

I n t e r n a t i o n a l T e l e c o m m u n i c a t i o n U n i o n

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

J.181

(01/2014)

SERIES J: CABLE NETWORKS AND TRANSMISSION
OF TELEVISION, SOUND PROGRAMME AND OTHER
MULTIMEDIA SIGNALS

Digital transmission of television signals

Digital program insertion cueing message for cable television systems

Recommendation ITU-T J.181

Recommendation ITU-T J.181

Digital program insertion cueing message for cable television systems

Summary

Recommendation ITU-T J.181 supports frame accurate signalling of events in MPEG-2 transport streams along with associated descriptive data. This Recommendation supports the splicing of MPEG-2 transport streams for the purpose of digital program insertion, which includes advertisement insertion and insertion of other content types. An in-stream messaging mechanism is defined to signal splicing and insertion opportunities and it is not intended to ensure seamless splicing. As such, this Recommendation does not specify the splicing method used or constraints applied to the streams being spliced, nor does it address constraints placed on splicing devices.

A fully compliant MPEG-2 transport stream (either multi program transport stream (MPTS) or single program transport stream (SPTS)) is assumed. No further constraints beyond the inclusion of the defined cueing messages are placed upon the stream.

This Recommendation specifies a technique for carrying notification of upcoming Splice Points and other timing information 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 one or more PID(s) referred to by that program's program map table (PMT). In this way, splice event notification can pass through most transport stream remultiplexers without need for special processing.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T J.181	2001-03-09	9	11.1002/1000/5386-en
1.1	ITU-T J.181 (2001) Amd. 1	2003-04-04	9	11.1002/1000/6355-en
2.0	ITU-T J.181	2004-06-29	9	11.1002/1000/7379-en
3.0	ITU-T J.181	2014-01-13	9	11.1002/1000/12102-en

* To access the Recommendation, type the URL <http://handle.itu.int/> in the address field of your web browser, followed by the Recommendation's unique ID. For example, <http://handle.itu.int/11.1002/1000/11830-en>.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2014

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

	Page
1 Scope	1
2 References.....	1
3 Definitions	2
3.1 Terms defined elsewhere	2
3.2 Terms defined in this Recommendation.....	2
4 Abbreviations and acronyms	4
5 Conventions	5
6 Introduction	5
6.1 Splice points (Informative).....	5
6.2 Program splice points (Informative).....	6
6.3 Splice events (Informative)	6
6.4 Content storage considerations (Informative)	6
6.5 PID selection	7
6.6 Message flow (Informative)	8
7 XML notational conventions	8
7.1 Normative XML schema	8
7.2 Unknown/unrecognized/unsupported XML elements and attributes	9
7.3 Element order	9
8 PMT descriptors	9
8.1 Registration descriptor.....	9
8.2 Cue identifier descriptor	10
8.3 Stream identifier descriptor	11
9 Splice information table.....	12
9.1 Overview	12
9.2 Splice info section	13
9.3 Splice commands.....	18
9.4 Constraints	29
10 Splice descriptors.....	31
10.1 Overview	31
10.2 Splice descriptor	31
10.3 Specific splice descriptors	33
11 Encryption	46
11.1 Overview	46
11.2 Fixed key encryption	46
11.3 Encryption algorithms	46
12 SCTE 35 XML elements and types	47
12.1 Ext element.....	47

	Page
12.2 PTSType	48
Appendix I – CableLabs metadata identifier	49
Bibliography.....	50

iv

Recommendation ITU-T J.181

Digital program insertion cueing message for cable television systems

1 Scope

This Recommendation supports frame accurate signalling of events in MPEG-2 transport streams along with associated descriptive data. This Recommendation supports the splicing of MPEG-2 transport streams for the purpose of Digital Program Insertion, which includes advertisement insertion and insertion of other content types. An in-stream messaging mechanism is defined to signal splicing and insertion opportunities and it is not intended to ensure seamless splicing. As such, this Recommendation does not specify the splicing method used or constraints applied to the streams being spliced, nor does it address constraints placed on splicing devices.

A fully compliant MPEG-2 transport stream (either multi program transport stream (MPTS) or single program transport stream (SPTS)) is assumed. No further constraints beyond the inclusion of the defined cueing messages are placed upon the stream.

This Recommendation specifies a technique for carrying notification of upcoming Splice Points and other timing information 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 one or more packet identifiers (PIDs) referred to by that program's program map table (PMT). In this way, splice event notification can pass through most transport stream remultiplexers without need for special processing.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- [ITU-T H.222.0] Recommendation ITU-T H.222.0 (2012) | ISO/IEC 13818-1:2013, *Information Technology – Generic coding of moving pictures and associated audio information: Systems*.
- [IETF RFC 3986] IETF RFC 3986 (2005), *Uniform Resource Identifier (URI): Generic Syntax*, www.ietf.org/rfc/rfc3986.txt/
- [ISO 15706-2] ISO 15706-2:2007 – *Information and documentation – International Standard Audiovisual Number (ISAN) – Part 2: Version identifier*.
- [ATSC A/57B] ATSC A/57B – *ATSC Standard: Content Identification and Labeling for ATSC Transport Document A/57B*, 26 May 2008.
- [FIPS PUB 46-3] FIPS PUB 46-3, *Data Encryption Standard (DES)*, 25 October 1999.
- [FIPS PUB 81] FIPS PUB 81, *DES Modes of Operation*, 2 December 1980.
- [SMPTE 330M] SMPTE 330M-2004 – *SMPTE Standard for Television – Unique Material Identifier*.
- [XML] W3C, *Extensible Markup Language (XML) 1.0 (Fourth Edition)*, W3C Recommendation 16 August 2006, <http://www.w3.org/TR/2006/REC-xml-20060816/>

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 pointer_field [ITU-T H.222.0]: The first byte of a transport packet payload, required when a section begins in that packet.

3.1.2 presentation unit (PU) [ITU-T H.222.0]: A decoded audio access unit or a decoded picture.

3.1.3 uniform resource identifier (URI): See [IETF RFC 3986].

3.1.4 extensible markup language (XML): See [XML].

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 access unit: A coded representation of a presentation unit.

3.2.2 advertisement (also called "ad"): An inducement to buy or patronize. As used in the cable industry, usually with a duration under 2 minutes (sometimes called "short-form" content).

3.2.3 analogue cue tone: In an analogue system, a signal that is usually either a sequence of dual tone multi-frequency (DTMF) tones or a contact closure that denotes to ad insertion equipment that an advertisement avail is about to begin or end.

3.2.4 avail: Time space provided to cable operators by cable programming services during a program for use by the community antenna television (CATV) operator; the time is usually sold to local advertisers or used for channel self promotion.

3.2.5 bit stream format: An encoding of information resulting in a compliant MPEG-2 transport stream.

3.2.6 break: Avail or an actual insertion in progress.

3.2.7 chapter: A short section of a longer program, usually situated to permit a viewer to easily locate a scene or section of the program.

3.2.8 component splice mode: A mode of the cueing message whereby the `program_splice_flag` is set to '0' and indicates that each PID/component that is intended to be spliced will be listed separately by the syntax that follows. Components not listed in the message are not to be spliced.

3.2.9 content: Generic term for television material, either advertisements or programs.

3.2.10 cueing message: See message.

3.2.11 event: A splice event or a viewing event.

3.2.12 in point: A point in the stream, suitable for entry, that lies on an elementary presentation unit boundary. An in point is actually between two presentation units rather than being a presentation unit itself.

3.2.13 in stream device: A device that receives the transport stream directly and is able to derive timing information directly from the transport stream.

3.2.14 message: In the context of this Recommendation a message is the contents of any `splice_info_section`.

- 3.2.15 multi program transport stream:** A transport stream with multiple programs.
- 3.2.16 out of stream device:** A device that receives the cue message from an in stream device over a separate connection from the transport stream. An out of stream device does not receive or pass the transport stream directly.
- 3.2.17 out point:** A point in the stream, suitable for exit, that lies on an elementary presentation unit boundary. An out point is actually between two presentation units rather than being a presentation unit itself.
- 3.2.18 packet identifier (PID):** A unique 13-bit value used to identify the type of data stored in the packet payload.
- 3.2.19 payload_unit_start_indicator:** A bit in the transport packet header that signals, among other things, that a section begins in the payload that follows.
- 3.2.20 PID stream:** All the packets with the same PID within a transport stream.
- 3.2.21 presentation time:** The time that a presentation unit is presented in the system target decoder.
- 3.2.22 presentation unit:** A decoded Audio Access Unit or a decoded picture (see [b-ITU-T H.262]).
- 3.2.23 program:** A collection of video, audio, and data PID streams that share a common program number within an MPTS (see [ITU-T H.222.0]). As used in the context of the segmentation descriptor, a performance or informative presentation broadcast on television, typically with a duration over 5 minutes (sometimes called "long-form" content).
- 3.2.24 program in point:** A group of PID stream in points that correspond in presentation time.
- 3.2.25 program out point:** A group of PID stream out points that correspond in presentation time.
- 3.2.26 program splice mode:** A mode of the cueing message whereby the `program_splice_flag` is set to '1' and indicates that the message refers to a Program Splice Point and that all PIDs/components of the program are to be spliced.
- 3.2.27 program splice point:** A program in point or a program out point.
- 3.2.28 receiving device:** A device that receives or interprets sections conforming to this Recommendation. Examples of these devices include splicers, ad servers, segmenters and satellite receivers.
- 3.2.29 registration descriptor:** Carried in the program map table (PMT) of a program to indicate that, when signalling splice events, `splice_info_sections` shall be carried in a PID stream within this program. The presence of the Registration Descriptor signifies a program's compliance with this Recommendation.
- 3.2.30 reserved:** The term "reserved", when used in the clauses defining the coded bit stream, indicates that the value may be used in the future for extensions to the Recommendation. Unless otherwise specified, all reserved bits shall be set to '1' and this field shall be ignored by receiving equipment.
- 3.2.31 segment:** Either a Program, a Chapter, a Provider Advertisement, a Distributor Advertisement, or an Unscheduled Event as listed in Table 10-8, `segmentation_type_id`.
- 3.2.32 single program transport stream:** A transport stream containing a single MPEG program.
- 3.2.33 splice event:** An opportunity to splice one or more PID streams.

3.2.34 splice immediate mode: A mode of the cueing message whereby the splicing device shall choose the nearest opportunity in the stream, relative to the splice_info_table, to splice. When not in this mode, the message gives a "pts_time" that, when modified by pts_adjustment, gives a presentation time for the intended splicing moment.

3.2.35 splice point: A point in a PID stream that is either an out point or an in point.

3.2.36 viewing event: A television program or a span of compressed material within a service; as opposed to a splice event, which is a point in time.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ad	advertisement
ADI	Asset Distribution Interface
Ad-ID	Advertisement Identifier
ASCII	American Standard Code for Information Interchange
ATSC	Advanced Television Systems Committee
bslbf	Bit string, left bit first, where left is the order in which bit strings are written.
CA	Conditional Access
CATV	Community Antenna Television
CBC	Cipher Block Chaining
CRC	Cyclic Redundancy Check
DTMF	Dual Tone Multi-Frequency
DES	Data Encryption Standard
DTS	Decoding Time-Stamp
DVB	Digital Video Broadcast
ECB	Electronic Codebook
EDE	Encrypt-Decrypt-Encrypt
EIDR	Entertainment Identifier Registry association
FIPS	Federal Information Processing Standard
Id	Identifier
InfoSet	Information Set
ISAN	International Standard Audiovisual Number
ISCI	Industry Standard Commercial Identifier
MID	Multiple UPID
MPEG	Moving Picture Experts Group
MPTS	Multi Program Transport Stream
MPU	Managed Private UPID
PCR	Program Clock Reference
PES	Packetized Elementary Stream

PID	Packet Identifier
PMT	Program Map Table (see [ITU-T H.222.0])
PTS	Presentation Time Stamp (see [ITU-T H.222.0])
rpchof	Remainder polynomial coefficients, highest order first
SCTE	Society of Cable Telecommunications Engineers
SMPTE	Society of Motion Picture & Television Engineers
SPTS	Single Program Transport Stream
STC	System Time Clock
TI	Turner Identifier
TID	Tribune Identifier
uimsbf	Unsigned integer, most significant bit first
UMID	Unique Material Identifier
URI	Uniform Resource Identifier (see [IETF RFC 3986])
UTC	Coordinated Universal Time
V-ISAN	Version-ISAN
XML	Extensible Markup Language

5 Conventions

Deprecated: Use is permissible for legacy purposes only. Deprecated features may be removed from future versions of this Recommendation. Implementations should avoid use of deprecated features.

6 Introduction

6.1 Splice points (Informative)

To enable the splicing of compressed bit streams, this Recommendation defines splice points. Splice points in an MPEG-2 transport stream provide opportunities to switch elementary streams from one source to another. They indicate a place to switch or a place in the bit stream where a switch can be made. Splicing at such splice points may or may not result in good visual and audio quality. That is determined by the performance of the splicing device.

Transport streams are created by multiplexing packet identifier (PID) streams. In this Recommendation, 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 acceptable to enter, from a splicing standpoint. Out points are places where it is acceptable to exit the bit stream. The grouping of in points of individual PID streams into program in points in order to enable the switching of entire programs (video with audio) is 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 elementary stream presentation units. Out points and in points are not necessarily transport packet aligned and are not necessarily PES packet aligned. An out point and an in point may be co-located; that is, a single presentation unit boundary may serve as both a safe place to leave a bit stream and a safe place to enter it.

The output of a simple switching operation will contain access unit data from one stream up until its out point followed by data from another stream starting with the first access unit following an in point. More complex splicing operations may exist whereby data prior to an out point or data after

an in point are modified by a splicing device. Splicing devices may also insert data between one stream's out point and the other stream's in point. The behaviour of splicing devices will not be specified or constrained in any way by this Recommendation.

6.2 Program splice points (Informative)

Program in points and program out points are sets of PID stream in points or out points that correspond in presentation time.

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 during a program.

This Recommendation defines two ways of signalling which splice points within a program are to be spliced. A `program_splice_flag`, when true, denotes that the program splice mode is active and that all PIDs of a program may be spliced (the splice information table PID is an exception; splicing or passage of these messages is beyond the scope of this Recommendation). A `program_splice_flag`, when false, indicates that the component splice mode is active and that the message will specify unambiguously which PIDs are to be spliced and may give a unique splice time for each. This is required to direct the splicing device to splice or not to splice various unspecified data types as well as video and audio.

While this Recommendation allows for a unique splice time to be given for each component of a program, it is expected that most component splice mode messages will utilize one splice time (a default splice time) for all components as described in clause 9. The facility for optionally specifying a separate splice time for each component is intended to be used when one or more components differ significantly in their start or stop time relative to other components within the same message. An example would be a downloaded applet that must arrive at a set-top box several seconds prior to an advertisement.

6.3 Splice events (Informative)

This Recommendation provides a method for in-band signalling of splice events using splice commands to downstream splicing equipment. Signalling a splice event identifies which splice point within a stream to use for a splice. A splicing device may choose to act or not act upon a signalled event (a signalled event should be interpreted as an opportunity to splice; not a command). A splice information table carries the notice of splice event opportunities. Each signalled splice event is analogous to an analogue 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 Recommendation establishes that the splice information table is carried on a per-program basis in one or more PID stream(s) with a designated `stream_type`. The program's splice information PID(s) are designated in the program's program map table (PMT). 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 that carry splice information tables. Remultiplexers or splicers may use this `stream_type` field to drop splice information prior to sending the transport stream to the end-user device.

The cue injection equipment may send messages at intervals that do not indicate a splice point to be used as heartbeat messages which help ensure the proper operation of the system. This could be performed by periodically issuing `splice_null()` messages or by sending encrypted `splice_insert` messages generated with a key that is not distributed. Since cues are currently sent twice per hour on a typical network, an average interval of 5 minutes would be a reasonable interval. If a message was not received in a 10 minute interval, a receiving device could alarm an operator to a possible system malfunction (such behaviour would be implementer dependent).

6.4 Content storage considerations (Informative)

The requirements for identifier uniqueness are written expecting the content to be playing in real time. If the content is stored, then the playback of the content does not place requirements upon the playback equipment to alter any of these identifiers (such as `splice_event_id` or `segmentation_event_id`). Downstream equipment parsing the identifiers should keep this in mind and, if applicable, rely upon other confirming information before reacting adversely to a seeming violation of the identifier uniqueness requirements of this Recommendation.

This Recommendation provides optional tools to assist with segmenting content into shorter sections which may be either chapters or advertisements. See clause 10.3.3.

6.5 PID selection

6.5.1 PID selection (Normative)

Splice Information can be carried in multiple PIDs. The maximum number of PIDs that can carry splice information shall not exceed 8. These PIDs can be either in the clear (where the transport scrambling_control bits are set to '00') or scrambled by a control access (CA) system. Each cue message PID may include the `cue_identifier_descriptor` defined in clause 8.2 to describe the splice commands included in the PID. When multiple PIDs are used to carry splice information, the first cue message PID in the program map table (PMT) shall only contain the splice command types 0x00 (`splice_null`), 0x04 (`splice_schedule`) and 0x05 (`splice_insert`). In addition, the `splice_event_id` shall be unique in all splice information PIDs within the program.

6.5.2 PID selection (Informative)

While the use of multiple cue message PIDs is an allowed practice, it should be noted that not all equipments may respond in the same manner to a stream that contains multiple cue message PIDs. Some equipment may limit the number of PIDs that the equipment can pass or receive. If a system utilizes multiple PIDs through various devices with the intention of reaching the set-top, it is suggested that thorough end-to-end testing be performed.

In many systems, the delivery of PIDs that carry splice information beyond the ad insertion equipment in the head-end is not desired. In these systems, the splicing or multiplexing device will drop any or all of these messages (PIDs) so they will not be delivered to the set-top. In other systems it may selectively pass certain PIDs to the set-top to enable set-top functionality. A third possibility is that the splicing or multiplexing device will aggregate the multiple PIDs that carry splice information into a single PID to handle downstream, set-top, issues with multiple PIDs. The action of ignoring or passing the message is recommended to be a user provisioned item, with a suitable default behaviour chosen by the implementer.

The default operation if a splicing or multiplexing device receives a PID based on this specification with the scrambling bits set in the header should be to drop that PID and not pass it through to the output. This ideally should be a user provisioned operation, as in some instances this PID may be descrambled by a downstream device.

The delivery of messages outside of the receive location to the customer may be based on business agreements. An example would be that one programmer wants the cue messages passed to set-tops to enable a targeted advertising method while a different programmer insists that the messages be dropped to ensure that a commercial killer may not utilize the messages.

When multiple splicing PIDs are identified in the PMT, the splicing device should process all of these PIDs. If the `cue_identifier_descriptor` is utilized, the splicing or multiplexing device may use that information to be more selective of the PIDs on which it will act.

Some possible reasons for utilizing multiple PIDs for this message include selective delivery of cue messages for different tiers of advertising or for separating cue messages from segmentation

messages. While one possible method of handling these issues is to use the encryption methods built in to this Recommendation, many delivery mechanisms can support conditional delivery by PID in a secure fashion. The delivery equipment (Satellite transmitter/receiver, remultiplexer) may PID filter the stream to only allow one or a small number of the PIDs to be passed in-stream. This method may be used to create multiple programs in the feed based on entitlement. The decision to use one or more PIDs will be based on the security required and the CA hardware available on the system.

6.6 Message flow (Informative)

The messages described in this Recommendation can originate from multiple sources. They are designed to be sent in-stream to downstream devices. The downstream devices may act on the messages or send them to a device that is not in-stream to act upon them. An example would be a splicer communicating via SCTE 30 protocol to an ad server (See [b-SCTE 30]). The in-stream devices could pass the messages to the next device in the transmission chain, or they could, optionally, drop the messages. Implementers are urged to make these decisions user provisioned, rather than arbitrarily hard-coded.

Any device that restamps pcr/pts/dts and that passes these cue messages to a downstream device should modify the pts_time field or the pts_adjustment field in the message in all PIDs conforming to this Recommendation. Modifying the pts_adjustment field is preferred because the restamping device will not have to be knowledgeable of the pts_time field that may occur in multiple commands (and possibly in future commands).

The bandwidth_reservation() message is intended as a message used on a closed path from a satellite origination system (encoder) to a receiver. It is also intended that this message will be dropped (replaced by a NULL packet) by the receiver, but this is not required. Should this message reach an in-stream device (e.g., a splicer) the message should not be forwarded to an out-of-stream device (e.g., Ad server) and can either be ignored or passed by an in-stream device. The action of ignoring or passing the message is recommended to be a user provisioned item, with a suitable default behaviour chosen by the implementer.

7 XML notational conventions

7.1 Normative XML schema

Descriptions of elements and attributes are normative and, when combined with the normative XML schema document (provided separately), comprise the full normative schema specification. Unless otherwise specified, the normative text and values assigned to elements or attributes in this Recommendation shall be constrained by the bit stream equivalent field.

Non-normative schema illustrations and instance examples are included herein for informational purposes only. Any real or implied usage, semantics, or structure indicated by the schema illustrations and examples shall not be considered part of the specification.

No XML documents representing the structures defined in the schema are considered conformant unless they are valid according to the schema document. Additionally, other SCTE 35 [b-SCTE 35] standard normative parts may impose additional rules or restrictions that shall be adhered to in order for XML documents to be considered conformant to those parts.

In the case where this Recommendation and the normative schema document (i.e., the separately provided XML 'xsd' file) conflict, this Recommendation shall take precedence over the XML schema document.

The inclusion of a normative XML schema document does not require or imply the specific use of the schema nor a requirement that an XML document be validated.

If the SCTE 35 schema is used in combination with other schemas, it is recommended to utilize the namespace prefix of "SCTE35". For example, SCTE35:SpliceInsertType to reference an SCTE 35 SpliceInsert Type.

7.2 **Unknown/unrecognized/unsupported XML elements and attributes**

Generally, unknown, unrecognized or unsupported XML elements and attributes contained within SCTE 35 elements should be ignored during XML document processing. Specifically, these are elements or attributes which the implementation does not understand or expect. XML parsers that encounter elements or attributes which are prohibited by a namespace should include exception handling

7.3 **Element order**

Element order is constrained by the schemas and must be preserved throughout processing of the XML document. In particular, the order of elements affects the end result of the processing. Consequently, an implementation failing to preserve the order may cause incorrect processing results. Subsequently, the process of producing an abstract XML information set (InfoSet) from a concrete XML document, e.g., by parsing it, shall always result in the same abstract InfoSet, with the same element order per XML InfoSet. (See [XMLInfoSet]) for additional information). Any intermediary processing may enhance the XML document but it shall not alter the abstract InfoSet element order (i.e., the XML elements comprising the document shall stay in document order).

8 **PMT descriptors**

8.1 **Registration descriptor**

The registration descriptor ([ITU-T H.222.0], Table 2-51 – Registration descriptor, clause 2.6.8) is defined to identify unambiguously the programs that comply with this Recommendation. The registration descriptor shall be carried in the program_info loop of the PMT for each program that complies with this Recommendation. It must reside in all PMTs of all complying programs within a multiplex. The presence of the registration descriptor also indicates that, when signalling splice events, splice_info_sections shall be carried in one or more PID stream(s) within this program.

Presence of this registration descriptor in the PMT signals the following:

- 1) The program elements do not include the splice information table defined by [b-SMPTE 312].
- 2) The only descriptors that can be present in the ES_descriptor_loop of the PMT for the PID(s) that carry the splice_information_table are those that are defined in this specification or user private descriptors.

Note that this descriptor applies to the indicated program and not to the entire multiplex. The content of the registration descriptor is specified in Table 8-1 and below:

Table 8-1 – registration_descriptor()

Syntax	Bits	Mnemonic
registration_descriptor() {		
descriptor_tag	8	uimbsf
descriptor_length	8	uimbsf
SCTE_splice_format_identifier	32	uimbsf
}		

8.1.1 Semantic definition of fields in registration descriptor

descriptor_tag – The descriptor_tag is an 8-bit field that identifies each descriptor. For registration purposes, this field shall be set to 0x05.

descriptor_length – The descriptor_length is 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 0x04.

SCTE_splice_format_identifier – SCTE has assigned a value of 0x43554549 (ASCII "CUEI") to this 4-byte field to identify the program (within a multiplex) in which it is carried as complying with this Recommendation.

8.2 Cue identifier descriptor

The cue_identifier_descriptor may be used in the PMT to label PIDs that carry splice commands so that they can be differentiated as to the type or level of splice commands they carry. The cue_identifier_descriptor, when present, shall be located in the elementary descriptor loop. If the cue_identifier_descriptor is not utilized, the stream may carry any valid command in this specification.

Table 8-2 – cue_identifier_descriptor()

Syntax	Bits	Mnemonic
cue identifier descriptor() {		
descriptor_tag	8	uimbsf
descriptor_length	8	uimbsf
cue_stream_type	8	uimbsf
}		

8.2.1 Semantic definition of fields in cue identifier descriptor

descriptor_tag – The descriptor_tag is an 8-bit field that identifies each descriptor. For cue_identifier_descriptor, this field shall be set to 0x8A.

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 descriptor, descriptor_length shall be set to 0x01.

cue_stream_type – This 8-bit field is defined in the following table.

Table 8-3 – cue_stream_type values

cue_stream_type	PID usage
0x00	splice_insert, splice_null, splice_schedule
0x01	All Commands
0x02	Segmentation
0x03	Tiered Splicing
0x04	Tiered Segmentation
0x05-0x7f	Reserved
0x80-0xff	User Defined

8.2.2 Description of cue_stream_type usage

0x00 – splice_insert, splice_null, splice_schedule – Only these cue messages are allowed in this PID stream. There shall be a maximum of one PID identified with this cue_stream_type. If this PID exists, it shall be the first stream complying with this Recommendation in the PMT elementary stream loop.

0x01 – All Commands – Default if this descriptor is not present. All messages can be used in this PID.

0x02 – Segmentation – This PID carries the time_signal command and the segmentation descriptor. It may also carry all other commands if needed for the application, but the primary purpose is to transmit content segmentation information.

x03 – Tiered Splicing – Tiered Splicing refers to an insertion system where the operator provides different inserted program possibilities in a given avail for different customers. The physical and logical implementation may be done in several different manners, some of them outside the scope of this Recommendation.

0x04 – Tiered Segmentation – Tiered Segmentation refers to a system where the operator provides different program segmentation possibilities for different customers. The physical and logical implementation may be done in several different manners, some of them outside the scope of this Recommendation.

0x05-0x7F – Reserved for future extensions to this Recommendation.

0x80-0xFF – User defined range.

8.3 Stream identifier descriptor

The stream identifier descriptor may be used in the PMT to label component streams of a service so that they can be differentiated. The stream identifier descriptor shall be located in the elementary descriptor loop following the relevant ES_info_length field. The stream identifier descriptor shall be used if either the program_splice_flag or the program_segmentation_flag is zero. If stream identifier descriptors are used, a stream identifier descriptor shall be present in each occurrence of the elementary stream loop within the PMT and shall have a unique component tag within the given program.

Table 8-4 – stream_identifier_descriptor()

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

8.3.1 Semantic definition of fields in stream identifier descriptor

descriptor_tag – The descriptor_tag is an 8-bit field that identifies each descriptor. For stream_identifier_descriptor, this field shall be set to 0x52.

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 descriptor, descriptor_length shall be set to 0x01.

component_tag – This 8-bit field identifies the component stream for associating it with a description given in a component descriptor. Within a program map section each stream identifier descriptor shall have a different value for this field.

9 Splice information table

9.1 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, examples of splice events 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 table 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. The splice_info_section is carried on one or more PID stream(s) with the PID(s) declared in that program's PMT.

A splice event indicates the opportunity to splice one or more elementary streams within a program. Each splice event is uniquely identified 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 may be given to execute the splice event at specified splice points. These three types of messages are sent via the splice_info_section. The splice_command_type field specifies the message being sent. Depending on the value of this field, different constraints apply to the remaining syntax.

The following command types are specified: splice_null(), splice_schedule(), splice_insert(), time_signal() and bandwidth_reservation(). If the receiving device does not support a command, it can ignore the entire splice_info_section.

The splice_null() command is provided for extensibility. It can be used as a means of providing a heartbeat message to downstream splicing equipment.

The splice_schedule() command is a command that allows a schedule of splice events to be conveyed in advance.

The splice_insert() command shall be sent at least once before each splice point. Packets containing the entirety of the splice_info_table shall always precede the packet that contains the related splice point (i.e., the first packet that contains the first byte of an access unit whose presentation time most closely matches the signalled time in the splice_info_section).

In order to give advance warning of the impending splice (a pre-roll function), the splice_insert() command could be sent multiple times before the splice point. For example, the splice_insert() command could be sent at 8, 5, 4 and 2 seconds prior to the packet containing the related splice point. In order to meet other splicing deadlines in the system, any message received with less than 4 seconds of advance notice may not create the desired result. The splice_insert() message shall be sent at least once a minimum of 4 seconds in advance of the desired splice time for a network out point condition. It is recommended that, if a return-to-network (an in point) message is sent, the same minimum 4 second pre-roll be provided.

The splice_insert() command provides for an optional break_duration() structure to identify the length of the commercial break. It is recommended that splice_insert() messages with the out_of_network_indicator set to 1 (a network out point) include a break_duration() structure to provide the splicer with an indication of when the network in point will occur. The break_duration() structure provides for an optional auto_return flag that, when set to 1, indicates that the splicer is to return to the network at the end of the break (defined as Auto Return Mode, refer to clause 9.4.2.2). It is recommended that this Auto Return Mode be used to support dynamic avail durations.