

SCTE | **STANDARDS**

Digital Video Subcommittee

SCTE STANDARD

SCTE 35 2022a

Digital Program Insertion Cueing Message

NOTICE

The Society of Cable Telecommunications Engineers (SCTE) Standards and Operational Practices (hereafter called “documents”) are intended to serve the public interest by providing specifications, test methods and procedures that promote uniformity of product, interoperability, interchangeability, best practices, and the long term reliability of broadband communications facilities. These documents shall not in any way preclude any member or non-member of SCTE from manufacturing or selling products not conforming to such documents, nor shall the existence of such standards preclude their voluntary use by those other than SCTE members. SCTE assumes no obligations or liability whatsoever to any party who may adopt the documents. Such adopting party assumes all risks associated with adoption of these documents and accepts full responsibility for any damage and/or claims arising from the adoption of such documents. NOTE: The user’s attention is called to the possibility that compliance with this document may require the use of an invention covered by patent rights. By publication of this document, no position is taken with respect to the validity of any such claim(s) or of any patent rights in connection therewith. If a patent holder has filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license, then details may be obtained from the standards developer. SCTE shall not be responsible for identifying patents for which a license may be required or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Patent holders who believe that they hold patents which are essential to the implementation of this document have been requested to provide information about those patents and any related licensing terms and conditions. Any such declarations made before or after publication of this document are available on the SCTE web site at <https://scte.org>.

All Rights Reserved
©2022 Society of Cable Telecommunications Engineers, Inc.
140 Philips Road
Exton, PA 19341

Document Types and Tags

Document Type: Specification

Document Tags:

- | | | |
|---|---|--|
| ☐ Test or Measurement | ☐ Checklist | ☐ Facility |
| ☒ Architecture or Framework | ☐ Metric | ☒ Access Network |
| ☐ Procedure, Process or Method | ☒ Cloud | ☐ Customer Premises |

Document Release History

Release	Date
SCTE 35 2001	12/14/2001
SCTE 35 2004	5/7/2004
SCTE 35 2007	11/30/2007
SCTE 35 2011	12/5/2011
SCTE 35 2012	8/22/2012
SCTE 35 2013	4/8/2013
SCTE 35 2013a	12/2/2013
SCTE 35 2015	1/5/2015
SCTE 35 2016	3/14/2016
SCTE 35 2017	5/30/2017
SCTE 35 2019	2/11/2019
SCTE 35 2019a	12/16/2019
SCTE 35 2020	12/30/2020
SCTE 35 2022	March 2022

Note: Standards that are released multiple times in the same year use: a, b, c, etc. to indicate normative balloted updates and/or r1, r2, r3, etc. to indicate editorial changes to a released document after the year.

Table of Contents

Title	Page Number
NOTICE	2
Document Types and Tags	3
Document Release History	3
Table of Contents	4
1. Introduction	9
1.1. Executive Summary	9
1.2. Scope	9
1.3. Benefits	9
1.4. Intended Audience	10
1.5. Areas for Further Investigation or to be Added in Future Versions	10
2. Normative References	10
2.1. SCTE References	10
2.2. Standards from Other Organizations	10
2.3. Published Materials	11
3. Informative References	11
3.1. SCTE References	11
3.2. Standards from Other Organizations	11
3.3. Published Materials	12
4. Compliance Notation	13
5. Abbreviations and Definitions	13
5.1. Abbreviations	13
5.2. Definitions	15
6. Introduction	17
6.1. Splice points (Informative)	17
6.2. Program splice points (Informative)	18
6.3. Splice events (Informative)	19
6.4. Content storage considerations (Informative)	19
6.5. PID selection	19
6.5.1. PID selection (Normative)	19
6.5.2. PID selection (Informative)	20
6.6. Message flow (Informative)	20
6.7. Usage Examples (Informative)	22
6.7.1. SCTE 35 Ad Break Example	22
6.7.2. SCTE 35 Program Example	23
6.7.3. SCTE 35 Complex example	24
7. Notational Conventions	26
7.1. Normative XML schema	26
7.1.1. XML namespace and version	26
7.2. Unknown/Unrecognized/Unsupported XML elements and attributes	26
7.3. Element order	26
7.4. Binary representation in XML	27
8. PMT Descriptors	27
8.1. Registration descriptor	27
8.1.1. Semantic definition of fields in Registration Descriptor	28
8.2. Cue Identifier Descriptor	28
8.2.1. Semantic definition of fields in Cue Identifier Descriptor	28
8.2.2. Description of cue_stream_type usage	29
8.3. Stream Identifier Descriptor	29
8.3.1. Semantic definition of fields in Stream Identifier Descriptor	30
9. Splice information table	30

9.1.	Overview	30
9.2.	Legacy Command Descriptions	31
9.3.	Time Signal Command	31
9.4.	Command Changes	31
9.5.	Time base discontinuities	32
9.6.	Splice info section	32
9.6.1.	Semantic definition of fields in splice_info_section()	35
9.7.	Splice commands	38
9.7.1.	splice_null()	38
9.7.2.	splice_schedule()	38
9.7.2.1.	Semantic definition of fields in splice_schedule()	40
9.7.3.	splice_insert()	42
9.7.3.1.	Semantic definition of fields in splice_insert()	44
9.7.4.	time_signal()	46
9.7.4.1.	Semantic definition of time_signal()	47
9.7.5.	bandwidth_reservation()	47
9.7.6.	private_command()	48
9.8.	Time	49
9.8.1.	splice_time()	49
9.8.1.1.	Semantic definition of fields in splice_time()	49
9.8.2.	break_duration()	50
9.8.2.1.	Semantic definition of fields in break_duration()	50
9.9.	Constraints	51
9.9.1.	Constraints on splice_info_section()	51
9.9.2.	Constraints on the interpretation of time	51
9.9.2.1.	Constraints on splice_time() for splice_insert()	51
9.9.2.2.	Constraints on break_duration() for splice_insert()	52
10.	Splice Descriptors	53
10.1.	Overview	53
10.2.	Splice descriptor	53
10.2.1.	Semantic definition of fields in splice_descriptor()	54
10.2.2.	Private Splice Descriptors	55
10.3.	Specific splice descriptors	56
10.3.1.	avail_descriptor()	56
10.3.1.1.	Semantic definition of fields in avail_descriptor()	57
10.3.2.	DTMF_descriptor()	57
10.3.2.1.	Semantic definition of fields in DTMF_descriptor()	58
10.3.3.	segmentation_descriptor()	58
10.3.3.1.	Segmentation descriptor details	61
10.3.3.2.	CableLabs metadata identifier	69
10.3.3.3.	MPU() definition and semantics	69
10.3.3.4.	MID() definition and semantics	69
10.3.3.5.	ADS format and definition	70
10.3.3.6.	SCR description and format	70
10.3.3.7.	Segmenting Content - Additional semantics	71
10.3.3.8.	Programs - Additional semantics	72
10.3.3.9.	Chapters - Additional semantics	73

10.3.3.10.Break – Additional semantics	74
10.3.3.11.Delivery Restrictions – Additional semantics	74
10.3.3.12.Content Identifiers – Additional semantics	75
10.3.3.13.Placement Opportunities – Additional semantics	75
10.3.3.14.Alternate Content Opportunities – Additional semantics	80
10.3.4. time_descriptor()	81
10.3.4.1. Informative description of TAI	81
10.3.4.2. Semantic definition of fields in time_descriptor()	82
10.3.4.1. Synchronized Client Clock	83
10.3.4.2. Synchronized Clock carriage in HLS Timed Metadata (ID3 tags)	84
10.3.5. audio_descriptor()	84
10.3.5.1. Semantic definition of fields in audio_descriptor()	86
11. Encryption	87
11.1. Overview	87
11.2. Fixed key encryption	88
11.3. Encryption algorithms	88
11.3.1. DES – ECB mode	88
11.3.2. DES – CBC mode	88
11.3.3. Triple DES EDE3 – ECB mode	89
11.3.4. User private algorithms	89
12. SCTE 35 Usage	89
12.1. SCTE 35 Usage in DASH	89
12.2. SCTE 35 Usage in HLS	89
12.2.1. SCTE 35 markup in HLS using EXT-X-DATERANGE	90
12.2.2. SCTE 35 markup in HLS using EXT-X-SCTE35	90
12.2.3. HLS cue tags	90
12.2.4. HLS playlist example	92
13. SCTE 35 XML elements and types	93
13.1. Ext element	93
13.2. PTSType	94
13.3. Segmentation Upid Element	94
14. Sample SCTE 35 Messages (Informative)	95
14.1. time_signal – Placement Opportunity Start	95
14.2. splice_insert	96
14.3. time_signal – Placement Opportunity End	97
14.4. time_signal – Program Start/End	97
14.5. time_signal – Program Overlap Start	98
14.6. time_signal – Program Blackout Override / Program End	99
14.7. time_signal – Program End	99
14.8. time_signal – Program Start/End - Placement Opportunity End	100

List of Figures

Title	Page Number
Figure 1 - SCTE 35 Ad Break Example	22
Figure 2 - SCTE 35 Program Example	23
Figure 3 - SCTE 35 Program Example	25
Figure 4 - SignalGroup	27

Figure 5 - SpliceInfoSection	35
Figure 6 - SpliceNull	38
Figure 7 - SpliceSchedule	40
Figure 8 - SpliceInsert	44
Figure 9 - TimeSignal	47
Figure 10 – BandwidthReservation	47
Figure 11 - PrivateCommand	48
Figure 12 - SpliceTime	49
Figure 13 - BreakDuration	50
Figure 14 - SpliceDescriptorType	54
Figure 15 - PrivateDescriptorType	55
Figure 16 - AvailDescriptor	56
Figure 17 - DTMFDescriptor	58
Figure 18 - SegmentationDescriptorType	60
Figure 19 - TimeDescriptor	82
Figure 20 – Time Relationships	83
Figure 21 – AudioDescriptor	86
Figure 22 - Ext Element	94
Figure 23 - SegmentationUpid	95

List of Tables

Title	Page Number
Table 1 - registration_descriptor()	28
Table 2 - cue_identifier_descriptor()	28
Table 3 - cue_stream_type values	29
Table 4 - stream_identifier_descriptor()	30
Table 5 - splice_info_section()	32
Table 6 - sap_type values	35
Table 7 - splice_command_type values	37
Table 8 - splice_null()	38
Table 9 - splice_schedule()	39
Table 10 - splice_insert()	43
Table 11 - time_signal()	46
Table 12 - bandwidth_reservation()	47
Table 13 - private_command()	48
Table 14 - splice_time()	49

SCTE 35 2022a

Table 15 - break_duration()	50
Table 16 - Splice Descriptor Tags	53
Table 17 - splice_descriptor()	54
Table 18 - avail_descriptor()	56
Table 19 - DTMF_descriptor()	57
Table 20 - segmentation_descriptor()	59
Table 21 - device_restrictions	62
Table 22 - segmentation_upid_type	63
Table 23 - segmentation_type_id	65
Table 24 – MPU()	69
Table 25 – MID()	69
Table 26 – Placement Opportunity Start Numbering Example	78
Table 27 - time_descriptor()	82
Table 28 - audio_descriptor()	85
Table 29 - Encryption algorithm	88
Table 30 - Tag #EXT-X-SCTE35	90
Table 31 - Tag attributes	91

1. Introduction

1.1. Executive Summary

This standard, “Digital Program Insertion Cueing Message for Cable” (SCTE 35), is the core signaling standard for advertising, Program and distribution control (e.g., blackouts) of content for content providers and content distributors. SCTE 35 is being applied to QAM/IP, Title VI/TVE (TV Everywhere), and live/time shifted (DVR, VOD, etc.) delivery. SCTE 35 signals can be used to identify advertising Breaks, advertising content, and programming content (e.g., specific Programs and Chapters within a Program).

SCTE 35 complements other Standards to complete the eco-systems. [SCTE 30] is used to support splicing of advertising into live QAM MPEG-2 transport streams. [SCTE 130-3] is used to support alternate content decisions (advertising, blackouts, stream switching) for live and time shifted delivery. [SCTE 214-1] defines how SCTE 35 is carried in MPEG-DASH. [SCTE 224] (ESNI) is used to pass event and policy information from provider or other systems to communicate distribution control instructions.

The recommended practices for SCTE 35 are contained in [SCTE 67] “Recommended Practice for Digital Program Insertion for Cable”.

1.2. Scope

This standard supports delivery of events, frame accurate or non-frame accurate, and associated descriptive data in MPEG-2 transport streams, MPEG-DASH and HLS. This standard supports the splicing of content (MPEG-2 transport streams, MPEG-DASH, etc.) for the purpose of Digital Program Insertion, which includes Advertisement insertion and insertion of other content types. This standard defines an in-stream messaging mechanism to signal splicing and insertion opportunities. As such, this standard does not specify the insertion method used or constraints applied to the content being inserted, nor does it address constraints placed on insertion devices. Fully compliant MPEG-2 transport stream (either Multi Program Transport Stream or Single Program Transport Stream), MPEG-DASH content, etc. is assumed. No further constraints beyond the inclusion of the defined cueing messages are placed upon the stream.

This standard specifies a technique for carrying notification of upcoming 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. For MPEG-2 transport streams, the splice information table, which pertains to a given program, is carried in one or more MPEG Sections carried in 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. For MPEG-DASH, the splice information table is carried in the DASH MPD (See [SCTE 214-1]) or in media segments (see [SCTE 214-2] and [SCTE 214-3]). Section 12.2 details how SCTE 35 messages are carried in HLS manifests.

1.3. Benefits

SCTE 35 is a key part of the eco-system to enable advertising and content distribution business. A common/well-formed signaling model enables downstream systems to be implemented in a cost effective, consistent and non-ambiguous fashion to achieve business objectives.

1.4. Intended Audience

The intended audience is Content Providers, Multi-Channel Video Program Distributors, TV Everywhere Providers/Distributors and vendors/developers who build solutions.

1.5. Areas for Further Investigation or to be Added in Future Versions

Follow-up work related to UHD and HDR *may* be addressed. The use of studio time is currently an active topic in the DASH-IF group and a future version of this standard *may* want to reference that work. Sections 10.3.3.5 to 10.3.3.12 *should* be reviewed.

2. Normative References

The following documents contain provisions, which, through reference in this text, constitute provisions of this document. At the time of Subcommittee approval, the editions indicated were valid. All documents are subject to revision; and while parties to any agreement based on this document are encouraged to investigate the possibility of applying the most recent editions of the documents listed below, they are reminded that newer editions of those documents might not be compatible with the referenced version.

2.1. SCTE References

- No normative references are applicable.

2.2. Standards from Other Organizations

[ATSC A/57B]	ATSC A/57B – ATSC Standard: Content Identification and Labeling for ATSC Transport Document A/57B, 26 May 2008
[ATSC A/52]	A/52:2018: Digital Audio Compression (AC-3) (E-AC-3) Standard
[CLADI1-1]	MD-SP-VOD-CONTENTv1.1 – C01-120803 – CableLabs Video-on-Demand Content Specification 1.1
[EIDR ID FORMAT]	EIDR ID Format – “EIDR: ID FORMAT Ver. 1.5.1 5 Oct 2017
[FIPS PUB 46-3]	FIPS PUB 46-3, 1999 October 25 – Data Encryption Standard
[FIPS PUB 81]	FIPS PUB 81, 1980 December 2 – DES Modes of Operation
[HLS]	RFC 8216 – “HTTP Live Streaming”
[ISO/BMFF]	ISO/IEC 14496-12:2020 Information technology - Coding of audio-visual objects - Part 12: ISO base media file format
[ISO 15706-2]	ISO 15706-2:2007 – Information and Documentation – International Standard Audiovisual Number (V-ISAN) – Part 2: Version Identifier
[ISO 639-2]	Codes for the Representation of Names of Languages
[MPEG Systems]	ITU-T Recommendation H.222.0 / ISO/IEC 13818-1 (2019) – Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Systems
[RFC 3986]	RFC 3986 – “Uniform Resource Identifier (URI): Generic Syntax”, January 2005
[RFC 4122]	RFC 4122 – “A Universally Unique Identifier (UUID) URN Namespace”, July 2005
[RFC 4648]	RFC 4648 – “The Base16, Base32, and Base64 Data Encodings”
[RFC 5646]	RFC 5646 – “Tags for Identifying Languages”
[SMPTE 330]	SMPTE ST 330:2011 Unique Material Identifier (UMID)

[XML]	W3C Recommendation, “Extensible Markup Language (XML) 1.0 (Fifth Edition)”
[XML Namespaces]	W3C Recommendation, “Namespaces In XML (Third Edition)”
[XML InfoSet]	W3C Recommendation, “XML InfoSet (Second Edition)”, John Cowan
[XML SchemaP1]	W3C Recommendation, “XML Schema Part 1: Structures (Second Edition)”, H. Thompson, et al, 28 October 2004
[XML SchemaP2]	W3C Recommendation, “XML Schema Part 2: Datatypes (Second Edition)”, P. Biron, et al, 28 October 2004

2.3. Published Materials

- No normative references are applicable.

3. Informative References

The following documents might provide valuable information to the reader but are not required when complying with this document.

3.1. SCTE References

[SCTE 30]	SCTE 30– Digital Program Insertion Splicing API
[SCTE 67]	SCTE 67– Recommended Practice for Digital Program Insertion for Cable
[SCTE 104]	SCTE 104– Automation System to Compression System Communications Applications Program Interface (API)
[SCTE 118-2]	SCTE 118-2– Program-Specific Ad Insertion – Content Provider to Traffic Communication Applications Data Model
[SCTE 130-3]	SCTE 130-3– Digital Program Insertion-Advertising Systems Interfaces – Part 3 – Ad Management Service (ADM) Interface
[SCTE 172]	SCTE 172– Constraints on AVC and HEVC Structured Video Coding for Digital Program Insertion
[SCTE 214-1]	SCTE 214-1– MPEG DASH for IP-Based Cable Services Part 1: MPD Constraints and Extensions
[SCTE 214-2]	SCTE 214-2– MPEG DASH for IP-Based Cable Services Part 2: DASH/TS Profile
[SCTE 214-3]	SCTE 214-3 MPEG DASH for IP-Based Cable Services Part 3: DASH/FF Profile
[SCTE 223]	SCTE 223 – Adaptive Transport Stream
[SCTE 224]	SCTE 224 – Event Scheduling and Notification Interface
[SCTE 236]	SCTE 236 – Content Metadata ¹
[SCTE 248]	Operational Practice on Multiple Audio Signaling
[SCTE 250]	Event and Signaling Management API (ESAM)

3.2. Standards from Other Organizations

[Ad-ID]	Advertising Digital Identification, LLC
---------	---

¹ Formerly CableLabs Content 3.0

[DASH]	ISO/IEC 23009-1 Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats
[DOI]	Digital Object Identifier
[EIDR]	Entertainment ID Registry Association (EIDR)
[ETSI 103 752-1]	ETSI TS 103 752-1 – Dynamic substitution of content in linear broadcast; Part 1: Carriage and signaling of placement opportunity information in DVB Transport Streams
[HLS-TMD]	Timed Metadata for HTTP Live Streaming
[ID3v2-struct]	ID3 Tag version 2.4.0 – Main Structure
[ISAN]	ISAN (International Standard Audiovisual Number) website
[ISO 13818-4]	ISO/IEC 13818-4 – Information Technology – Generic coding of moving pictures and associated audio information – Part 4: Conformance testing
[ISO 15706-1]	ISO 15706-1 – Information and Documentation – International Standard Audiovisual Number (ISAN)
[ISO 15706-1 Amd 1]	ISO 15706-1 Amd 1 – Alternate Encodings and editorial changes
[ITU H.262]	ITU-T Recommendation H.262 / ISO/IEC 13818-2 (2000), Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Video
[PTP]	IEEE 1588-2008, IEEE, 24 July 2008, doi:10.1109/IEEESTD.2008.4579760
[RFC5905]	“Network Time Protocol Version 4: Protocol and Algorithms Specification”
[SMPTE 312]	SMPTE ST 312:2001 – SMPTE STANDARD for Television – Splice Points for MPEG-2 Transport Streams
[SMPTE 2092-1]	SMPTE RP 2092-1:2015 – SMPTE Recommended Practice, Advertising Digital Identifier (Ad-ID(R)) Representations
[SMPTE 2079]	SMPTE RP 2079:2013 – Digital Object Identifier (DOI) Name and Entertainment ID Registry (EIDR) Identifier Representations
[SMPTE RA]	SMPTE Registration Authority, LLC

3.3. Published Materials

[ETDS]	Media Perspectives Event Triggering Distribution Specification (ETDS) and Event Triggering Distribution Specification Supplement (ETDSS)
--------	--

4. Compliance Notation

<i>shall</i>	This word or the adjective “ <i>required</i> ” means that the item is an absolute requirement of this document.
<i>shall not</i>	This phrase means that the item is an absolute prohibition of this document.
<i>forbidden</i>	This word means the value specified <i>shall</i> never be used.
<i>should</i>	This word or the adjective “ <i>recommended</i> ” means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighted before choosing a different course.
<i>should not</i>	This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
<i>may</i>	This word or the adjective “ <i>optional</i> ” means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.
<i>deprecated</i>	Use is permissible for legacy purposes only. Deprecated features may be removed from future versions of this document. Implementations should avoid use of deprecated features.

5. Abbreviations and Definitions

5.1. Abbreviations

AC-3	Audio Codec 3 (also known as Dolby Digital)
ADI	Asset Distribution Interface
ADM	Ad Management Service
Ad-ID	Advertisement Identifier
API	Application Programming Interface
ATSC	Advanced Television Systems Committee.
AVC	Advanced Video Coding
bslbf	Bit string, left bit first, where left is the order in which bit strings are written.
CBC	Cipher Block Chaining
CRC	Cyclic Redundancy Check
DASH	Dynamic Adaptive Streaming over HTTP
DASH-IF	DASH Industry Forum
DES	Data Encryption Standard
DOI	Digital Object Identifier
DTMF	Dual Tone Multi Frequency
DTS	Decoding Time Stamp
DVB	Digital Video Broadcasting
E-AC-3	Enhanced Audio Codec 3 (also known as Dolby Digital Plus)
ECB	Electronic CodeBook
EIDR	Entertainment Identifier Registry

ESAM	Event Signaling and Management API
FIPS	Federal Information Processing Standard
GPS	Global Positioning System
HEVC	High Efficiency Video Coding
HLS	HTTP Live Streaming
HTTP	HyperText Transfer Protocol
IEC	International Electro-technical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISAN	International Standard Audiovisual Number (see [ISO 15706-1] and [ISO 15706-1 Amd 1])
ISCI	Industry Standard Commercial Identifier
ISO	International Organization for Standardization
IP	Internet Protocol
ITU-T	International Telecommunication Union Telecommunications sector
MID	Multiple UPID
MPEG	Moving Picture Experts Group
MPD	Media Presentation Description
MPTS	a Multi Program Transport Stream.
MPU	Managed Private UPID
MVPD	Multi-Channel Video Program Distributors; also cited as “content distributor”, “distributor”, or affiliate.
NTP	Network Time Protocol as defined in IETF [RFC5905].
PCR	Program Clock Reference
PID	Packet Identifier; a unique 13-bit value used to identify the type of data stored in the packet payload (see [MPEG Systems]).
PMT	Program Map Table (see [MPEG Systems]).
PTP	Precision Time Protocol as defined in IEEE Std 1588-2008 PTP.
PTS	Presentation Time Stamp (see [MPEG Systems]).
Rpchof	Remainder polynomial coefficients, highest order first.
RFC	Request for Comments
SAP	Stream Access Point (see [ISOBMFF])
SCTE	Society of Cable Telecommunications Engineers
SMPTE	Society of Motion Picture and Television Engineers
SPTS	a Single Program Transport Stream.
STC	System Time Clock
TAI	International Atomic Time (TAI, from the French name Temps Atomique International) This time reference scale was established by the BIPM (Bureau International des Poids et Mesures) on the basis of atomic clock readings from various laboratories around the world.
TI	Turner Identifier
TID	Tribune Identifier
TS	Transport Stream
uimsbf	Unsigned integer, most significant bit first.

UMID	Unique Material Identifier
UPID	Unique Program Identifier
URI	Uniform Resource Identifier – See [RFC 3986].
UTC	Coordinated Universal Time. This is the primary time standard by which the world regulates clocks and time.
UUID	Universally Unique Identifier
V-ISAN	Version-ISAN (core ISAN number plus a version number) – See [ISO 15706-2]).
XML	Extensible Markup Language – See [W3C Recommendation, “Extensible Markup Language (XML) 1.0]
W3C	World Wide Web Consortium

5.2. Definitions

Access Unit	A coded representation of a presentation unit (see [ITU H.262]).
Alternate Content Opportunity	A potentially constrained location relative to digital content where Program based content alterations can occur. The alterations may include insertions, replacements, or deletions of Program based content in whole or in part (e.g., one or more Chapters with or without Breaks).
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).
Ad Block	A potentially constrained location relative to digital content where Ad based content alterations can occur within a Placement Opportunity. The location is constrained as a particular selection of contiguously scheduled Ads so that the sum of their durations is available for insertion or modification rather than each Ad’s individual duration.
Analog Cue Tone	In an analog system, a signal that is usually either a sequence of DTMF tones or a contact closure that denotes to ad insertion equipment that an Advertisement avail is about to begin or end.
Avail	Time space provided to cable operators by cable programming services during a Program for use by the MVPD operator; the time is usually sold to local advertisers or used for channel self-promotion.
Barter Ad	Advertisement media incorporated into program content that shall be distributed according to the carriage and or licensing agreement for that program.
Bit Stream Format	An encoding of information resulting in a compliant MPEG-2 transport stream. See [MPEG Systems].
Break	Avail or actual insertion in progress. In a sales context, a Break is divided into sellable units (Avails). In an insertion context, a Break is divisible into individual insertion events (slots). Also, a SCTE 35 Segmentation Type, “Break Start/End” that is typically used to mark all non-program sections. These would include promotional material and advertisements including barter ads.
Chapter	A short section of a longer Program, usually situated to permit a viewer to easily locate a scene or section of the Program.
Component Splice Mode	A mode of the cueing message whereby the program_splice_flag is set to ‘0’, indicating 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.
Content	Generic term for television material, either Advertisements or Programs.
Content Distributor	Typically an MVPD; also cited as “distributor”.
Content Provider	Typically a network programmer; also cited as “programmer” or “provider”.
Cueing Message	See message.
Deprecated	Use is permissible for legacy purposes only. <i>Deprecated</i> features <i>may</i> be removed from future versions of the standard. Implementations <i>should</i> avoid use of <i>deprecated</i> features.
Event	A splice event or a viewing event.
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.
In Stream Device	A device that receives the transport stream directly and is able to derive timing information directly from the transport stream.
Message	In the context of this document a message is the content of any splice info section.
Multi Program Transport Stream	A transport stream with multiple programs.
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.
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.
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 (see [MPEG Systems]).
Placement Opportunity	A potentially constrained location relative to digital content where ad insertion or content alterations can occur. The alterations may include insertions, replacements, or deletions of content in whole or in part. These locations, which contain the opportunity for content insertion, have traditionally been referred to as Avails [SCTE35] for linear video content; however, Placement Opportunity refers to address and time locations where content may be placed, regardless of platform (i.e. Video in VOD, Banner Images on Menus and ITV channels, etc.).
PID stream	All the packets with the same PID within a transport stream.
pointer_field	The first byte of a transport packet payload, required when a section begins in that packet (see [MPEG Systems]).
Presentation Time	The time that a presentation unit is presented in the system target decoder (see [MPEG Systems]).
Presentation Unit	A decoded Audio Access Unit or a decoded picture (see [ITU H.262]).
Program	A collection of video, audio, and data PID streams that share a common program number within an MPTS (see [MPEG Systems]). 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).
Program In Point	A group of PID stream In Points that correspond in presentation time.

Program Out Point	A group of PID stream Out Points that correspond in presentation time.
Program Splice Mode	A mode of the cueing message whereby the program_splice_flag is set to '1', indicating that the message refers to a Program Splice Point and that all PIDs/components of the program are to be spliced.
Program Splice Point	A Program In Point or a Program Out Point.
Promo	An individual, self-contained audio-visual non-merchandisable or promotional item. If a content provider does not support dedicated identification of Promos, these Segments can be signaled as Advertisements
Receiving Device	A device that receives or interprets sections conforming to this standard. Examples of these devices include splicers, ad servers, segmenters and satellite receivers.
Registration Descriptor	Carried in the PMT of a program to indicate that, when signaling 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 standard.
reserved	The term "reserved", when used in the clauses defining the coded bit stream, indicates that the value <i>may</i> be used in the future for extensions to the standard. Unless otherwise specified, all reserved bits shall be set to '1' and this field shall be ignored by receiving equipment.
Segment	Either a <i>Program</i> , a <i>Chapter</i> , a <i>Provider Advertisement</i> , a <i>Distributor Advertisement</i> , a <i>Provider Promo</i> , a <i>Distributor Promo</i> or an <i>Unscheduled Event</i> as listed in Table 23, segmentation_type_id.
Single Program Transport Stream	A transport stream containing a single MPEG program.
Splice Event	An opportunity to splice one or more PID streams.
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.
Splice Point	A point in a PID stream that is either an Out Point or an In Point.
UPID	Identifier for the content or content Segment that programmers' systems utilize.
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.

6. Introduction

6.1. Splice points (Informative)

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 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.

The use of “advanced” video compression, such as documented by ITU-T H.264/ ISO/IEC 14496-10 (“AVC”) or ITU-T H.265/ ISO/IEC 23008-2 (“HEVC”) typically requires advance notice of Splice Points to permit the video encoder to create a spliceable point in the video elementary stream. In most systems this is provided by the use of [SCTE 104] to provide a communications path from the programmer’s Automation System to their Compression System. These AVC and HEVC coding constraints for splice points are documented in [SCTE 172]. 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 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 behavior of splicing devices will not be specified or constrained in any way by this standard.

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 standard defines two ways of signaling 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 standard). 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 standard 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 Section 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 standard provides a method for in-band signaling of splice events using splice commands to downstream splicing equipment. Signaling 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 signaled event (a signaled event *should* be interpreted as an opportunity to splice; not a command). A splice information table carries the notice of splice event opportunities. Each signaled splice event is analogous to an analog 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 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 behavior 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 standard.

This standard provides optional tools to assist with segmenting content into shorter sections which *may* be either Chapters or Advertisements (see Section 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 CA system. Each cue message PID *may* include the cue_identifier_descriptor defined in Section 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 **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 equipment *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 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 behavior 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 standard, 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 document 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 [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 re-stamps 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 standard. 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 an 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 behavior chosen by the implementer.

6.7. Usage Examples (Informative)

The following sections are examples on the usage of SCTE 35 for marking various Program and Advertisement Segments. The various SCTE 35 Cueing Messages are typically placed in affiliate programming tiers by the programmer, so an affiliate will only see the cues for which they are authorized. See Section 14 for sample SCTE 35 Messages.

6.7.1. SCTE 35 Ad Break Example

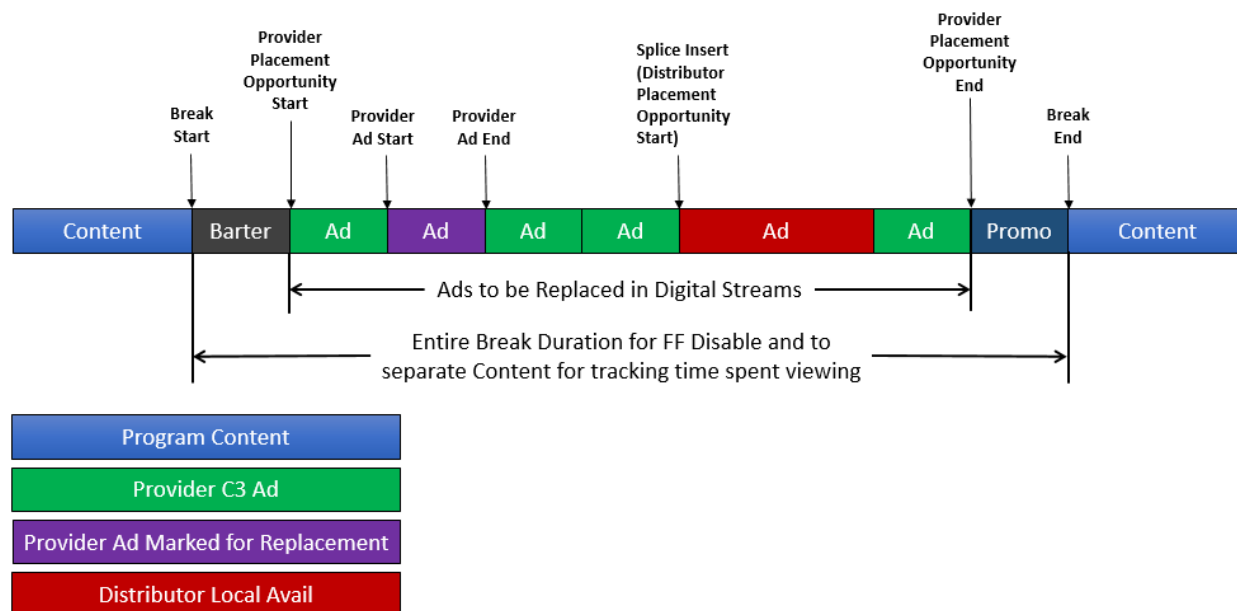


Figure 1 - SCTE 35 Ad Break Example

When a Content Provider inserts non-program content into a Program, it typically falls into one of several categories.

- 1) Promotional Items – These are clips that usually promote other programming by the programmer or may be public service announcements. In children’s programming keeping these intact may be a regulatory matter and they are typically not replaced.
- 2) Barter ads – Some programmers purchase content from studios and the studio reserves the right to insert some advertising. As such, these ads are not usually replaceable.
- 3) Provider C3 ad – These are programmer inserted Advertisements. They are meant to be displayed to a measured audience and may be replaced on devices that are not measured in the traditional sense. Some programmers may use out of band methods such as SCTE 224 to allow these ads to be replaced after a certain amount of time, typically after the C3 windows expires.
- 4) Provider ad marked for replacement – One version of addressable advertising sells a particular ad slot to a single advertiser and plays an ad addressed to the most likely viewing demographic audience (network, show, time of day, etc.) in the original programmer feed. It also marks the individual ad for possible addressable replacement if the device is capable and if the audience information to do so is available. One example would be a car company with a minivan ad for a typical family show, but able to switch to a sports car or truck ad if it is known that the targeted demographic is not watching.

- 5) Distributor Local Avail – This is a legacy distributor Break. It should be signaled with a Distributor Placement Opportunity instead of a splice_insert, although currently deployed systems may not work.

Explanation of the SCTE 35 signaling used:

- 1) Break Start/End – These are inserted to mark the edges of non-program material. This can be used, for example, to disable fast forward during the Break to ensure barter ads and others are seen and measured. It could also be used to remove the entire Break to repurpose the content for future use.
- 2) Provider Placement Opportunity Start/End – These signals mark an entire block of ads that *may* be replaced. The decision to actually replace the entire Break is based on many factors including live vs restart vs lookback viewing, device, demographics, etc. If the distributor local avail is embedded inside this Break, one should use the MID structure to include information about pod position and length of the local avail; in this way, one will be able to stitch together all of the programmer and distributor ads and any required slate for a seamless user presentation.
- 3) Provider Ad Start/End – If a programmer wants to have a downstream system replace an individual ad, they can use this to mark that ad. They would typically use an Ad-ID identifier for the segmentation UPID.
- 4) splice_insert – This is a legacy construct indicating where the local affiliate Breaks occur.

6.7.2. SCTE 35 Program Example

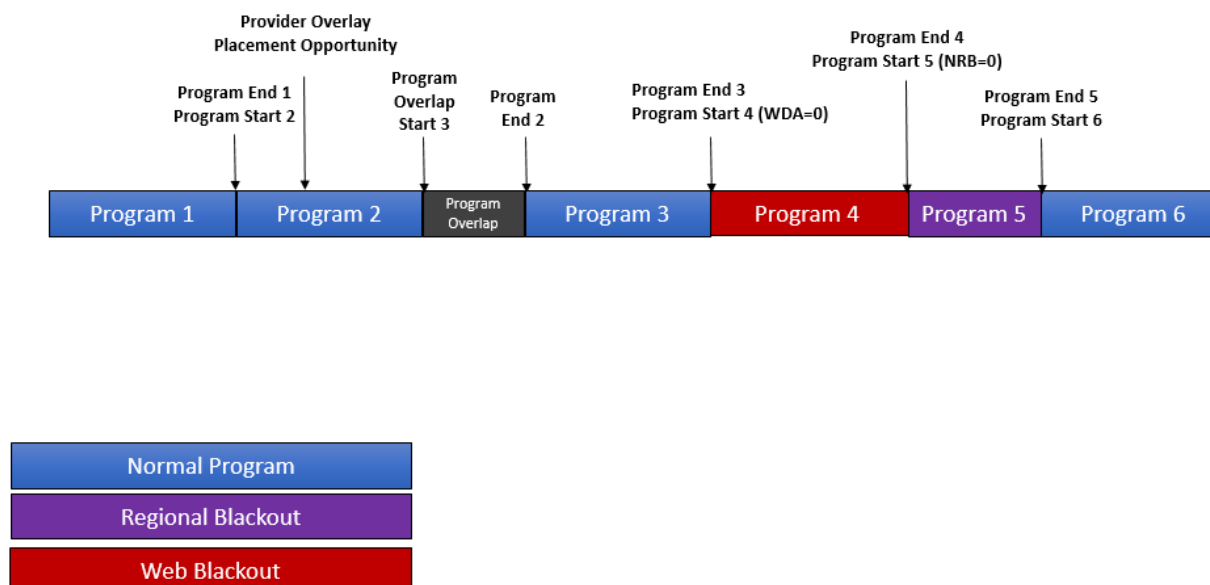


Figure 2 - SCTE 35 Program Example

Ad Breaks described in the previous section are typically embedded at various points within the Program content; they are not shown in this diagram.

This example shows the segmentation types used to delineate common programming sequences. The programmer should set the segmentation Unique Program Identifier (UPID) to a value that can be cross referenced to the SCTE 224 feed in order to deliver additional information about the Program.

WDA is the Web Delivery Allowed flag [see Section 10.3.3]. Setting this to false (0) indicates that there is a web blackout. An implementer would need to verify exactly what the programmer means by this, but it is typically used to indicate anything delivered to a subscriber over the open internet as opposed to a closed MVPD network.

NRB is the No Regional Blackout flag [see Section 10.3.3]. When set to false (0), this indicates that this Program is subject to a regional blackout. An implementer would, likely, refer to a SCTE 224 feed or a programmer or sports league website to determine the appropriate blackout geography.

6.7.3. SCTE 35 Complex example

This is an additional example showing various uses of the Provider and Distributor Placement Opportunity in more complex overlapping scenarios as well as application of the Alternate Content Opportunity.

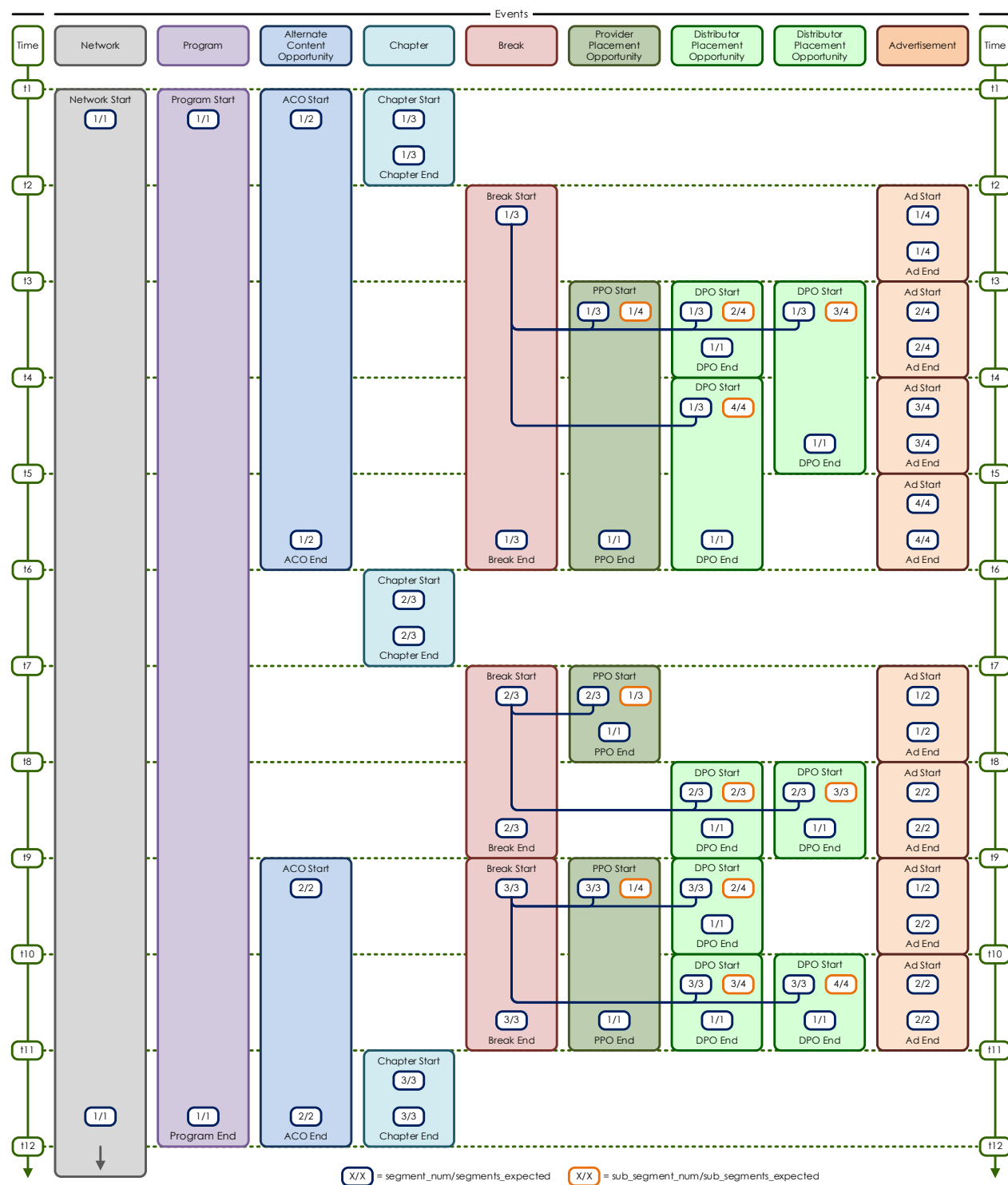


Figure 3 - SCTE 35 Program Example

7. 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 specification **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 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 document and the normative schema document (i.e., the separately provided XML ‘xsd’ file) conflict, this document **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.

7.1.1. XML namespace and version

The following XML namespaces and versions are defined in the normative XML schema for this specification:

Prefix	Namespace	Version
scte35	http://www.scte.org/schemas/35	20211104

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 **shall** 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 [XML InfoSet] 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).

7.4. Binary representation in XML

The content of a SpliceInfoSection (see Section 9.6) *may* be represented in XML as either a fully parsed structure or as a binary representation. This choice is provided in the XML Schema as a Signal group (Figure 4). The binary representation *shall* be Base64 encoded per [RFC 4648]. The Binary element *should* be assumed to contain a SpliceInfoSection unless explicitly indicated by the signalType attribute as something else. The signalType attribute is provided to allow some flexibility in how the XML Schema is used for binary representations. Whenever the value of signalType is not “SpliceInfoSection” it *shall* be prefixed with the text “private:”

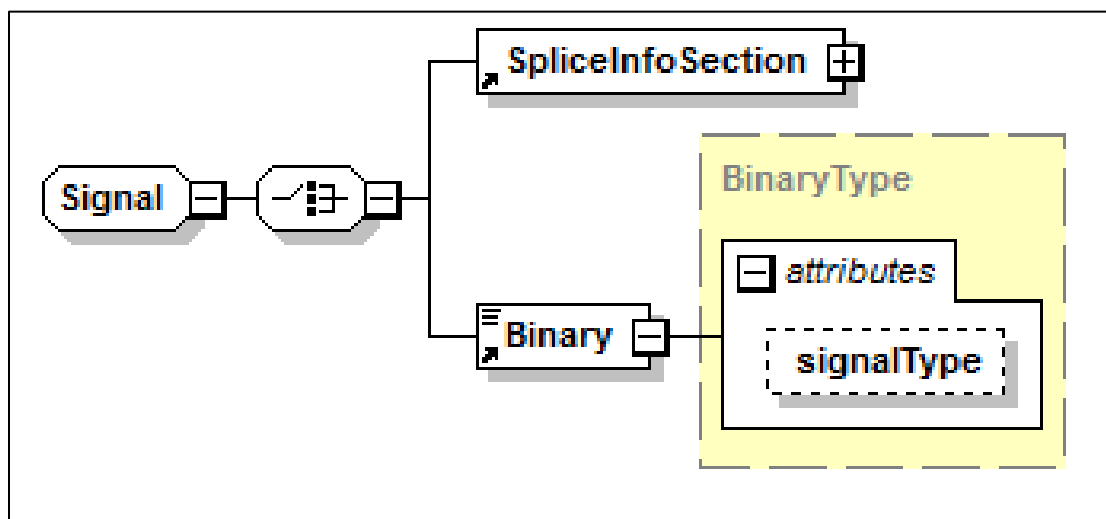


Figure 4 - SignalGroup

8. PMT Descriptors

8.1. Registration descriptor

The registration descriptor (see [MPEG Systems], table 2-46 – Registration Descriptor, clause 2.6.8) is defined to identify unambiguously the programs that comply with this standard. The registration descriptor *shall* be carried in the program_info loop of the PMT for each program that complies with this standard. It *shall* reside in all PMTs of all complying programs within a multiplex. The presence of the registration descriptor also indicates that, when signaling 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 [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 1 and below:

Table 1 - registration_descriptor()

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

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 standard.

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 2 - cue_identifier_descriptor()

Syntax	Bits	Mnemonic
cue_identifier_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
cue_stream_type	8	uimsbf
}		

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 is 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 Table 3.

Table 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 standard 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.

0x03 – 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 standard.

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 standard.

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

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 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 pre-roll 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(), splice_schedule() and splice_insert() are the legacy commands used in this standard and there is substantial installed base that expects to use these commands in order to perform ad insertions. An alternative mechanism is defined using the time_signal() command to signal splice events in new applications where there is control over the installed equipment. In order to give advance warning of an impending splice (a pre-roll function), a splice_insert() or time_signal() command could be sent multiple times before the splice point. For example, the 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.

Please refer to section 9.9.2.1 for further constraints on the use of pre-roll with splice_insert().

9.2. Legacy Command Descriptions

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 legacy 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 signaled time in the splice_info_section).

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 Section 9.9.2.2). It is recommended that this Auto Return Mode be used to support dynamic avail durations.

The bandwidth_reservation() command is provided to allow command insertion devices to utilize a consistent amount of transport stream bandwidth. Descriptors *may* be used in this command, but they cannot be expected to be processed and sent downstream to provide signaling information.

9.3. Time Signal Command

The time_signal() command is provided for extensibility by adding descriptors while preserving the precise timing allowed in the splice_insert() command. This is to allow for new features utilizing the timing capabilities of this specification while causing minimal impact to the splicing devices that conform to this specification.

A splice_insert() command typically instructs an encoder to condition the stream for a splice event. When a time_signal() command is used to signal splice events, it **shall** carry one or more segmentation_descriptors() with additional information to determine what *may* need to be done to the stream. For example, a segmentation_descriptor() with a Provider Placement Opportunity or Program Start would likely indicate that the stream should be conditioned at the time indicated.

There *may* also be out of band communications that can control the implementation for example [SCTE 224] and/or [SCTE 250].

9.4. Command Changes

There are two methods for changing the parameters of a command once it has been issued. One method is to cancel the issued command by sending a splice_info_section with the splice_event_cancel_indicator set and then to send a new splice_info_section with the correct/new parameters. The other method is to simply send a subsequent message with the new data (without canceling the old message via a cue message that has the splice_event_cancel_indicator bit set).