used for 64-QAM.	Not valid	
0x03	SCTE_mode_2 — The virtual channel has a symbol rate of 5.361 Msps, transmitted in accordance with <i>Digital Transmission Standard for Cable Television</i> , Ref. [12] (Mode 2). Typically, mode 2 will be used for 256-QAM.	Not valid	
0x04	ATSC (8 VSB) — The virtual channel uses the 8-VSB modulation method conforming to the ATSC Digital Television Standard.		Not valid
0x05 -0x7F	[Reserved for future use by ATSC]		
0x80	Modulation parameters are defined by a private descriptor		
0x81-0xFF	[User Private]		

carrier_frequency— A 32-bit unsigned integer that represents the carrier frequency associated with the analog or digital transmission associated with this virtual channel, in units of one Hz. For VSB-modulated signals, the given `carrier_frequency` represents the location of the pilot tone; for analog signals, it represents the frequency of the picture carrier. In the case of a digital terrestrial broadcast signal that is transmitted at multiple carrier frequencies (via one or more translators), the `carrier_frequency` may be specified as zero. In such cases, the receiver is expected to associate the Transport Stream identified by the given `transport_stream_id` with the frequency tuned to acquire it.

For the ATSC Digital Television Standard, where the PTC bandwidth is 6 MHz, the pilot tone is located 310 kHz above the lower edge of the physical transmission channel, or 2.690 MHz below the specified center of the band. Similarly, for analog NTSC transmitted in the US, the picture carrier is 1.25 MHz above the lower edge of the 6 MHz physical transmission channel.

channel_TSID— A 16-bit unsigned integer field in the range 0x0000 to 0xFFFF that represents the MPEG-2 Transport Stream ID associated with the Transport Stream carrying the MPEG-2 program referenced by this virtual channel. The receiver may use the `channel_TSID` to verify that a TS acquired at the referenced carrier frequency is actually the desired multiplex. Analog signals may have a TSID provided that it is different from any DTV Transport Stream identifier; that is,

it shall be truly unique if present.² A value of 0xFFFF for channel_TSID shall be specified for analog channels that do not have a valid TSID.

program_number — A 16-bit unsigned integer number that associates the virtual channel being defined here with the MPEG-2 PROGRAM ASSOCIATION and TS PROGRAM MAP tables. For virtual channels representing analog services, a value of 0xFFFF shall be specified for program_number.

ETM_location — This 2-bit field specifies the existence and the location of an Extended Text Message (ETM), based on Table 6.6.

Table 6.6 ETM location

ETM_location	Meaning
0x00	No ETM
0x01	ETM located in the PTC carrying this PSIP
0x02	ETM located in the PTC specified by the channel_TSID
0x03	[Reserved for future ATSC use]

access_controlled — A 1-bit Boolean flag that indicates, when set, that the events associated with this virtual channel may be access controlled. When the flag is set to 0, event access is not restricted.

hidden — A 1-bit Boolean flag that indicates, when set, that the virtual channel is not accessed by the user by direct entry of the virtual channel number. Hidden virtual channels are skipped when the user is channel surfing, and appear as if undefined, if accessed by direct channel entry. Typical applications for hidden channels are test signals and NVOD services.

service_type — A 6-bit enumerated type field that identifies the type of service carried in this virtual channel, based on Table 6.7.

² A method to include such a unique 16-bit "Transmission Signal ID" in the NTSC VBI is specified in the EIA-752 specification.

Table 6.7 Service Types

service_type	Meaning
0x00	[Reserved]
0x01	analog_television — The virtual channel carries analog television programming
0x02	ATSC_digital_television — The virtual channel carries television programming (audio, video and data) conforming to the ATSC Digital Television Standard
0x03	ATSC_audio_only — The virtual channel conforms to the ATSC Digital Television Standard, and has one or more standard audio and data components but no video.
0x04	ATSC_data_broadcast_service — Conforming to the ATSC data broadcast standard under development by T3/S13.
0x05-0x3F	[Reserved for future ATSC use]

source_id— A 16-bit unsigned integer number that identifies the programming source associated with the virtual channel. In this context, a *source* is one specific source of video, text, data, or audio programming. Source ID value zero is reserved. Source ID values in the range 0x0001 to 0x0FFF shall be unique within the Transport Stream that carries the VCT, while values 0x1000 to 0xFFFF shall be unique at the regional level. Values for *source_ids* 0x1000 and above shall be issued and administered by a Registration Authority designated by the ATSC.

descriptors_length — Total length (in bytes) of the descriptors for this virtual channel that follows.

additional_descriptors_length — Total length (in bytes) of the VCT descriptor list that follows.

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 A of ISO/IEC 13818-1 “MPEG-2 Systems” after processing the entire Terrestrial Virtual Channel Table section.

6.3.2 Cable Virtual Channel Table

The Cable Virtual Channel Table is carried in private sections with table ID 0xC9, 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 Stream packets carrying the VCT sections:

- PID for Cable VCT shall have the value 0x1FFB (base_PID)
- *transport_scrambling_control* bits shall have the value ‘00’
- *adaptation_field_control* bits shall have the value ‘01’

The bit stream syntax for the Cable Virtual Channel Table is shown in Table 6.8. The semantics for the CVCT are the same as the TVCT except for those fields explicitly defined below.

table_id — An 8-bit unsigned integer number that indicates the type of table section being defined here. For the *cable_vct_section*, the *table_id* shall be 0xC9.

major_channel_number — A 10-bit number in the range 1 to 999 that represents the “major” virtual channel number associated with the virtual channel being defined in this iteration of the “for” loop. Each virtual channel must be associated with a major and a minor virtual channel number.

The major virtual channel number, along with the minor virtual channel number, act as the user's reference number for the virtual channel.

minor_channel_number — A 10-bit number in the range 0 to 999 that represents the “minor” or “sub-“ virtual channel number. This field, together with *major_channel_number*, performs a two-part virtual channel number, where *minor_channel_number* represents the second or right-hand part of the number

Table 6.9 Path Select

path_select	Meaning
0	path 1
1	path 2

out_of_band — A Boolean flag that indicates, when set, that the virtual channel defined in this iteration of the “for” loop is carried on the cable on an out-of-band physical transmission channel whose frequency is indicated by **carrier_frequency**. When clear, the virtual channel is carried within a standard tuned multiplex at that frequency.

source_id — A 16-bit unsigned integer number that identifies the programming source associated with the virtual channel. In this context, a *source* is one specific source of video, text, data, or audio programming. Source ID value zero is reserved to indicate that the programming source is not identified. Source ID values in the range 0x0001 to 0x0FFF shall be unique within the Transport Stream that carries the VCT, while values 0x1000 to 0xFFFF shall be unique at the regional level. Values for **source_ids** 0x1000 and above shall be issued and administered by a Registration Authority designated by the ATSC.

6.4 Rating Region Table (RRT)

The Rating Region Table (RRT) carries rating information for multiple geographical regions. Each RRT instance, identified by **rating_region** (the 8 least significant bits of **table_id_extension**), conveys the rating system information for one specific region. The size of each RRT instance shall not be more than 1024 bytes (including section header and trailer), and it shall be carried by only one MPEG-2 private section.

The following constraints apply to the Transport Stream packets carrying the RRT sections.

- **PID** shall have the value 0x1FFB (**base_PID**)
- **transport_scrambling_control** bits shall have the value ‘00’
- **adaptation_field_control** bits shall have the value ‘01’

The bit stream syntax for the Rating Region Table is shown in Table 6.10.

table_id — This is an 8-bit field, which shall be set to 0xCA, identifying this table as the Rating Region Table (RRT).

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’.

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 no larger than 1021.

Table 6.10 Bit Stream Syntax for the Rating Region Table

Syntax	Bits	Format
rating_region_table_section () {		
table_id	8	0xCA
section_syntax_indicator	1	'1'
private_indicator	1	'1'
zero	2	'00'
section_length	12	uimsbf
table_id_extension{		
reserved	8	0xFF
rating_region	8	uimsbf
}		
reserved	2	'11'
version_number	5	uimsbf
current_next_indicator	1	'1'
section_number	8	uimsbf
last_section_number	8	uimsbf
protocol_version	8	uimsbf
rating_region_name_length	8	uimsbf
rating_region_name_text()	var	
dimensions_defined	8	uimsbf
for(i=0; i<dimensions_defined;i++) {		
dimension_name_length	8	uimsbf
dimension_name_text()	var	
reserved	3	'111'
graduated_scale	1	bslbf
values_defined	4	uimsbf
for (j=0;j<values_defined;j++) {		
abbrev_rating_value_length	8	uimsbf
abbrev_rating_value_text()	var	
rating_value_length	8	uimsbf
rating_value_text()	var	
}		
}		
reserved	6	'111111'
descriptors_length	10	uimsbf
for (i=0;i<N;i++) {		
descriptors()	var	
}		
CRC_32	32	rpchof
}		

rating_region — An 8-bit unsigned integer number that defines the rating region to be associated with the text in this `rating_region_table_section()`. The value of this field is the identifier of this rating region, and thus this field may be used by the other tables (e.g. MGT) for referring to a specific rating region table. Values of `rating_region` are defined in Table 6.11.

Table 6.11 Rating Regions

rating_region	Rating Region Name
0x00	Forbidden
0x01	US (50 states + possessions)
0x02-0xFF	[Reserved]

version_number — This 5-bit field is the version number of the Rating Region table identified by combination of the fields `table_id` and `table_id_extension`. The version number shall be incremented by 1 modulo 32 when any field in this instance of the Rating Region Table changes. The value of this field shall be the same as that of the corresponding entry in MGT.

current_next_indicator — This 1-bit indicator is always set to '1'.

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

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

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

rating_region_name_length — An 8-bit unsigned integer number that defines the total length (in bytes) of the `rating_region_name_text()` field to follow.

rating_region_name_text() — A data structure containing a multiple string structure which represents the rating region name, e.g. "U.S. (50 states + possessions)", associated with the value given by `rating_region`. Text strings are formatted according to the rules outlined in Section 6.8. The display string for the rating region name shall be limited to 32 characters or less.

dimensions_defined — This 8-bit field (1-255) specifies the number of dimensions defined in this `rating_region_table_section()`.

dimension_name_length — An 8-bit unsigned integer number that defines the total length in bytes of the `dimension_name_text()` field to follow.

dimension_name_text() — A data structure containing a multiple string structure which represents the dimension name being described in the loop. One dimension in the U.S. rating region, for example, is used to describe the MPAA list. The dimension name for such a case may be defined as "MPAA". Text strings are formatted according to the rules outlined in Section 6.8. The dimension name display string shall be limited to 20 characters or less.

graduated_scale — This 1-bit flag indicates whether or not the rating values in this dimension represent a graduated scale, i.e., higher rating values represent increasing levels of rated content within the dimension. Value 1 means yes, while value 0 means no.

values_defined — This 4-bit field (1-15) specifies the number of values defined for this particular dimension.

abbrev_rating_value_length — An 8-bit unsigned integer number that defines the total length (in bytes) of the `abbrev_rating_value_text()` field to follow.

abbrev_rating_value_text() — A data structure containing a multiple string structure which represents the abbreviated name for one particular rating value. The abbreviated name for rating

value 0 shall be set to a null string, i.e., "". Text strings are formatted according to the rules outlined in Section 6.8. The abbreviated value display string shall be limited to 8 characters or less.

rating_value_length — An 8-bit unsigned integer number that defines the total length (in bytes) of the **rating_value_text()** field to follow.

rating_value_text() — A data structure containing a multiple string structure which represents the full name for one particular rating value. The full name for rating value 0 shall be set to a null string, i.e., "". Text strings are formatted according to the rules outlined in Section 6.8. The rating value display string shall be limited to 150 characters or less.

descriptors_length — Length (in bytes) of all of the descriptors that follow this 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 A of ISO/IEC 13818-1 "MPEG-2 Systems" after processing the entire Rating Region Table section.

6.5 Event Information Table (EIT)

The Event Information Table (EIT) contains information (titles, start times, etc.) for events on defined virtual channels. An event is, in most cases, a typical TV program, however its definition may be extended to include particular data broadcasting sessions and other information segments. Up to 128 EITs may be transmitted and each of them is referred to as EIT-k, with $k = 0, 1, \dots, 127$.

Each EIT-k can have multiple instances, each of which contains information for one virtual channel, and each of which is identified by the combination of **table_id** and **source_id**. Each EIT-k instance may be segmented into as many as 256 sections. One section may contain information for several events, but the information for one event shall not be segmented and put into two or more sections. Thus the first field after **protocol_version** for each section shall be **num_events_in_section**.

The PSIP shall have at least four EITs and no more than 128 EITs, each of which provides the event information for a certain time span. Any event programmed for a time interval that extends over one or more EITs shall be described in each of these EITs, with the same **event_id**. For instance, an event that starts at 17:30 UTC and lasts until 19:30 UTC will appear in two EITs with the same **event_id**, the EIT covering 15:00-18:00 (UTC) as well as the EIT covering 18:00-21:00 (UTC). For a particular virtual channel, an **event_id** identifies uniquely each of the events programmed for the 3-hour interval of an EIT.

Each virtual channel defined in the VCT shall have a corresponding instance of EIT-k, unless the virtual channel belongs to a group sharing the same **source_id**. Virtual channels sharing a **source_id** appear in applications such as NVOD. In such a case, the entire group will have a unique instance of EIT-k identified precisely by the **source_id**. If a virtual channel has no event in the time span covered by EIT-k, its corresponding EIT instance shall have only one section, and the field **num_events_in_section** shall be set to zero.

Events shall be in the order of their starting times, i.e., the start time of the first event shall be ahead of that of the second event, and the start time of the last event in section one shall be equal or less than that of the first event in section two with the equality holding only when both events are the same..

The Event Information Table is carried in private sections with table ID 0xCB, 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 Stream packets carrying the EIT sections:

- PID for EIT-k shall have the same value as specified in the MGT, and shall be unique among the collection of `table_type_PID` values listed in the MGT.
- `transport_scrambling_control` bits shall have the value '00'.
- `adaptation_field_control` bits shall have the value '01'.

The bit stream syntax for the Event Information Table is shown in Table 6.12.

table_id — This is an 8-bit field which shall be set to 0xCB, 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'.

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 the `CRC_32` field. The value of this field shall not exceed 4093.

source_id — This 16-bit field specifies the `source_id` of the virtual channel carrying the events described in this section.

version_number — This 5-bit field is the version number of EIT-i. The version number shall be incremented by 1 modulo 32 when any field in the EIT-i changes. Note that the `version_number` for EIT-i has no relation with that for EIT-j when j is not equal to i. The value of this field shall be identical to that of the corresponding entry in the MGT.

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

section_number — This 8-bit field gives the number of this section.

last_section_number — This 8-bit field specifies the number of the last section.

protocol_version — An 8-bit unsigned integer field whose function is to allow, in the future, this table type to carry parameters that may be structured differently than those defined in the current protocol. At present, the only valid value for `protocol_version` is zero. Non-zero values of `protocol_version` may only be processed by decoders designed to accommodate the later versions as they become standardized.

Table 6.12 Bit Stream Syntax for the Event Information Table

Syntax	Bits	Format
<code>event_information_table_section () {</code>		
<code>table_id</code>	8	0xCB
<code>section_syntax_indicator</code>	1	'1'
<code>private_indicator</code>	1	'1'
<code>reserved</code>	2	'11'
<code>section_length</code>	12	uimsbf
<code>source_id</code>	16	uimsbf
<code>zero</code>	2	'00'
<code>version_number</code>	5	uimsbf
<code>current_next_indicator</code>	1	'1'
<code>section_number</code>	8	uimsbf
<code>last_section_number</code>	8	uimsbf
<code>protocol_version</code>	8	uimsbf
<code>num_events_in_section</code>	8	uimsbf
<code>for (j = 0; j < num_events_in_section; j++) {</code>		
<code>reserved</code>	2	'11'
<code>event_id</code>	14	uimsbf
<code>start_time</code>	32	uimsbf
<code>reserved</code>	2	'11'
<code>ETM_location</code>	2	uimsbf
<code>length_in_seconds</code>	20	uimsbf
<code>title_length</code>	8	uimsbf
<code>title_text()</code>	var	
<code>reserved</code>	4	'1111'
<code>descriptors_length</code>	12	
<code>for (i=0; i < N; i++) {</code>		
<code>descriptor()</code>		
<code>}</code>		
<code>}</code>		
<code>CRC_32</code>	32	rpchof
<code>}</code>		

num_events_in_section — Indicates the number of events in this EIT section. Value 0 indicates no events defined in this section.

event_id — This field specifies the identification number of the event described. This number will serve as a part of the event ETM_id (identifier for event extended text message).

start_time — A 32-bit unsigned integer quantity representing the start time of this event as the number of GPS seconds since 12 am, January 6th, 1980.

ETM_location — This 2-bit field specifies the existence and the location of an Extended Text Message (ETM), based on Table 6.13

Table 6.13 ETM_location

ETM_location	Meaning
0x00	No ETM
0x01	ETM located in the PTC carrying this PSIP
0x02	ETM located in the PTC carrying this event
0x03	[Reserved for future ATSC use]

length_in_seconds — Duration (in seconds) of this event.

title_length — This field specifies the length (in bytes) of the `title_text()`. Value 0 means that no title exists for this event.

title_text() — The event title in the format of a multiple string structure (see Section 6.8).

descriptors_length — Total length (in bytes) of the event descriptor list that follows.

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 A of ISO-13818-1 “MPEG-2 Systems” after processing the entire Event Information Table section.

6.6 Extended Text Table

The Extended Text Table (ETT) contains Extended Text Message (ETM) streams, which are optional and are used to provide detailed descriptions of virtual channels (channel ETM) and events (event ETM). An ETM is a multiple string data structure (see Section 6.8), and thus, it 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 Extended Text Message is carried on a private section with table ID 0xCC. Each description is distinguished by its unique 32-bit `ETM_id` immediately after the field `protocol_version`. This allows the receiver to search for a single description quickly without having to parse the payload of a large table.

The ETT section for a virtual channel or an event is carried in the home physical transmission channel (the physical transmission channel carrying that virtual channel or event) with PID specified by the field `table_type_PID` in corresponding entries in the MGT. This specific PID is exclusively reserved for the ETT stream.

The following constraints apply to the Transport Stream packets carrying the ETT sections.

- PID for ETT shall have the same value as the field `table_type_PID` in corresponding entries in the MGT, and shall be unique among the collection of `table_type_PID` values listed in the MGT.
- `transport_scrambling_control` bits shall have the value ‘00’
- `adaptation_field_control` bits shall have the value ‘01’

The bit stream syntax for the Extended Text Table is shown in Table 6.14.

Table 6.14 Bit Stream Syntax for the Extended Text Table

Syntax	Bits	Format
extended_text_table_section () {		
table_id	8	0xCC
section_syntax_indicator	1	'1'
private_indicator	1	'1'
reserved	2	'11'
section_length	12	uimsbf
table_id_extension	16	0x00
reserved	2	'11'
version_number	5	0x00
current_next_indicator	1	'1'
section_number	8	0x00
last_section_number	8	0x00
protocol_version	8	uimsbf
ETM_id	32	uimsbf
extended_text_message ()	var	
CRC_32	32	rpchof
}		

table_id — Identifies this section as belonging to a Extended Text Table. (0xCC)

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'.

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. The value of the **section_length** shall be no larger than 4093.

table_id_extension — This 16-bit field shall be set to 0x00.

version_number — For the channel ETT, this 5-bit field indicates the version number of the channel ETT. The version number shall be incremented by 1 modulo 32 when any ETM in the channel ETT changes. For event ETT, this 5-bit field indicates the version number of event ETT-*i*, where *i*, as in the EIT case, is the index of time span. The version number shall be incremented by 1 modulo 32 when any ETM in the event ETT-*i* changes. Note that the **version_number** for event ETT-*i* has no relation with that for event ETT-*j* when *j* is not equal to *i*. The value of this field shall be identical to that of the corresponding entry in the MGT.

current_next_indicator — This 1-bit indicator is always set to '1' for ETT sections; the ETT 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.

protocol_version — An 8-bit unsigned integer field whose function is to allow, in the future, this table type to carry parameters that may be structured differently than those defined in the current protocol. At present, the only valid value for **protocol_version** is zero. Non-zero values of

protocol_version may only be processed by decoders designed to accommodate the later versions as they become standardized.

ETM_id — Unique 32-bit identifier of this extended text message. This identifier is assigned by the rule shown in Table 6.15.

Table 6.15 ETM ID

Bit	MSB				LSB	
	31	16	15	2	1	0
channel ETM_id	source_id		0	0	0 0
event ETM_id	source_id		event_id		1	0

extended_text_message() — The extended text message in the format of a multiple string structure (see Section 6.8).

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 A of ISO-13818-1 “MPEG-2 Systems” after processing the entire Transport Stream ETT section.

6.7 Core Descriptors

Table 6.16 lists all of the core descriptors and their descriptor tags. Asterisks mark the tables where the descriptors may appear. The range of MPEG-2 defined or reserved descriptor tags is between 0x02 and 0x3F.

Table 6.16 List of Descriptors for PSIP Tables.

Descriptor Name	Descriptor tag	Terrestrial				Cable		
		PMT	MGT	VCT	EIT	PMT	MGT	VCT
stuffing descriptor	0x80	*	*	*	*	*	*	*
AC-3 audio descriptor	0x81	*			*	*		
program identifier descriptor	0x85	*				*		
caption service descriptor	0x86	*			*	*		
content advisory descriptor	0x87	*			*	*		
extended channel name descriptor	0xA0			*				*
service location descriptor	0xA1			*				
time-shifted service descriptor	0xA2			*				*
component name descriptor	0xA3					*		
user private	0xC0-0xFF		*	*	*		*	*

6.7.1 AC-3 Audio Descriptor

The AC-3 audio descriptor, as defined in Ref. [1] and constrained in Annex B of Ref. [2], may be used in the PMT and/or in EITs.

6.7.2 Program Identifier Descriptor

The `program_identifier_descriptor`, as defined in Ref. [5], may be used in the PMT.

6.7.3 Caption Service Descriptor

The caption service descriptor provides closed captioning information, such as closed captioning type and language code for events with closed captioning service. This descriptor shall not appear on events with no closed captioning service.

The bit stream syntax for the closed captioning service descriptor is shown in Table 6.17.

Table 6.17 Bit Stream Syntax for the Caption Service Descriptor

Syntax	Bits	Format
<code>caption_service_descriptor () {</code>		
<code>descriptor_tag</code>	8	0x86
<code>descriptor_length</code>	8	uimsbf
<code>reserved</code>	3	'111'
<code>number_of_services</code>	5	uimsbf
for (<code>i=0;i<number_of_services;i++</code>) {		
<code>language</code>	8*3	uimsbf
<code>cc_type</code>	1	bslbf
<code>reserved</code>	1	'1'
if (<code>cc_type==line21</code>) {		
<code>reserved</code>	5	'11111'
<code>line21_field</code>	1	bslbf
}		
else		
<code>caption_service_number</code>	6	uimsbf
<code>easy_reader</code>	1	bslbf
<code>wide_aspect_ratio</code>	1	bslbf
<code>reserved</code>	14	'1111111111111111'
}		
}		

descriptor_tag — An 8-bit field that identifies the type of descriptor. For the `caption_service_descriptor()` the value is 0x86.

descriptor_length — An 8-bit count of the number of bytes following the `descriptor_length` itself.

number_of_services — An unsigned 5-bit integer in the range 1 to 16 that indicates the number of closed caption services present in the associated video service. Note that if the video service does not carry television closed captioning, the `caption_service_descriptor()` shall not be present either in the Program Map Table or in the Event Information Table.

Each iteration of the “for” loop defines one closed caption service present as a sub-stream within the 9600 bit per second closed captioning stream. Each iteration provides the sub-stream’s language, attributes, and (for advanced captions) the associated Service Number reference. Refer to Ref. [13] for a description of the use of the Service Number field within the syntax of the closed caption stream.

language — A 3-byte language code per ISO 639.2/B (Ref. [7]) defining the language associated with one closed caption service. The `ISO_639_language_code` field contains a three-character code as specified by ISO 639.2/B. Each character is coded into 8 bits according to ISO 8859-1 (ISO Latin-1) and inserted in order into the 24-bit field.

cc_type — A flag that indicates, when set, that an advanced television closed caption service is present in accordance with Ref. [13]. When the flag is clear, a line-21 closed caption service is present. For line 21 closed captions, the `line21_field` field indicates whether the service is carried in the even or odd field.

line21_field — A flag that indicates, when set, that the line 21 closed caption service is associated with the field 2 of the NTSC waveform. When the flag is clear, the line-21 closed caption service is associated with field 1 of the NTSC waveform. The `line21_field` flag is defined only if the `cc_type` flag indicates line-21 closed caption service.

caption_service_number — A 6-bit unsigned integer value in the range zero to 63 that identifies the Service Number within the closed captioning stream that is associated with the language and attributes defined in this iteration of the “for” loop. See Ref. [13] for a description of the use of the Service Number. The `caption_service_number` field is defined only if the `cc_type` flag indicates closed captioning in accordance with Ref. [13].

easy_reader — A Boolean flag which indicates, when set, that the closed caption service contains text tailored to the needs of beginning readers. Refer to Ref. [13] for a description of “easy reader” television closed captioning services. When the flag is clear, the closed caption service is not so tailored.

wide_aspect_ratio — A Boolean flag which indicates, when set, that the closed caption service is formatted for displays with 16:9 aspect ratio. When the flag is clear, the closed caption service is formatted for 4:3 display, but may be optionally displayed centered within a 16:9 display.

6.7.4 Content Advisory Descriptor

The Content Advisory Descriptor is used to indicate, for a given event, ratings for any or all of the rating dimensions defined in the RRT (Rating Region Table). Ratings may be given for any or all of the defined regions, up to a maximum of 8 regions per event. An Event without a Content Advisory Descriptor indicates that the rating value for any rating dimension defined in any rating region is zero. The absence of ratings for a specific dimension is completely equivalent to having a zero-valued rating for such a dimension. The absence of ratings for a specific region implies the absence of ratings for all of the dimensions in the region. The absence of a Content Advisory Descriptor for a specific event implies the absence of ratings for all of the regions for the event.

The bit stream syntax for the Content Advisory Descriptor is shown in Table 6.18.

descriptor_tag — This 8-bit unsigned integer shall have the value 0x87, identifying this descriptor as `content_advisory_descriptor`.

descriptor_length — This 8-bit unsigned integer specifies the length (in bytes) immediately following this field up to the end of this descriptor.

rating_region_count — A 6-bit unsigned integer value in the range 1 to 8 that indicates the number of rating region specifications to follow.

rating_region — An unsigned 8-bit integer that specifies the rating region for which the data in the bytes to follow is defined. The `rating_region` associates ratings data given here with data defined in a Ratings Region Table tagged with the corresponding rating region.

rated_dimensions — An 8-bit unsigned integer field that specifies the number of rating dimensions for which content advisories are specified for this event. The value of this field shall not be greater than the value specified by the field `dimensions_defined` in the corresponding RRT section.

Table 6.18 Bit Stream Syntax for the Content Advisory Descriptor

Syntax	Bits	Format
<code>content_advisory_descriptor () {</code>		
descriptor_tag	8	0x87
descriptor_length	8	uimsbf
reserved	2	'11'
rating_region_count	6	
for (i=0; i<rating_region_count; i++) {		
rating_region	8	uimsbf
rated_dimensions	8	uimsbf
for (j=0; j<rated_dimensions; j++) {		
rating_dimension_j	8	uimsbf
reserved	4	'1111'
rating_value	4	uimsbf
}		
rating_description_length	8	uimsbf
rating_description_text()	var	
}		
}		

rating_dimension_j — An 8-bit unsigned integer field specifies the dimension index into the RRT instance for the region specified by the field `rating_region`. These dimension indices shall be listed in numerical order, i.e., the value of `rating_dimension_j+1` shall be greater than that of `rating_dimension_j`.

rating_value — A 4-bit field represents the rating value of the dimension specified by the field `rating_dimension_j` for the region given by `rating_region`.

rating_description_length — An 8-bit unsigned integer value in the range zero to 80 that represents the length of the `rating_description_text()` field to follow.

rating_description_text() — The rating description in the format of a multiple string structure (see Section 6.8). The `rating_description` display string shall be limited to 16 characters or less. The

rating description text shall represent the program's rating in an abbreviated form suitable for on-screen display. The rating description text collects multidimensional text information into a single small text string. If "xxx" and "yyy" are abbreviated forms for rating values in two dimensions, then "xxx-yyy" and "xxx (yyy)" are examples of possible strings represented in `rating_description_text()`.

6.7.5 Extended Channel Name Descriptor

The extended channel name descriptor provides the long channel name for the virtual channel containing this descriptor.

The bit stream syntax for the extended channel name descriptor is shown in Table 6.19.

Table 6.19 Bit Stream Syntax for the Extended Channel Name Descriptor

Syntax	Bits	Format
<code>extended_channel_name_descriptor () {</code>		
descriptor_tag	8	0xA0
descriptor_length	8	uimsbf
long_channel_name_text()	var	
<code>}</code>		

descriptor_tag — This 8-bit unsigned integer shall have the value 0xA0, identifying this descriptor as `extended_channel_name_descriptor()`.

descriptor_length — This 8-bit unsigned integer specifies the length (in bytes) immediately following this field up to the end of this descriptor.

long_channel_name_text() — The long channel name in the format of a multiple string structure (see Section 6.8).

6.7.6 Service Location Descriptor

This descriptor specifies the stream types, PID and language code for each elementary stream. This descriptor shall appear in the TVCT, and must be valid for the current event in the corresponding virtual channel.

The bit stream syntax for the service location descriptor is shown in Table 6.20.

Table 6.20 Bit Stream Syntax for the Service Location Descriptor

Syntax	Bits	Format
<code>service_location_descriptor () {</code>		
descriptor_tag	8	0xA1
descriptor_length	8	uimsbf
reserved	3	'111'
PCR_PID	13	uimsbf
number_elements	8	uimsbf
for (i=0;i<number_elements;i++) {		
stream_type	8	uimsbf
reserved	3	'111'
elementary_PID	13	uimsbf
ISO_639_language_code	8*3	uimsbf
}		
}		

descriptor_tag — This 8-bit unsigned integer shall have the value 0xA1, identifying this descriptor as `service_location_descriptor()`.

descriptor_length — This 8-bit unsigned integer specifies the length (in bytes) immediately following this field up to the end of this descriptor.

PCR_PID — This is a 13 bit field indicating the PID of the Transport Stream packets which shall contain the PCR fields valid for the program specified by `program_number`. If no PCR is associated with a program definition for private streams then this field shall take the value of 0x1FFF.

number_elements — This 8-bit unsigned integer indicates the number of PIDs used for this program.

stream_type — This 8-bit unsigned integer field specifies the type of the elementary stream according to Table 6.21.

Table 6.21 Stream Type Assignments

Value	Description
0x00	ITU-T ISO/IEC Reserved
0x01-0x7F	As specified in Table 2.29 (Stream type assignments) of Ref. [10]
0x80	[Used in other systems]
0x81	ATSC A/53 audio
0x82-0x84	[Used in other systems]
0x85	UPID (Ref.[5])
0x86-0xBF	Reserved
0xC0-0xFF	User Private

elementary_PID — Packet Identifier for the elementary stream.

ISO_639_language_code — This 3-byte (24 bits) field, based on ISO 639.2/B, specifies the language used for the elementary stream. In case of no language specified for this elementary stream, e.g. video, each byte shall have the value 0x00.

6.7.7 Time-Shifted Service Descriptor

This descriptor links one virtual channel with one or more virtual channels that carry the same programming on a time-shifted basis. The typical application is for Near Video On Demand (NVOD) services.

The bit stream syntax for the `time_shifted_service_descriptor()` is shown in Table 6.22.

Table 6.22 Bit Stream Syntax for the Time Shifted Service Descriptor

Syntax	Bits	Format
<code>time_shifted_service_descriptor () {</code>		
descriptor_tag	8	0xA2
descriptor_length	8	uimsbf
reserved	3	'111'
number_of_services	5	uimsbf
for (<code>i=0;i<number_of_services;i++</code>) {		
reserved	6	'111111'
time_shift	10	uimsbf
reserved	4	'1111'
major_channel_number	10	uimsbf
minor_channel_number	10	uimsbf
}		
}		

descriptor_tag — This 8-bit unsigned integer shall have the value 0xA2, identifying this descriptor as `time_shifted_service_descriptor()`.

descriptor_length — This 8-bit unsigned integer specifies the length (in bytes) immediately following this field up to the end of this descriptor.

number_of_services — A 5-bit number in the range 1 to 20 that indicates the number of time-shifted services being defined here.

time_shift — A 10-bit number in the range 1 to 720 that represents the number of minutes the time-shifted service indicated by `major_channel_number` and `minor_channel_number` is time-shifted from the virtual channel associated with this descriptor.

major_channel_number — A 10-bit number in the range 1 to 999 that represents the “major” channel number associated with a time-shifted service.

minor_channel_number — A 10-bit number in the range 0 to 999 that, when non-zero, represents the “minor” or “sub-” channel number of the virtual channel that carries a time-shifted service.

6.7.8 Component Name Descriptor

Table 6.23 defines the `component_name_descriptor()`, which serves to define an optional textual name tag for any component of the service.

Table 6.23 Bit Stream Syntax for the Component Name Descriptor

Syntax	Bits	Format
component_name_descriptor() {		
descriptor_tag		80xA3
descriptor_length		8uimsbf
component_name_string()		v
	ar	
}		

descriptor_tag — This 8-bit unsigned integer shall have the value 0xA3, identifying this descriptor as component_name_descriptor.

descriptor_length — This 8-bit unsigned integer specifies the length (in bytes) immediately following this field up to the end of this descriptor.

component_name_string() — The name string in the format of a multiple string structure (see Section 6.8).

6.7.9 Stuffing Descriptor

For certain applications it is necessary to define a block of N bytes as a placeholder. The N bytes themselves are not to be processed or interpreted. The `stuffing_descriptor()` is specified for this purpose. The `stuffing_descriptor()` is simply a descriptor type for which the contents, as indicated by the `descriptor_length` field, are to be disregarded. The tag type for the stuffing descriptor is 0x80. The `stuffing_descriptor()` may appear where descriptors are allowed in any table defined in the PSIP.

6.8 Multiple String Structure

This is a general data structure used specifically for text strings. Text strings appear as event titles, long channel names, the ETT messages, and RRT text items. The bit stream syntax for the Multiple String Structure is shown in Table 6.24.

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

ISO_639_language_code — This 3-byte (24 bits) field, based on ISO 639.2/B, specifies the language used for the i^{th} string.

number_segments — This 8-bit unsigned integer field identifies the number of segments in the following data. A specific mode is assigned for each segment.

Table 6.24 Bit Stream Syntax for the Multiple String Structure

Syntax	Bits	Format
<code>multiple_string_structure () {</code>		
<code>number_strings</code>	8	uimsbf
<code>for (i= 0;i< number_strings;i++) {</code>		
<code>ISO_639_language_code</code>	8*3	uimsbf
<code>number_segments</code>	8	uimsbf
<code>for (j=0;j<number_segments;j++) {</code>		
<code>compression_type</code>	8	uimsbf
<code>mode</code>	8	uimsbf
<code>number_bytes</code>	8	uimsbf
<code>for (k= 0;k<number_bytes;k++)</code>		
<code>compressed_string_byte [k]</code>	8	bslbf
<code>}</code>		
<code>}</code>		
<code>}</code>		

compression_type — This 8-bit field identifies the compression type for the j^{th} segment. Allowed values for this field are shown in Table 6.25.

Table 6.25 Compression Types

compression_type	compression method
0x00	No compression
0x01	Huffman coding using standard encode/decode tables defined in Table C.4 and C.5 in Annex C.
0x02	Huffman coding using standard encode/decode tables defined in Table C.6 and C.7 in Annex C.
0x03 to 0xAF	reserved
0xB0 to 0xFF	user private

mode — An 8-bit value representing the text mode to be used to interpret characters in the segment to follow. See Table 6.26 for definition. Mode values in the range zero through 0x3E select 8-bit Unicode™ character code pages. Mode value 0x3F selects 16-bit Unicode™ character coding. Mode values 0x40 through 0xDF are reserved for future use by ATSC. Mode values 0xE0 through 0xFE are user private. Mode value 0xFF indicates the text mode is not applicable. Decoders shall ignore string bytes associated with unknown or unsupported mode values.

number_bytes — This 8-bit unsigned integer field identifies the number of bytes that follows.

compressed_string_byte[k] — The k^{th} byte of the j^{th} segment.