SMPTE 312M-1999

SMPTE STANDARD

for Television — Splice Points for MPEG-2 Transport Streams

1 Scope

This standard defines constraints on the encoding of and syntax for MPEG-2 transport streams such that they may be spliced without modifying the PES packet payload. Generic MPEG-2 transport streams, which do not comply with the constraints in this standard, may require more sophisticated techniques for splicing.

The constraints specified here are applied individually to programs within transport streams. A program is a collection of video, audio, and data streams which share a common time base. The presence of a video component is not assumed. The standard enables splicing of programs within a multiprogram transport stream either simultaneously or independently. Splice Points in different programs may be presentation-time-coincident, but do not have to be. The standard may also be used with single-program transport streams.

The document specifies constraints for both seamless and nonseamless Splice Points. Seamless Splice Points must adhere to all the constraints. Nonseamless Splice Points must adhere to all constraints except those prefaced with the clause "to enable seamless splicing." A bit stream which is compliant with this standard shall conform to the constraints defined in clauses 5 and 6. Such a bit stream may contain any number of seamless, nonseamless, or both types of Splice Points. If a bit stream does not contain splice event command and control information, the constraints in clause 7 do not apply to the bit stream. Mechanisms for transmission of time code in MPEG-2 transport/elementary bit streams shall be addressed by other standards.

In addition to constraints for creating spliceable bit streams, this standard specifies the technique for carrying notification of upcoming Splice Points in the transport stream. A splice information table is defined for notifying downstream devices of splice events, such as a network break or return from a network break. The splice information table which pertains to a given program is carried in a separate PID stream referred to by that program's program map table. In this way, splice event notification can pass through transport stream remultiplexers without need for special processing. A bit stream which is compliant with this standard and which carries splice event command and control information shall conform to the constraints in clauses 5, 6, and 7.

The standard does not address constraints on splicing devices. Annex A outlines several issues that should be considered in the design of such devices.

NOTE – Numbers given in brackets are subject to confirmation.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

ANSI/SMPTE 12M-1995, Television, Audio and Film — Time and Control Code

ATSC A/53, Digital Television

595 W. Hartsdale Ave., White Plains, NY 10607
(914) 761-1100

American National Standard

Approved
April 8, 1999

ITU-T Rec. H.222.0 / ISO/IEC 13818-1 (1996-04), Information Technology — Generic Coding of Moving Pictures and Associated Audio Information: Systems

ITU-T Rec. H.262 / ISO/IEC 13818- 2(1996-05), Information Technology — Generic Coding of Moving Pictures and Associated Audio Information: Video

3 Introduction

3.1 Buffer issues

Splicing of MPEG bit streams requires managing buffer fullness of the decoder's buffers. When MPEG bit streams are encoded, there is an inherent buffer occupancy at every point in time (see figure 1). The buffer fullness corresponds to a delay, the amount of time that a byte spends in the buffer. When splicing two separately encoded bit streams, the delay at the Splice Point will not usually match. This mismatch in delay can cause the buffer to overflow or underflow at some time in the future (see figure 2).

To avoid unpredictable underflows and overflows, two splicing techniques have been defined. The seamless splicing method requires that the MPEG encoder match the delay at splicing points to a given value. The nonseamless method does not require the encoder to match the delay. Instead, the splicing device is responsible for matching the delay of the new material and the old material as well as it can. In some cases, this will result in a controlled decoder buffer underflow. This underflow can be masked in the decoder by holding the last frame of the outgoing video and muting the audio until the first access unit of the new stream has been decoded. In the worst case, this underflow may last for a few frame times. Both splicing methods may cause an underflow of the audio buffer, and consequently a gap in the presentation of audio at the receiver. The perceived quality of the splice in both cases will benefit from audio decoders that can handle a gap in audio data gracefully.

3.2 Splice Points

To enable the splicing of compressed bit streams, this standard defines Splice Points. Splice Points in an MPEG-2 transport stream provide opportunities to switch from one program to another. They indicate a safe place to switch, a place in the bit stream where a switch can be made, and result in good visual and audio quality. In this way, they are analogous to the vertical interval used to switch uncompressed video. Unlike uncompressed video, frame boundaries in an MPEG-2 bit stream are not evenly spaced. Therefore, the syntax of the transport packet itself is used to convey where these Splice Points occur.

Transport streams are created by multiplexing PID streams. In this standard, two types of Splice Points for PID streams are defined: Out Points and In Points. In Points are places in the bit streams where it is safe to enter and start decoding the bit stream. Out Points are places where it is safe to exit the bit stream. Ways to group In Points of individual PID streams into Program In Points in order to enable the switching of entire programs (video with audio) are defined. Program Out Points for exiting a program are also defined.

Out Points and In Points are imaginary points in the bit stream located between two transport stream packets. An Out Point and an In Point may be co-located; that is, a single packet boundary may serve as both a safe place to leave a bit stream and a safe place to enter it (see figure 3).

The output of a switching operation (see figure 4) will begin with packets from one stream up until its Out Point followed by packets from another stream starting with the first packet following an In Point.

3.3 Program Splice Points

Part of this standard describes requirements for grouping In Points of a set of PID streams into Program In Points and for grouping Out Points of a set of PID streams into Program Out Points. Program In Points and Program Out Points are sets of PID stream In Points or Out Points which correspond in presentation time to the underlying data. In MPEG, audio is typically organized into audio frames. Because video and audio frames have different durations and their presentation times do not necessarily align, this standard defines exactly what it means for PID stream Splice Points to correspond in time.

Figure 5 shows a splice between two programs, one called the old stream and one called the new stream. Each program contains a video PID and an audio PID. The output of the splice is shown below the two input streams. In the old stream, the position of PID stream Out Points which create a program Out Point is shown. In the new stream, the position of PID stream In Points which create a Program In Point is shown.

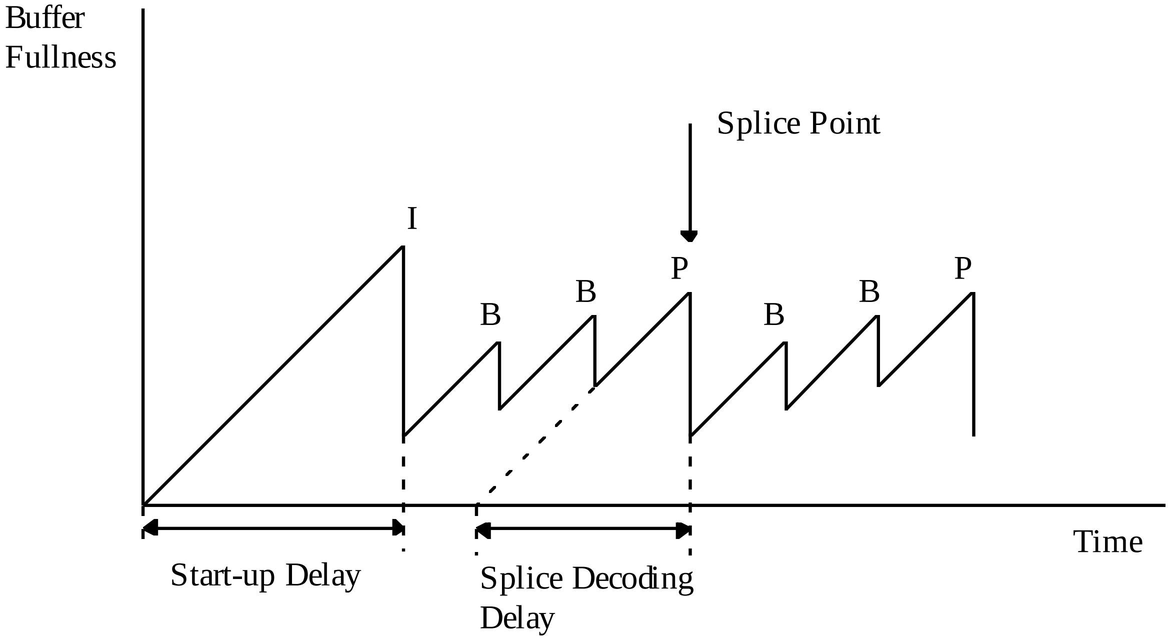
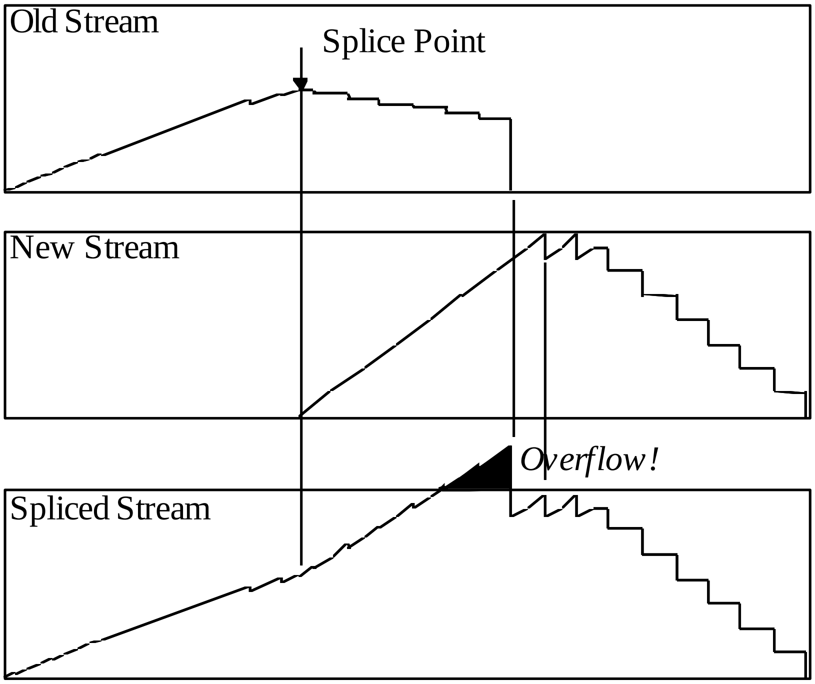


**Figure 1 – Fullness of a theoretical decoder's buffer model
(as described in ISO/IEC 13818-2 annex C)**



Here is an example of an old stream and a new stream that results in a splice overflow.

At the time of the splice bits from the old stream stop entering the buffer and bits from the new stream begin entering.

When spliced the two streams overflow the buffer.

Figure 2 – Example of decoder buffer overflow as a result of an unconstrained splice

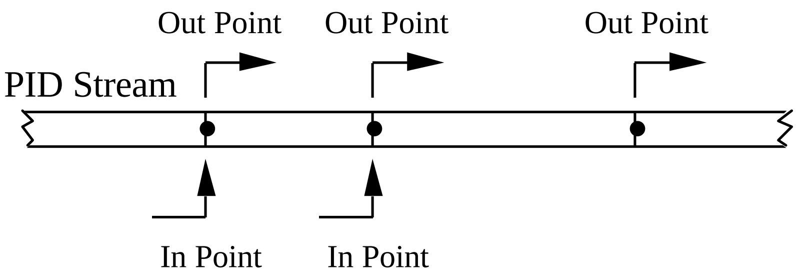


Figure 3 – In Points and Out Points in a PID stream

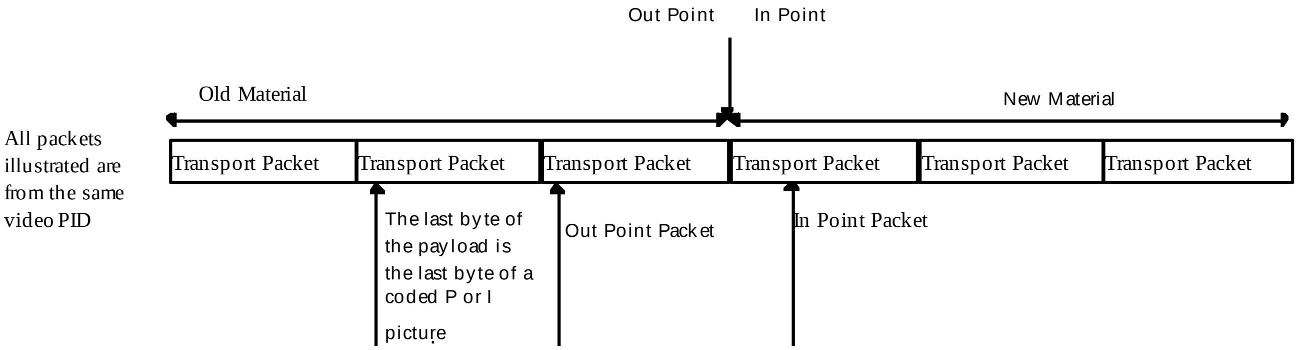


Figure 4 – Example of a co-located Out Point and In Point in a video PID stream

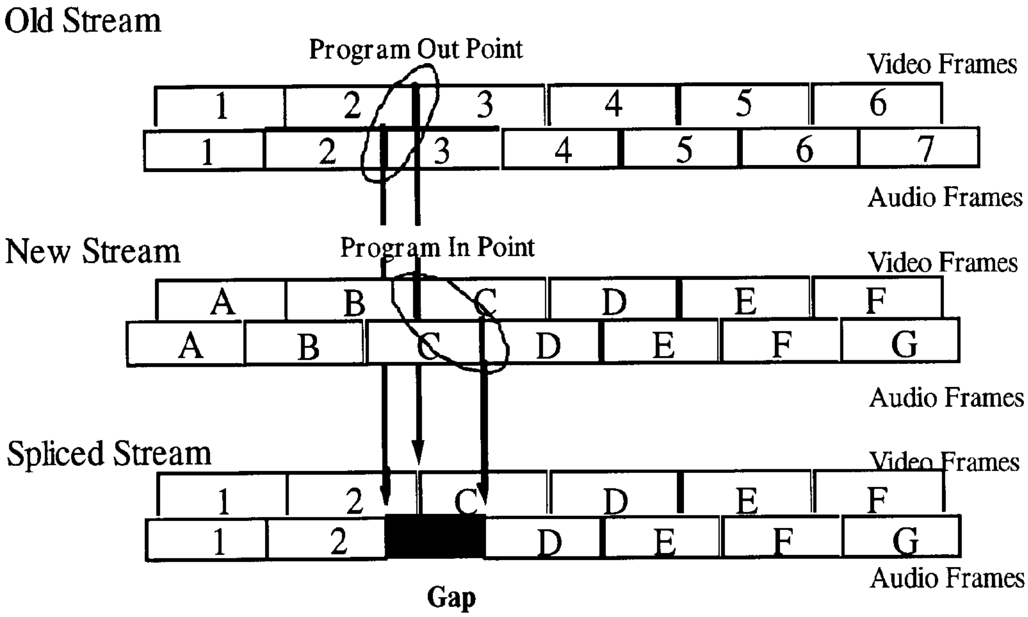


Figure 5 – Presentation time representation of a Program Out Point and a Program In Point

Although Splice Points in a Program Splice Point correspond in presentation time, they do not usually appear near each other in the transport stream. Because compressed video takes much longer to decode than audio, the audio Splice Points may lag the video Splice Points by as much as hundreds of milliseconds and by an amount that can vary from moment to moment (this relationship is shown in figure 6). This standard defines the relationship of Splice Points in bit stream order as well as in presentation time.

3.4 Splice events

This standard provides a method for in-band signalling of schedule, preroll, and execute splice event messages to downstream splicing equipment. A splice event identifies which Splice Point to use for a splice. A splice information table carries splice events. Each splice event is analogous to a cue tone. The splice information table incorporates the functionality of cue tones and extends it to enable the scheduling of splice events in advance.

This standard establishes that the splice information table be carried on a per-program basis in a PID stream with a designated stream_type. The program's splice information PID is designated in the program's program map table. In this way, the splice information table is switched with the program as it goes through remultiplexing operations. A common stream_type identifies all PID streams which carry splice information tables. Remultiplexers may use this stream_type field to drop splice information prior to sending the transport stream to the end-user device.

4 Definition of terms

Throughout this standard the terms below have given specific meanings. Because some of the terms which are defined in ISO/IEC 13818 have very specific technical meanings, the reader is referred to the original source for their definition. For terms defined by this standard, brief definitions are given below. More extensive descriptions of some terms are given in 3.2. Constraints in clause 5 provide the specific technical definition.

4.1 ATSC: Advanced Television Systems Committee.

4.2 bslbf: Bit string, left bit first, where left is the order in which bit strings are written.

4.3 decoding delay: The time from when a packet enters the decoder buffer until it is removed.

4.4 DTS: Decoding time stamp (see ITU-T Rec. H.222.0 / ISO/IEC 13818-1).

4.5 DTS_next_AU: DTS value of the next access unit (see ITU-T Rec. H.222.0 / ISO/IEC 13818-1).

4.6 first presentation unit (FPU): In regard to an In Point, the presentation unit which follows the given In Point that has the earliest presentation time.

4.7 422P@ML: 422 Profile at Main Level (see ITU-T Rec. H.222.0 / ISO/IEC 13818-2).

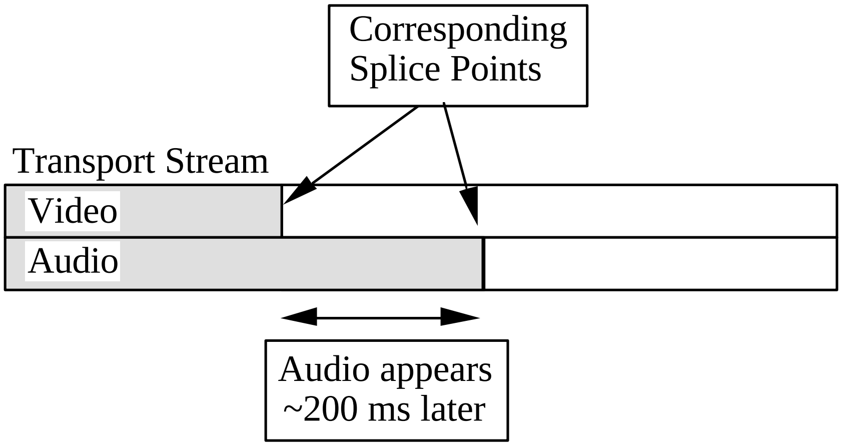


Figure 6 – Bit stream order representation of a Program Out Point

4.8 In Point: A point in a PID stream where a splicing device may enter.

4.9 In Point packet (IPP): The first packet after an In Point in a PID stream.

4.10 last presentation unit (LPU): In regard to an Out Point, the presentation unit which precedes the given Out Point that has the latest presentation time.

4.11 max_splice_rate: (see ITU-T Rec. H.222.0 / ISO/IEC 13818-1).

4.12 MP@HL: Main Profile at High Level (see ITU-T Rec. H.262 / ISO/IEC 13818-2).

4.13 MP@ML: Main Profile at Main Level (see ITU-T Rec. H.262/ ISO/IEC 13818-2).

4.14 Out Point: A point in a PID stream where a splicing device may exit.

4.15 Out Point packet (OPP): The last packet prior to an Out Point in a PID stream.

4.16 PCR: Program clock reference (see ITU-T Rec. H.222.0 / ISO/IEC 13818-1).

4.17 PCR_flag: (see ITU-T Rec. H.2220 / ISO/IEC 13818-1).

4.18 PCR_PID: Identifier carried in the program map table. The PID contained in a program that has been selected to carry the PCR (see ITU-T Rec. H.222.0 / ISO/IEC 13818-1).

4.19 PES: Packetized elementary stream (see ITU-T Rec. H.222.0 / ISO/IEC 13818-1).

4.20 picture_structure: (see ITU-T Rec. H.262 / ISO/IEC 13818-2).

4.21 PID: Packet identifier; a unique 13-bit value used to identify the type of data stored in the packet payload (see ITU-T Rec. H.222.0 / ISO/IEC 13818-1).

4.22 PID stream: All the packets with the same PID within a transport stream.

4.23 PMT: Program map table (see ITU-T Rec. H.222.0 / ISO/IEC 13818-1).

4.24 Profile@Level: Designation of the subset of the MPEG-2 video coding specification (see ITU-T Rec. H.262 / ISO/IEC 13818-2).

4.25 program: A collection of video, audio, and data PID streams which share a common time base.

4.26 Program In Point (PIP): A group of PID stream In Points which correspond in presentation time. This standard defines correspondence. PID streams with In Points contained in a Program In Point may be a subset of all PID streams contained within a program as defined by the PMT.

4.27 Program Out Point (POP): A group of PID stream Out Points which correspond in presentation time. This standard defines correspondence. PID streams with Out Points contained in a Program Out Point may be a subset of all PID streams contained within a program as defined by the PMT.

4.28 program splice point (PSP): Either a Program Out Point or a Program In Point.

4.29 progressive_frame: (see ITU-T Rec. H.262 / ISO/IEC 13818-2).

4.30 progressive_sequence: (see ITU-T Rec. H.262 / ISO/IEC 13818-2).

4.31 PTS: Presentation time stamp (see ITU-T Rec. H.222.0 / ISO/IEC 13818-1).

4.32 repeat_first_field: (see ITU-T Rec. H.262 / ISO/IEC 13818-2).

4.33 rpchof: Remainder polynomial coefficients, highest order first.

4.34 seamless_splice_flag: (see ITU-T Rec. H.222.0 / ISO/IEC 13818-1).

4.35 sequence_end_code: (see ITU-T Rec. H.262 / ISO/IEC 13818-2).

4.36 splice_countdown: (see ITU-T Rec. H.222.0 / ISO/IEC 13818-1).

4.37 splice_decoding_delay: (see ITU-T Rec. H.222.0 / ISO/IEC 13818-1).

4.38 splice event: A splice of one or more PID streams.

4.39 Splice Point: A point in a PID stream that is either an Out Point or an In Point.

4.40 splice_type: (see ITU-T Rec. H.222.0 / ISO/IEC 13818-1).

4.41 splicing_point_flag: (see ITU-T Rec. H.222.0 / ISO/IEC 13818-1).

4.42 top_field_first: (see ITU-T Rec. H.222.0 / ISO/IEC 13818-1).

4.43 uimsbf: Unsigned integer, most significant bit first.

5 Constraints

Constraints fall into three broad categories: general constraints, Out Point constraints, and In Point constraints. General constraints apply to the transport stream, but are not specific to Out Points or In Points. Out Point constraints define the nature of Out Points. In Point constraints define the nature of In Points. Out Point constraints are divided into subcategories: Out Point constraints that apply to all spliceable PID streams, additional constraints for video PID streams, additional constraints for audio PID streams, and constraints which define Program Out Points. In Point constraints are divided into subcategories: In Point constraints that apply to all spliceable PID streams, additional constraints for video PID streams, additional constraints for audio PID streams, and constraints which define Program In Points. Note that both video and audio PID streams must adhere to the Splice Point constraints for spliceable PID streams.

In Points and Out Points may be created to enable either seamless splicing or nonseamless splicing. Both seamless and nonseamless Splice Points may be introduced in the same PID stream. A seamless Splice Point may be used as a nonseamless Splice Point.

5.1 General constraints

5.1.1 If there is a Splice Point in any PID stream of a transport stream, the entire transport stream shall be compliant with ITU-T Rec. H.2220 / ISO/IEC 13818-1.

5.1.2 Every In Point, except at the beginning of a bit stream, shall be co-located with an Out Point. An Out Point does not need to be immediately followed by an In Point.

5.2 Out Point constraints

5.2.1 Out Point constraints for all spliceable PID streams

5.2.1.1 The splicing_point_flag shall be set to 1 in the Out Point packet.

5.2.1.2 The splice_countdown shall be set to 0 (0×00) in the Out Point packet.

5.2.1.3 The last byte of the Out Point packet payload shall be the last byte of a PES packet.

5.2.1.4 If PID equals PCR_PID, the Out Point packet shall have the PCR_flag set to 1 and carry a PCR value.

5.2.1.5 The seamless_splice_flag shall be set to 1 in the Out Point packet.

NOTE – The seamless_splice_flag is an MPEG-2 syntax element which when set to 1 indicates the presence of two other fields: splice_type and DTS_next_AU. DTS_next_AU is required by this standard in both seamless and nonseamless Out Point packets. Therefore, the seamless_splice_flag shall be set to 1 for all Out Point packets.

5.2.1.6 DTS_next_AU shall be set in the Out Point packet, according to the definition in ISO/IEC 13818-1.

5.2.1.7 The Out Point packet shall carry the splice_type field. The splice_type value for video is given by 5.2.2.3. For audio, the value is given by 5.2.3.2.

5.2.2 Additional Out Point constraints for video PID streams

5.2.2.1 The last picture (in presentation order) preceding an Out Point shall be either a P or an I picture. An Out Point shall not occur between the two fields of a coded frame (as defined in ISO/IEC 13818-2 clause 6.1.1.4.1).

5.2.2.2 The Out Point packet shall contain a payload of exactly four bytes. The value of these four bytes shall be either 0×00000000 or

0×000001b7. If zero, these bytes may later be replaced by a splicing device with a sequence_end_code (0×000001b7). These bytes are considered to be the last bytes of a video PES packet and, thus, satisfy 5.2.1.3.

5.2.2.3 The value of splice_type shall be selected from table 1. To enable seamless splicing, the value shall be selected from the first eight entries according to Profile@Level and application. For Out Points which do not satisfy the constraints for seamless splicing, the value shall be the last table entry in table 1.

NOTE -- ISO/IEC 13818-1 as amended by amendment 4 provides splice_type values which are replicated in table 1.

5.2.2.4 To enable seamless splicing, the last payload byte of the Out Point packet shall remain in the VBV buffer an amount of time equal to:

splice_decoding_delay - display_period_last_AU$_{old}$

where display_period_last_AU$_{old}$ is the display duration of the video access unit of the old material which begins presentation at the time when the last video access unit is removed from the buffer (see ITU-T Rec. H.222.0 / ISO/IEC 13818-1).

5.2.2.5 To enable seamless splicing, the last picture (in presentation order) before an Out Point shall be either a frame picture or a bottom field picture.

Table 1 – Splice_type

Application[1]	Profile@Level	splice_type	splice_decoding_delay	max_splice_rate[4]
ATSC transmission	MP@HL	1100[2]	250 ms	19 Mb/s
Other transmission	MP@ML	0011	250 ms	7.2 Mb/s
Contribution	422P@ML	0100	250 ms	36 Mb/s
HDTV contribution	422P@HL	0100[3]	250 ms	180 Mb/s
Studio	422P@ML	0001	90 ms	50 Mb/s
HDTV studio	422P@HL	0001[3]	90 ms	300 Mb/s
Studio	422P@ML	0000	45 ms	50 Mb/s
HDTV Studio	422P@HL	0000[3]	45 ms	300 Mb/s
Nonseamless	Any	1111[2]	Undefined	Undefined

NOTE -- The use of other splice_type values within the scope of this standard is reserved.

[1] When these applications are implemented with constrained bit streams, as specified in this standard, the value of splice_type shall be selected from table 1 accordingly.

[2] These values are assigned by SMPTE from the user-defined values.

[3] Values of splice_type for 422P@HL are assigned by SMPTE.

[4] The value of max_splice_rate refers to a video bit rate value (refer to ISO/IEC 13818-1 for a complete definition of max_splice_rate).

[5] The value of 180 Mb/s may not be suitable for all HDTV contribution applications, expecially those with maximum bit rates which are much lower than 180 Mb/s. The primary distribution services of contribution quality HDTV signals are still under discussion within SMPTE and other standards bodies. As the bit rates for these services are established, more appropriate values for the seamless splicing parameters may be standardized.

In the case of an interlaced sequence (progressive_sequence equals 0), the following constraints on the use of top_field_first and repeat_first_field shall apply:

– If the last picture (in presentation order) before an Out Point is a frame picture with the top_field_first bit equal to 1, then the repeat_first_field bit of that picture shall be 0.

– If the last picture (in presentation order) before an Out Point is a frame picture with the top_field_first bit equal to 0, then the repeat_first_field bit of that picture shall be 1.

5.2.3 Additional Out Point constraints for audio PID streams

5.2.3.1 If audio is organized into frames, then the last byte of an Out Point packet shall be the last byte of an audio frame.

5.2.3.2 The value of splice_type in the audio Out Point packet shall be set to 0000.

5.2.4 Program Out Point constraints

5.2.4.1 A Program Out Point shall consist of a set of Out Points, one per PID stream, which correspond in presentation time. Within a Program Out Point, one Out Point in each non-PCR PID shall correspond to a single Out Point in the PCR PID. This correspondence is defined in 5.2.4.2.

5.2.4.2 For an Out Point in the PCR PID stream, all non-PCR PID streams shall contain an Out Point such that the time which is the sum of the presentation time of the non-PCR PID stream's last presentation unit (LPU) plus its duration shall not be later than but shall be contemporaneous with or earlier than the time which is the sum of the presentation time of the PCR PID stream's LPU plus its duration, by an amount not to exceed the maximum frame duration of the elementary stream in the non-PCR PID stream.

$$\mathrm{PTS}_{\mathrm{PCR\_LPU}} + \mathrm{Duration}_{\mathrm{PCR\_LPU}} - \mathrm{MaxDuration}_{\mathrm{nonPCR}} < \mathrm{PTS}_{\mathrm{nonPCR\_LPU}} + \mathrm{Duration}_{\mathrm{nonPCR\_LPU}} <= \mathrm{PTS}_{\mathrm{PCR\_LPU}} + \mathrm{Duration}_{\mathrm{PCR\_LPU}}$$

5.2.4.3 Of the Out Points in a Program Out Point, the Out Point packet of the PCR_PID shall occur first in the transport stream.

5.3 In Point constraints

5.3.1 In Point constraints for all spliceable PID streams

5.3.1.1 The splice_point_flag shall be set to 1 in the In Point packet.

5.3.1.2 The splice_countdown field shall be set to –1 (0×FF) in the In Point packet.

5.3.1.3 If the PID equals PCR_PID, the In Point packet shall have the PCR_flag set to 1 and shall carry a PCR value.

NOTE – The combination of 5.3.1.3 and 5.3.1.8 implies that the In Point packet of the PCR_PID must carry both a PCR and PTS/DTS value. Therefore, every In Point packet of the PCR_PID must have the adaptation_field_control parameter (in the transport stream packet header) set to 11. This indicates that the In Point packet of the PCR_PID has both an adaptation field (for carrying the PCR) and a payload (for carrying the PTS/DTS fields).

5.3.1.4 The payload_unit_start_indicator shall be set to 1 in the In Point packet.

5.3.1.5 The data_alignment_indicator of the PES packet shall be set to 1.

5.3.1.6 The random_access_indicator shall be set to 1 in the In Point packet.

5.3.1.7 Bit streams which are compliant with this standard shall not have both random_access_indicator set to 1 and splice_countdown equal to --1, except in an In Point packet.

5.3.1.8 The first payload byte of the In Point packet shall be the first byte of a PES header, which shall carry a PTS. It shall carry DTS if DTS does not equal PTS.

5.3.1.9 The seamless_splice_flag shall be set to 1 in the In Point packet.

NOTE – The seamless_splice_flag is an MPEG-2 syntax element, which when set to 1 indicates the presence of two other fields, splice_type and DTS_next_AU. DTS_next_AU is required by this standard in both seamless and nonseamless In Point packets. Therefore, the seamless_splice_flag shall be set to 1 for all In Point packets.

5.3.1.10 DTS_next_AU shall be set in the In Point packet to the same value as the DTS value carried in the PES header in the In Point packet. If DTS is not present in the PES header, then DTS_next_AU shall be set to the value of PTS.

5.3.1.11 The In Point packet shall carry the splice_type field. The splice_type value for video is given by 5.3.2.2. The value for audio is given in 5.3.3.3.

5.3.2 Additional In Point constraints for video PID streams

5.3.2.1 The first PES packet payload following an In Point shall begin with a sequence_header. The first coded picture after the sequence_header shall be an I picture. Any B pictures following an In Point shall not use prediction which references pictures prior to the In Point.

5.3.2.2 The value of splice_type shall be selected from table 1. To enable seamless splicing, the value shall not be 1111. For In Points which do not satisfy the constraints for seamless splicing, the value shall be 1111.

5.3.2.3 To enable seamless splicing, the time between when the first byte of the PES payload following an In Point enters the VBV buffer and the time when that byte is removed from the VBV buffer shall be equal to the splice_decoding_delay time given in table 1 as determined by the value of splice_type in the In Point packet and the profile_and_level_indication in the sequence_extension.

5.3.2.4 To enable seamless splicing, the picture_structure of the first picture (in presentation order) after an In Point shall be either frame picture (11) or top field (01). In the case of an interlaced sequence (progressive_sequence equals 0), the following constraint shall hold:

– If the picture_structure of the first picture (in presentation order) after an In Point is frame picture, then the top_fleld_first bit shall be equal to 1 for that picture.

5.3.2.5 If closed caption information is carried according to ATSC A/53, then one of the following shall be true for the picture user_data of the first coded picture following an In Point (see ATSC A/53 for the definition of these syntax elements):

– process_cc_data_flag shall be set to 0, or

For the first iteration of the cc_data loop where cc_valid equals 1

– cc_type shall be set to 00, 01, or 11 (NTSC or ATSC packet start);

– cc_type shall not be set to 10 (ATSC packet data).

5.3.3 Additional In Point constraints for audio PID streams

5.3.3.1 If audio is organized into frames, the first payload byte following an In Point shall be the first byte of an audio frame.

5.3.3.2 Data required for decoding the audio access units following the In Point shall not be contained in any audio frames prior to the In Point.

NOTE – Some audio compression methods (MPEG-2 layer III) make use of a bit reservoir in preceding compressed audio frames. The technique is explicitly disallowed at an audio In Point.

5.3.3.3 The value of splice_type in the audio In Point packet shall be set to 0000.

5.3.4 Program In Point constraints

5.3.4.1 A Program In Point shall consist of a set of In Points, one per PID stream, which correspond in presentation time. Within a Program In Point, one In Point in each non-PCR PID stream shall correspond to a single In Point in the PCR PID stream. This correspondence is defined in 5.3.4.2.

5.3.4.2 For an In Point in the PCR PID stream, all non-PCR_PID streams shall contain an In Point such that the presentation time of the non-PCR_PID stream's first presentation unit (FPU) shall not be earlier than but contemporaneous with or later than the presentation time of the PCR PID stream's FPU by an amount not to exceed the maximum frame duration of the elementary stream in the non-PCR PID stream.

$$\mathrm{PTS}_{\mathrm{PCR\_FPU}} <= \mathrm{PTS}_{\mathrm{nonPCR\_FPU}} < \mathrm{PTS}_{\mathrm{PCR\_FPU}} + \mathrm{MaxDuration}_{\mathrm{nonPCR}}$$

5.3.4.3 Of the In Points in a Program In Point, the In Point packet of PCR_PID shall occur first in the transport stream.

6 Registration descriptor

The registration descriptor (ITU-T Rec. H.222.0 / ISO/IEC 13818-1, *table 2-46 – Registration Descriptor*, clause 2.6.8) is defined to identify unambiguously the transport streams which comply with this standard. The registration descriptor shall be carried in the transport stream description table (PID = 0×0002) in the TS_description_section (table_id = 0×03) (see ISO/IEC 13818-1 amendment 3). The content of the registration descriptor is specified in table 2 and below:

6.1 descriptor_tag: The descriptor_tag is an 8-bit field which identifies each descriptor. For registration descriptors, this field shall be set to 0×05.

6.2 descriptor_length: The descriptor_length in an 8-bit field specifying the number of bytes of the descriptor immediately following descriptor_length field. For this registration descriptor, descriptor_length shall be set to 0×04.

6.3 SMPTE_splice_format_identifier: SMPTE has assigned a value of 0×53504C43 (ASCII String SPLC) to this 4-byte field to identify the transport stream in which it is carried as complying with this standard.

7 Splice information table

7.1 Overview

The splice information table provides command and control information to the splicer. It notifies the splicer of splice events in advance of those events. It is designed to accommodate ad insertion in network feeds. In this environment, an example of a splice event would include 1) a splice out of a network feed into an ad, or 2) the splice out of an ad to return to the network feed. The splice information section may be sent multiple times and splice events may be cancelled. Syntax for a splice_info_section is defined to convey the splice information table.

A splice event is the act of splicing one or more elementary PID streams within a program. A splice event is identified uniquely with a splice_event_id. Splice events may be communicated in three ways: they may be scheduled ahead of time, a preroll warning may be given, or a command given to execute the splice event at specified Splice Points. These three messages are sent via the splice_info_section. The different messages are specified by the splice_command_type field. Depending on the value of this field, different constraints apply to the remaining syntax.

When signalling splice events, the execute message must be sent at least once for each splice event. A preroll message may be sent one or more times prior to each splice event. For example, a preroll message could be sent at 8, 5, 4, and 2 seconds prior to the time of the splice event to give the splicer warning to set up for the impending splice. The use of preroll messages is similar to the use of analog cue tones in existing systems. Sending preroll messages is optional. A schedule of splice events may be conveyed in advance with the schedule message. Schedule messages are also optional. The complete syntax is presented below, followed by definition of terms, followed by constraints.

7.2 Splice information section syntax

Fields in tables 3 to 14 which are represented by the syntax, *function_name()*, indicate a complex field which is described in a separate table. The number of bits (Bits), and the description of those bits (Mnemonic), are given for each field that is not a function defined in a different table.

Table 2 – Registration descriptor syntax

Syntax	Bits	Mnemonic
registration_descriptor() {		
descriptor_tag	**8**	**uimsbf**
descriptor_length	**8**	**uimsbf**
SMPTE_splice_format_identifier	**32**	**uimsbf**
}		

Table 3 – splice_info_section

Syntax	Bits	Mnemonic
splice_info_section() {		
table_id	**8**	**uimsbf**
section_syntax_indicator	**1**	**bslbf**
private_indicator	**1**	**bslbf**
reserved	**2**	**bslbf**
section_length	**12**	**uimsbf**
if (section_syntax_indicator == 0) {		
for (i=0); i<section_length; i++) {		
stuffing_byte	**8**	**bslbf**
}		
}		
else {		
table_id_extension	**16**	**bslbf**
reserved	**2**	**bslbf**
version_number	**5**	**uimsbf**
current_next_indicator	**1**	**bslbf**
section_number	**8**	**uimsbf**
last_section_number	**8**	**uimsbf**
protocol_version	**8**	**uimsbf**
splice_command_type	**8**	**uimsbf**
if (splice_command_type == 0×01)		
splice_preroll()		
if (splice_command_type == 0×02)		
splice_execute()		
if (splice_command_type == 0×03)		
splice_schedule()		
CRC_32	**32**	**rpchof**
}		
}		

Table 4 – splice_preroll

Syntax	Bits	Mnemonic
splice_preroll() {		
splice_event_id	**32**	**bslbf**
out_of_network_indicator	**1**	**bslbf**
duration_flag	**1**	**bslbf**
reserved	**6**	**bslbf**
relative_splice_time()		
if (duration_flag == 1)		
break_duration()		
}		

Table 5 – splice_execute

Syntax	Bits	Mnemonic
splice_execute() {		
splice_event_id	**32**	**bslbf**
splice_event_cancel_indicator	**1**	**bslbf**
reserved	**7**	**bslbf**
if (splice_event_cancel_indicator == 0) {		
out_of_network_indicator	**1**	**bslbf**
program_splice_flag	**1**	**bslbf**
startup_delay_flag	**1**	**bslbf**
duration_flag	**1**	**bslbf**
reserved	**4**	**bslbf**
if (program_splice_flag == 1)		
splice_time()		
if (program_splice_flag == 0) {		
component_count	**8**	**uimsbf**
for (i=0; i<component_count; i++) {		
component_tag	**8**	**uimsbf**
splice_time()		
}		
}		
if (out_of_network_indicator == 0 &&		
startup_delay_flag == 1)		
startup_delay()		
if (duration_flag == 1)		
break_duration()		
}		
}		

Table 6 – splice_schedule

Syntax	Bits	Mnemonic
splice_schedule() {		
splice_count	**8**	**uimsbf**
for (i=0; i<splice_count; i++) {		
splice_event_id	**32**	**bslbf**
splice_event_cancel_indictor	**1**	**bslbf**
reserved	**7**	**bslbf**
if (splice_event_cancel_indicator == 0) {		
out_of_network_indicator	**1**	**bslbf**
program_splice_flag	**1**	**bslbf**
duration_flag	**1**	**bslbf**
reserved	**5**	**bslbf**
if (program_splice_flag == 1)		
splice_time()		
if (program_splice_flag == 0) {		
component_count	**8**	**uimsbf**
for (j=0; j<component_count; j++) {		
component_tag	**8**	**uimsbf**
splice_time()		
es_descriptor_count	**8**	**uimsbf**
for (k=0; k<es_descriptor_count; k++) {		
es_descriptor()		
}		
}		
}		
if (duration_flag)		
break_duration()		
}		
}		
}		

Table 7 – relative_splice_time

Syntax
relative_splice_time() {
time()
}

Table 8 – splice_time

Syntax
splice_time () {
time()
}

Table 9 – startup_delay

Syntax
startup_delay () {
pts_dts_time()
}

Table 10 -- break_duration

Syntax
break_duration() {
time()
}

Table 11 – time

Syntax	Bits	Mnemonic
time() {		
SMPTE_time_specified	**1**	bslbf
pts_dts_time_specified	**1**	bslbf
reserved	**6**	bslbf
if (SMPTE_time_specified == 1)		
SMPTE_time()		
if (pts_dts_time_specified == 1)		
pts_dts_time ()		
}		

Table 12 – SMPTE_time

Syntax	Bits	Mnemonic
SMPTE_time() {		
SMPTE12M_time_code	**64**	bslbf[1)]
reserved	**4**	**bslbf**
frame_rate	**4**	See ISO/IEC 13818-2 table 6-4
}		

[1)] The format of SMPTE12_time_code is specified by ANSI/SMPTE 12M, table 11, *Summation of VITC and LTC Code Word Bit Definitions*. Refer to that document for the definitions of the bit fields.

Table 13 – pts_dts_time

Syntax	Bits	Mnemonic
pts_dts_time() {		
reserved	**7**	bslbf
pts_dts_time	**33**	**uimsbf**
}		

Table 14 – stream_identifier_descriptor

Syntax	Bits	Mnemonic
stream_identifier_descriptor () {		
descriptor_tag	**8**	uimsbf
descriptor_length	**8**	uimsbf
component_tag	**8**	uimsbf
}		

7.2.1 splice_info_section syntax

7.2.1.1 table_id: This is an 8-bit field. Its value shall be [0×FE]. (This value is subject to confirmation.)

7.2.1.2 section_syntax_indicator: The section_syntax_indicator is a 1-bit field which determines if the section which follows contains valid splicing information or stuffing. This bit may be used to remove a splice_info_section from a transport stream packet without disturbing data for packet headers or other sections in the same packet. If splice_syntax_indicator is 0 following the section_length field, there shall be section_length number of stuffing bytes.

7.2.1.3 private_indicator: This is a 1-bit flag which shall be set to 0.

7.2.1.4 section_length: This is a 12-bit field specifying the number of remaining bytes of the section immediately following the section_length field, and including the CRC. The value in this field shall not exceed 4093.

7.2.1.5 stuffing_byte: A fixed 8-bit value equal to 1111 1111.

7.2.1.6 table_id_extension: This is a 16-bit field. Its value shall be 0×0000.

7.2.1.7 version_number: This 5-bit field is the version number of this section. The version_number shall be incremented by 1 modulo 32 when a change in the information carried within the splice_info_section occurs. When the current_next_indicator is set to 0, then the version_number shall be that of the next applicable section with the same table_id and section_number.

7.2.1.8 current_next_indicator: A 1-bit field, which when set to 1 indicates that the splice_info_section sent is currently applicable. When the current_next_indicator is set to 1, then the version_number shall be that of the currently applicable splice_info_section. When the bit is set to 0, it indicates that the splice_info_section sent is not yet applicable and will be the next splice_info_section with the same section_number and table_id to become valid.

7.2.1.9 section_number: This 8-bit field gives the number of the splice_info_section. The section_number of the first section in a splice information table shall be 0×00. The section_number shall be incremented by 1 with each additional section in this splice information table.

7.2.1.10 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 splice information table of which this section is a part.

7.2.1.11 protocol_version: An 8-bit unsigned integer which indicates the version number of the segment of the full table delivered with this section. The value of protocol_version shall be 0×00.

7.2.1.12 splice_command_type: An 8-bit unsigned integer assigned one of the values shown in table 15.

Table 15 – splice_command_type values

splice_command_type value	Command
0×00	Forbidden
0×01	Preroll
0×02	Execute
0×03	Schedule
0×04 -- 0×ff	Reserved

7.2.1.13 CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in ITU-T Rec. H.222.0 / ISO/IEC 13818-1 annex A after processing the entire splice_info_section.

7.2.2 splice_preroll(), splice_execute() and splice_schedule() syntax elements

7.2.2.1 splice_event_id: A 32-bit unique splice event identifier.

7.2.2.2 splice_event_cancel_indicator: A 1-bit flag that when set to 1 indicates that a previously sent splice event, identified by splice_event_id, has been cancelled.

7.2.2.3 out_of_network_indicator: A 1-bit flag. When set to 1, indicates that the splice event shall be away from the network feed and that the value of splice_time() or relative_splice_time() shall refer to an Out Point or a Program Out Point. When set to 0, the flag indicates that the splice event shall be used to switch back to the network and that the value of splice_time() or relative_splice_time() shall refer to an In Point or a Program In Point.

7.2.2.4 program_splice_flag: A 1-bit flag which when set to 1 indicates that splice_time() refers to a Program Splice Point, and when set to 0 indicates that splice_time() will be specified individually for each PID within the program that is intended to be spliced.

7.2.2.5 duration_flag: A 1-bit flag which when set to 1 indicates the presence of the break_duration() field.

7.2.2.6 startup_delay_flag: A 1-bit flag which indicates the presence of startup_delay().

7.2.2.7 component_count: An 8-bit unsigned integer that specifies the number of instances of elementary PID stream data in the loop that follows. Components are equivalent to elementary PID streams.

7.2.2.8 component_tag: An 8-bit value which identifies the elementary PID stream containing the Splice Point specified by the value of splice_time() which follows. The value shall be the same as the value used in the stream_identification_descriptor() to identify that elementary PID stream.

7.2.3 Syntax elements for splice_schedule() only

7.2.3.1 splice_count: An 8-bit unsigned integer that indicates the number of splice events specified in the loop that follows.

7.2.3.2 es_descriptor_count: An 8-bit unsigned integer that indicates the number of instances of elementary stream descriptors in the loop that follows.

7.2.3.3 es_descriptor(): This structure defines attributes of a given elementary stream. This structure is optional. The es_descriptor() is a descriptor which defines the elementary stream. It may be one of the descriptors specified in *table 2-39, Program and Program Element Descriptors*, of ITU-T Rec. H.222.0 / ISO/IEC 13818-1, as appropriate.

7.2.4 Syntax elements for specifying time: relative_splice_time(), splice_time(), break_duration(), startup_delay()

7.2.4.1 break_duration(): Duration of the commercial break(s). It is an optional field. It may be used to give the splicer an approximate idea of when the break will be over and when the network In Point will occur. When specified as SMPTE_time() format, the duration is given in minutes, seconds, and frames at the given frame rate. The hour bits are set to zero. The drop_frame_flag shall be 0. When specified as pts_dts_time() format, the duration is given as 90-kHz clock ticks. The value of break_duration shall be approximate to within 1 second.

7.2.4.2 relative_splice_time(): The time remaining before the splice event (see 7.4 for constraints on the value).

7.2.4.3 splice_time(): Time of the splice event (see 7.4 for constraints on the value).

7.2.4.4 startup_delay(): This is an optional field. It may be provided for In Points. If provided, the startup_delay is a duration expressed in pts_dts_time() format (90-kHz clock ticks). It is the difference in time between the original PCR value and the DTS_next_AU value carried in the In Point packet of the PCR_PID. Because of PCR jitter, the value of startup_delay approximates the actual startup delay when the bit stream arrives at the splicer.

7.2.5 Syntax elements for time(), SMPTE_time(), and pts_dts_time()

7.2.5.1 SMPTE_time_specified: A 1-bit flag indicating the presence of the SMPTE_time field.

7.2.5.2 pts_dts_time_specified: A 1-bit flag indicating the presence of the pts_dts_time field.

7.2.5.3 SMPTE12M_time_code: This is a 64-bit field. The format is given by ANSI/SMPTE 12M, *table 11, Summation of VITC and LTC Code Word Bit Definitions.* The 64 bits correspond to VITC bits, excluding the sync bits and CRC check, stored LSB first. An LTC time code source shall be mapped to the VITC format prior to use in this field. Extensions to ANSI/SMPTE 12M are being developed to provide for time zone, date, and clock time reference.

7.2.5.4 frame_rate: A 4-bit unsigned integer selected from *table 6-4, frame_rate_value* in ISO/IEC 13818-2. The value specifies the frame rate which should be used when interpreting the frame bits of SMPTE 12M time code.

7.2.5.5 pts_dts_time: A 33-bit field which indicates time in terms of ticks of a 90-kHz clock.

7.2.6 Syntax elements for stream_identifier_descriptor

7.2.6.1 descriptor_tag: An 8-bit unsigned integer which shall be set to 0×52. This value has been assigned by DVB.

7.2.6.2 descriptor_length: An 8-bit unsigned integer specifying the number of bytes of the descriptor immediately following the descriptor_length field. For this descriptor, descriptor_length shall be set to 0×01.

7.2.6.3 component_tag: An 8-bit unsigned integer which shall be selected to be unique for all the PID streams contained within a program and listed in the PMT for that program, according to 7.3.5.

7.3 Constraints on splice_info_section

7.3.1 The splice_info_section shall be carried in a PID stream which is specific to a program and referred to in the PMT. The splice_info_section PID shall be identified in the PMT by stream_type equal to [0×86]. (This value is subject to confirmation.)

7.3.2 The splice_info_sectio**n** carried in a PID stream referenced in a program's PMT shall contain only information about splice events which occur in that program.

7.3.3 A splice event shall be defined by a single value of splice_event_id.

7.3.4 The out_of_network_indicator shall be set to 1 to indicate an Out Point. It shall be set to 0 to indicate an In Point.

7.3.5 Each elementary PID stream shall be identified by a stream_identifier_descriptor carried in the PMT loop, one for each PID. The stream_identifier_descriptor shall carry an identifier called the component_tag, which uniquely corresponds to one PID stream among those contained within a program and listed in the PMT for that program. The format of the stream_identifier_descriptor is given in table 14.

7.3.6 Any splice_event_id which is sent in a splice_info_section with splice_command_type equal to 0×01 or equal to 0×03 must be sent again prior to the event using splice_command_type equal to 0×02.

7.3.7 Splice information sections with different values of splice_command_type shall not be sent within the same transport stream packet.

7.3.8 When specifying splice_time(), if both SMPTE_time_specified is set to 1 and pts_dts_time_specified is set to 1, a correspondence between SMPTE time code and the 90-kHz clock shall be established. This correspondence shall remain in effect until a new correspondence is established in this manner.

7.3.9 When specifying splice_time(), if both SMPTE_time_specified and pts_dts_time_specified are set to 0, then the time shall be interpreted as the current time. The splice shall occur at the next available Splice Point(s).

7.4 Constraints on the interpretation of time()

7.4.1 Constraints on relative_splice_time for splice_preroll()

For splice_command_type equal to 0×01 (preroll), the following constraints on the data elements of relative_splice_time() shall apply:

7.4.1.1 The value given in time() is interpreted as the relative time from when the splice_info_section arrives until the presentation time of the PCR_PID Splice Point.

7.4.1.2 The value given in time() is considered to be approximate to within ± one second.

7.4.1.3 The syntax element relative_splice_time() may be expressed as either SMPTE_time() format or as pts_dts_time() format. When specified as SMPTE_time() format, time is expressed as seconds and frames at the given frame rate. The hours and minutes bits are set to 0. The drop_frame_flag shall be 0. When specified as pts_dts_time() format, time is expressed as 90-kHz clock ticks.

7.4.2 Constraints on splice_time for splice_execute

For splice_command_type equal to 0×02 (execute), the following constraints on splice_time() shall apply:

7.4.2.1 The flag pts_dts_time_specified shall be 1. A valid value shall be carried in the pts_dts_time field in accordance with the constraints below. SMPTE_time() format may be specified in addition to pts_dts_time() format, but it is not required.

7.4.2.2 For specifying a Program Out Point, i.e., when the program_splice_flag equals 1, the value of pts_dts_time shall equal the DTS_next_AU value of the Out Point packet of the PCR_PID.

7.4.2.3 For specifying an Out Point in an elementary PID stream, i.e., when the program_splice_flag equals 0, the value of pts_dts_time shall equal the DTS_next_AU value of the Out Point packet of the elementary PID stream which corresponds to the value of component_tag.

7.4.2.4 For specifying a Program In Point, i.e., when the program_splice_flag equals 1, then the value of pts_dts_time shall equal the DTS_next_AU value of the In Point packet of the PCR_PID.

7.4.2.5 For specifying an In Point in an elementary PID stream, i.e., when the program_splice_flag equals 0, then the value of pts_dts_time shall equal the DTS_next_AU value of the In Point packet of the elementary PID stream which corresponds to the value of component_tag.

7.4.3 Constraints on splice_time for splice_schedule

For splice_command_type equal to 0×03 (schedule), the following constraints on the data elements of splice_time() shall apply:

7.4.3.1 The time base used to specify splice_time() is assumed to be continuous and to wrap around once per 24-hour day.

7.4.3.2 The syntax element splice_time() may be expressed in either SMPTE_time() format or pts_dts_time() format. If specified as pts_dts_time() format, the splicer may assume that the 90-kHz clock is a continuous clock with no discontinuities (except one per day when the clock wraps around).

Annex A (informative)
Splice Points and application implications

A.1 Frequency of Splice Points

The frequency of Splice Points is not specified by this standard. It is envisioned that in video and audio many Splice Points may exist. In some applications, such as a studio environment where low-delay and flexibility in switching are important, Splice Points might occur as frequently as every frame (in an all I-frame environment). In a distribution environment, Splice Points might occur at regular intervals during normal program playout and more frequently surrounding break times. Since Out Points may be specified at either I or P frame boundaries (in presentation order), they may occur more frequently than In Points (which may only occur preceding I frames).

A.2 PCR, PTS/DTS, and timing discontinuities

Splicing devices are responsible for proper handling of PCR, PTS/DTS, and DTS_next_AU. If a PCR discontinuity results from the splice, the splicing device sets the discontinuity_indicator in the adaptation_field of the transport_packet. ITU-T Rec. H.222.0 / ISO/IEC 13818-1 requires that time-base discontinuities be handled as follows:

> "Prior to the occurrence of a system time-base discontinuity, the first byte of a transport stream packet which contains a PTS or DTS which refers to the new system time-base shall not arrive at the input of the T-STD. After the occurrence of a system time-base discontinuity, the first byte of a transport stream packet which contains a PTS or DTS which refers to the previous system time-base shall not arrive at the input of the T-STD."

Splicing devices which introduce time-base discontinuities must ensure that the output stream meets these requirements.

Alternatively, splicing devices may restamp transport stream packets. Splicing devices that restamp PTS, DTS, and DTS_next_AU must take care to adjust the references to those values in the splice_info_section. Splicers which restamp should take care with altering DTS_next_AU values. Some splicing devices may rely on the relative difference between PCR and DTS_next_AU to calculate the decoding delay at a Splice Point. Restamping devices should take care to preserve the proper relationship between these two values at both In Points and Out Points.

A.3 PMT/PID assignments

ITU-T Rec. H.222.0 / ISO/IEC 13818-1, clause 2.4.4, specifies that all transport streams include valid program specific information (PSI) packets, which describe the contents of the transport stream. If a splice causes the contents of a transport stream, including the PID values within the transport stream, to change, then the changes must be reflected in valid PSI packets. Splicing devices are responsible for sending any alterations to the PMT required to accommodate changes in the number of PIDs after a splice.

However, in order to prevent commercial killing devices from taking advantage of changes in PSI, systems with splicing are encouraged to avoid changes to the PMT by reusing existing video and audio PIDs after a splice. The new MPEG-2 program may be preprocessed to match the old program. Example preprocessing functions include, but are not exclusive to: PID remapping, PID duplication, and PID dropping.

– **PID remapping**: If the PID values of the new program differ from those of the old program, the new program PID values are changed to match those in the old program.

– **PID duplication:** If the new program does not have corresponding elementary streams for all of the elementary streams in the old program, appropriate elementary stream(s) in the new program can be duplicated and assigned the missing PID values. For example, if a two-language program is spliced to a one-language program, the single audio track can be copied, using the PID value of the missing audio track.

– **PID dropping:** If the new program has more elementary streams than the old program, those elementary streams without corresponding elementary streams in the old program may be dropped from the new program.

Note that these PID preprocessing functions, as well as the functions of PSI generation and monitoring, are not within the scope of this standard.

A.4 Closed captioning

Closed captioning may be carried either in the video picture user_data (as in ATSC A/53) or in a separate PID. In the case of carrying closed captioning in user_data, the bit stream must adhere to constraint 5.3.2.5. In the case of a separate PID, the closed captioning data are treated as any other time-stamped PID stream and the bit stream must adhere to constraints 5.1, 5.2.1, and 5.3.1.

A.5 Statistical multiplexing

The definition of splice points specified in this standard can facilitate splicing in stat mux or other VBR environments. In addition to meeting the Splice Point constraints, stat mux environments must allocate the bandwidth of a total channel. Bandwidth allocation is not within the scope of this standard. For example, operational practice may be established as to the available bandwidth for local advertisements. Signaling of this information may be out of band or fall within the scope of other standards.

A.6 Encrypted bit stream splicing

Systems that do not decrypt before splicing would need to address additional practical considerations such as ensuring that the syntax elements required by this specification are never encrypted or are made available through other methods best determined according to the overall design of a given system.

An additional practical consideration that should be addressed in encrypted bit stream systems that perform bit stream splicing is the transmission of decryption keys (which themselves are encrypted), commonly called entitlement control messages (ECMs). These ECMs must be transmitted in advance of their use, so that a receiver has enough time to decrypt an ECM before using the resulting descrambling key (or keyword). It may be difficult for a system that splices between encrypted bit streams to ensure that the proper ECMs for each stream arrive far enough in advance to be of use.

Annex B (informative)
Bibliography

SMPTE 302M-1998, Television — Linear PCM Digital Audio in an MPEG-2 Transport Stream

ATSC A/52, Digital Audio Compression (AC-3)

ATSC A/54, Guide to the Use of the ATSC Digital Terrestrial Television Standard

ETSI/ETR 154:1997-09, Digital Video Broadcasting (DVB); Implementation Guidelines for the Use of MPEG-2 Systems, Video and Audio in Satellite, Cable and Terrestrial Broadcasting Applications

ISO/IEC 13818-3:1998, Information Technology — Generic Coding of Moving Pictures and Associated Audio Information — Part 3: Audio