



***Society of Cable
Telecommunications
Engineers***

ENGINEERING COMMITTEE
Digital Video Subcommittee

AMERICAN NATIONAL STANDARD

ANSI/SCTE 67 2010

Recommended Practice for SCTE 35
Digital Program Insertion Cueing Message for Cable

NOTICE

The Society of Cable Telecommunications Engineers (SCTE) Standards are intended to serve the public interest by providing specifications, test methods and procedures that promote uniformity of product, interchangeability and ultimately 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, whether used domestically or internationally.

SCTE assumes no obligations or liability whatsoever to any party who may adopt the Standards. Such adopting party assumes all risks associated with adoption of these Standards, and accepts full responsibility for any damage and/or claims arising from the adoption of such Standards.

Attention is called to the possibility that implementation of this standard may require the use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. 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 standard 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 <http://www.scte.org>.

All Rights Reserved

© Society of Cable Telecommunications Engineers, Inc. 2010

140 Philips Road

Exton, PA 19341

Table of Contents

1	INTRODUCTION.....	- 1 -
2	INFORMATIVE REFERENCES	- 1 -
3	GLOSSARY OF TERMS AND ACRONYMS.....	- 2 -
4	OVERVIEW	- 4 -
4.1	SCOPE	- 6 -
4.2	PURPOSE	- 6 -
5	APPLICATION GUIDELINES	- 6 -
5.1	PRACTICAL BOUNDARIES FOR SPLICE_TIME() IN SPLICE_INSERT()	- 6 -
5.2	SPLICE TIME ACCURACY.....	- 7 -
5.3	SPLICE_EVENT_ID USAGE AND UNIQUENESS	- 8 -
5.4	USE OF SPLICE_SCHEDULE() COMMAND	- 10 -
5.5	COMPONENT SPLICE MODE.....	- 10 -
5.5.1	<i>Erroneous Component Splice Commands</i>	<i>- 11 -</i>
5.6	PRE-ROLL FUNCTIONALITY - ACCOMPLISHING A PRE-ROLL FUNCTION	- 11 -
5.7	CONDITIONAL ACCESS AND CUE ENCRYPTION	- 12 -
5.7.1	<i>What to Encrypt</i>	<i>- 12 -</i>
5.7.2	<i>Operation in a Cue Insertion Device.....</i>	<i>- 12 -</i>
5.7.3	<i>Operation in an Ad Insertion Device.....</i>	<i>- 12 -</i>
5.7.4	<i>Theory of Operation.....</i>	<i>- 13 -</i>
5.8	USAGE OF UNIQUE_PROGRAM_ID.....	- 18 -
5.8.1	<i>What is a "Program"?</i>	<i>- 18 -</i>
5.8.2	<i>What is a "program_id"?</i>	<i>- 18 -</i>
5.8.3	<i>Why should programs be identified and differentiated?</i>	<i>- 18 -</i>
5.8.4	<i>Why does the time at which a program is scheduled not identify it?</i>	<i>- 19 -</i>
5.8.5	<i>How will a unique_program_id alleviate problems?</i>	<i>- 19 -</i>
5.9	AVAIL FIELDS USAGE	- 19 -
5.9.1	<i>What is an Avail?</i>	<i>- 19 -</i>
5.9.2	<i>How many Avails occur within a program?</i>	<i>- 20 -</i>
5.9.3	<i>Why is it important to identify the Avails within a program?</i>	<i>- 20 -</i>
5.9.4	<i>How do the Avail fields provide for this?</i>	<i>- 20 -</i>
5.9.5	<i>What does the Avails_expected field do?</i>	<i>- 20 -</i>
5.9.6	<i>Conditional Avails</i>	<i>- 21 -</i>
5.10	CUEING USAGE	- 23 -
5.10.1	<i>Starting a Break.....</i>	<i>- 23 -</i>
5.10.2	<i>Ending a Break</i>	<i>- 24 -</i>
5.10.3	<i>Spot Sharing Within a Break</i>	<i>- 24 -</i>
5.11	CREATION AND USAGE OF PRIVATE SPLICE DESCRIPTORS.....	- 25 -
5.11.1	<i>What are Descriptors.....</i>	<i>- 25 -</i>
5.11.2	<i>Registration.....</i>	<i>- 25 -</i>

5.11.3	Creating Compatible Private Descriptors.....	- 26 -
5.11.4	Using the avail_descriptor.....	- 27 -
5.11.5	Using the DTMF Descriptor.....	- 27 -
5.12	HANDLING TIME BASE DISCONTINUITIES	- 29 -
5.13	CASCADED SPLICING DEVICES.....	- 30 -
5.13.1	Restamping SCTE 35 Cue Messages	- 31 -
5.13.2	Cue Propagation.....	- 31 -
5.13.3	Delay.....	- 32 -
5.13.4	Logical Cascading	- 32 -
5.14	USAGE OF SEGMENTATION DESCRIPTORS	- 32 -
5.14.1	Segmentation Descriptor Field Usage.....	- 32 -
5.14.2	Delivery of Segmentation Descriptors.....	- 34 -
5.14.3	Processing of Segmentation Descriptors.....	- 34 -
5.14.4	Specific use cases.....	- 35 -
5.15	BANDWIDTH RESERVATION COMMAND	- 40 -
5.15.1	Why use a Bandwidth Reservation Command?	- 40 -
5.16	HEARTBEAT MESSAGES	- 40 -
5.16.1	Why use a Heartbeat Message.....	- 40 -
5.17	TIME SIGNAL COMMAND	- 41 -
5.17.1	Uses for the Time Signal Command	- 41 -
6	ADDITIONAL INFORMATION.....	- 41 -
6.1	CONSIDERATIONS FOR EVALUATION OF MPEG-2 SPLICING DEVICES	- 41 -
6.1.1	Overview	- 41 -
6.1.2	Splicer Technology	- 42 -
6.1.3	Environment.....	- 43 -
6.1.4	Splicer Performance	- 46 -

List of Figures

Figure 1 - System Overview.....	- 5 -
Figure 2 – SCTE 35 Cue Message Insertion Points.....	- 9 -
Figure 3 - DES ECB Example	- 15 -
Figure 4 - DES CBC Encryption Example	- 15 -
Figure 5 - DES CBC Decryption Example	- 16 -
Figure 6 - Triple-DES ECB Encryption Example	- 17 -
Figure 7 - Triple-DES ECB Decryption Example	- 17 -
Figure 8 - Cascading of Splicer / Server Devices	- 31 -

List of Tables

Table 5-1 -- Avail incrementing/skipping Example	- 22 -
Table 8-2. (of SCTE 35 [1]): splice_descriptor().....	- 26 -

Table 8-3. (of SCTE 35 [1]): avail_descriptor() - 26 -

Recommended Practice for SCTE 35

Digital Program Insertion Cueing Message for Cable

1 Introduction

The goal of this Interpretation document is to serve as an informational enhancement to SCTE 35, Digital Program Insertion Cueing Message for Cable. SCTE 35 is necessarily brief in many areas in order to maintain conciseness and accuracy. This document serves as a companion to SCTE 35.

2 Informative References

At the time of publication, the editions indicated below were current. All standards are subject to revision, and parties to agreement based on this document are encouraged to apply the most recent editions of the documents listed below when appropriate.

- [1] ANSI/SCTE 35 2007 - Digital Program Insertion Cueing Message for Cable.
- [2] ANSI/SCTE 30 2009 - Digital Program Insertion Splicing API.
- [3] ITU-T Rec. H.222.0 / ISO/IEC 13818-1 2000 - Information technology – Generic coding of moving pictures and associated audio information: systems
- [4] ITU-T Rec. H.262.0 / ISO/IEC 13818-2 2000 - Information technology – Generic coding of moving pictures and associated audio information: video
- [5] ISO/IEC 13818-4 2004 - Information technology — Generic coding of moving pictures and associated audio information — Part 4: Conformance testing
- [6] SMPTE 312M 2001- Splice Points for MPEG-2 Transport Streams
- [7] ANSI/SCTE 40 2004- Digital Cable Network Interface Standard
- [8] ANSI/SCTE 118-1 2006 – Program-Specific Ad Insertion - Data Field Definitions, Functional Overview and Application Guidelines.
- [9] ANSI/SCTE 118-2 2007 – Program-Specific Ad Insertion – Content Provider to Traffic Communication Applications Data Model.
- [10] ANSI/SCTE 118-3 2006 – Program-Specific Ad Insertion – Traffic System to Ad Insertion System File Format Specification.
- [11] ANSI/SCTE 54 2009 - Digital Video Service Multiplex and Transport System Standard for Cable Television.

3 Glossary of Terms and Acronyms

Throughout this document, the terms used have specific meanings. Because some of the terms that are defined in ISO/IEC 13818-1 [3] have very specific technical meanings, the reader is referred to the original source for their definition. For terms used in this document, brief definitions are given below.

TERM	DESCRIPTION
Access Unit	The coded representation of a video picture or an audio frame [3].
Analog Cue Tone	In an analog system, a signal which 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.
ATSC	Advanced Television Systems Committee
Avail	Time space provided to cable operators by cable programming services during a program for use by the CATV operator; the time is usually sold to local advertisers or used for channel self promotion.
Break	Avail or an actual insertion in progress.
CBC	Cipher Block Chaining. This is a specific method of encryption. It is one of the methods used in DES.
CBR	Constant Bit Rate
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 be spliced.
CRC	Cyclic Redundancy Check. A method to verify the integrity of a transmitted Message.
Cueing Message	See Message.
DES	Data Encryption Standard. A method for encrypting data with symmetric keys.
DVB	Digital Video Broadcasting. An international consortium for the development of digital television systems.
SCTE 35 Cue Message	See Message.
ECB	Electronic Code Book. This is a specific method of encryption. It is one of the methods used in DES.

TERM	DESCRIPTION
ECM	Entitlement Control Message. These are private conditional access information messages which specify control words and possibly other, typically stream-specific, scrambling and/or control parameters.
EMM	Entitlement Management Message. These are private conditional access information messages which specify the authorization levels or the services of specific decoders. They may be addressed to single decoders or groups of decoders.
Event	A splice event or a Viewing Event as defined below.
In Point	A point in the stream, suitable for entry, that lies on an Access Unit boundary.
Message	In the context of this document a message is the contents of any splice_info_section.
MPTS	A Multi Program Transport Stream.
Out Point	A point in the stream, suitable for exit, that lies on an Access Unit boundary.
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].
PID	Packet identifier; a unique 13-bit value used to identify elementary streams of a program in a single or multi-program Transport Stream [3].
PID stream	A stream of packets with the same PID within a transport stream.
PMT	Program Map Table [3].
Presentation Time	The time that a presentation unit is presented in the system target decoder [3].
Program	A collection of video, audio, and data PID streams which share a common program number within an MPTS [3].
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' and indicates that the Message refers to a Program Splice Point and that all PIDs/components of the program are to be spliced.

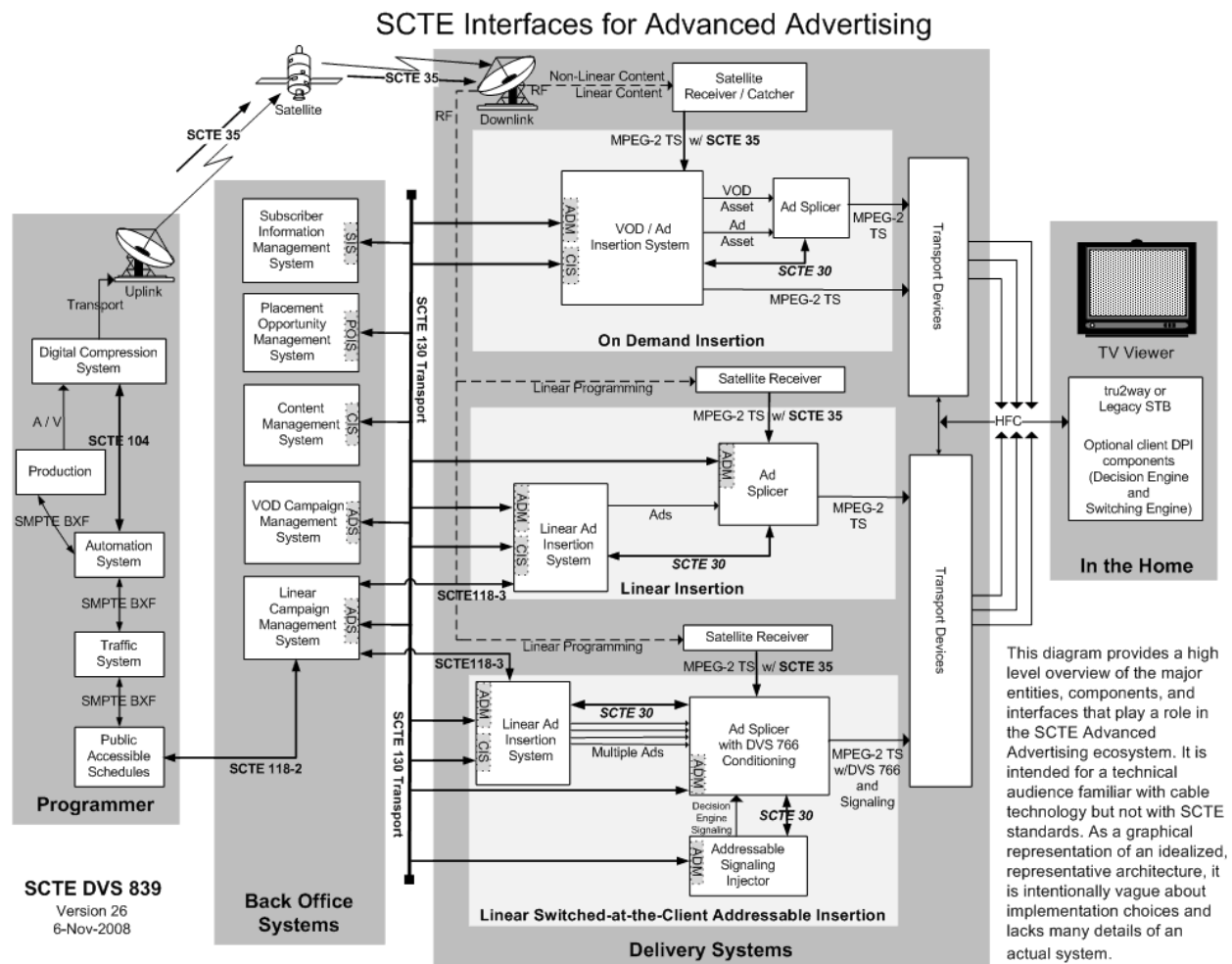
TERM	DESCRIPTION
Program Splice Point	A Program In Point or a Program Out Point.
PTS	Presentation Time Stamp [3].
Registration Descriptor	Carried in the PMT of a program to indicate that, when signaling Splice Events, splice_info_sections is carried in a PID stream within this program. The presence of the Registration Descriptor signifies a program's compliance with SCTE 35 [1].
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 standard. Unless otherwise specified in SCTE 35 [1], all reserved bits are set to '1'.
Splice Event	An opportunity to splice one or more PID streams.
Splice Immediate Mode	A mode of the Cueing Message whereby the splicing device chooses 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", which is 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.
SPTS	A Single Program Transport Stream.
T-STD	Transport Stream System Target Decoder
uimsbf	Unsigned integer, most significant bit first
VBR	Variable Bit Rate
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 Overview

The SCTE 35 standard [1] supports the splicing of MPEG-2 transport streams for the purpose of Digital Program Insertion, which includes insertion of advertisement and other content types. An in-stream messaging mechanism is defined in SCTE 35 [1] to signal splicing and insertion opportunities. A splicing device is free to ignore Splicing Events signaled by the SCTE 35 Cue Message because the Message is not a command to splice, but is an indicator of the presence of an ad Avail. The taking of an Avail is optional.

As shown in the following diagram, SCTE 35 Cue Messages are received and acted upon in the cable system headends by splicer and server devices to affect the insertion of local advertisements by splicing the ad bit stream (typically containing the commercial content) into the bit stream of programming content. SCTE 35[1] does not differentiate between a splicing device and a server, as does SCTE 30[2]. When SCTE 35[1] uses the terms "splicer" or "splicing device" the meaning of the sentence may apply to a splicer/server combination as well. In actual

practice it is common for ad servers (and not splicers) to parse, interpret and initiate action upon SCTE 35 Cue Messages. Since splicer and server devices can be combined into one, this document often uses the term server/splicer to denote a device or set of devices that together perform both functions. This block diagram describes the overall functionality and interoperability associated with the systems that accomplish this.



In Figure 1, a compliant MPEG-2 transport stream (either Multi Program Transport Stream or Single Program Transport Stream) is assumed for the network stream. No further constraints beyond the inclusion of the defined Cueing Messages are placed upon the stream. It is expected that transport packet boundary splicing, as intended by ISO/IEC 13818-1[3] and by SMPTE 312M[6], will not be suitable in cable plants due to as the use of statistical multiplexing (VBR) and progressive refresh (no I-frames); both of which may require the removal of the transport layer.

SCTE 35[1] specifies a technique for carrying notification of upcoming splice points in the transport stream. A splice information table is defined for notifying downstream devices of

Splice Events, such as a network Break or return from a network Break. The splice information table, which pertains to a given program, is carried in a separate PID 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. However, remultiplexers may need to obey certain constraints when they carry the SCTE 35 Cue Message. These constraints are addressed in SCTE 35[1] and are elaborated on within this document.

SCTE 35[1] does not address constraints on splicing devices and SCTE 35's [1] splice_info_table syntax never suggests picture or splice quality. SCTE 35[1] is not intended to guarantee seamless splicing.

4.1 Scope

This document is an informational companion to SCTE 35[1]. It is not in itself a specification or a standard. The information within is intended as guideline information. Where this document contradicts SCTE 35[1], SCTE 35[1] takes precedence.

4.2 Purpose

The purpose of this document is to aid splicing equipment designers, ad insertion equipment designers and purchasers and users of such equipment. Also expected to be interested are the networks that will originate SCTE 35 Cue Messages from their uplink sites and the manufacturers of the equipment to do this. This document is also expected to aid in the system integration of advertising related equipment, both at the Message origination end and at the Message reception end.

There may be crucial information within this document for manufacturers of equipment that pass the SCTE 35 Cue Message as part of the MPEG stream. An example of such equipment is a rate altering re-multiplexer, which performs complex processing of the stream. When the stream is demultiplexed and processed and then re-multiplexed, it is very important to place the SCTE 35 Cue Message in the proper position relative to the video service and relative to nearby time base discontinuities. Such equipment may also be required to alter the Message before retransmission.

5 Application Guidelines

5.1 Practical Boundaries for splice_time() in splice_insert()

How far ahead of the splice must a splice_insert Message be sent, relative to the picture it refers to, in order to be safely responded to by an ad insertion system?

The "arm time" denotes the time an SCTE 35 Cue Message precedes that actual insertion. The arm time is typically in the range of 5 - 8 seconds. This is in line with the pre-roll time for analog cue-tones. The arm time cannot be so short that the Avail passes by before the ad insertion system has time to respond. A minimum of 4 seconds is required for "start" Cue Messages (out_of_network_indicator = 1) and a minimum of 4 seconds is recommended for "stop" Cue Messages (out_of_network_indicator = 0).

More thought about the possible consequences is required if it is desirable to extend the arm time beyond the recommended eight seconds. To specify a maximum arm time therefore seems premature.

The splice_info_section itself does not have any PTS or DTS associated with it. Therefore it is not defined when the Message should be decoded or when it should be presented, i.e. when it should take effect. By choosing the minimum required arm time as 4 seconds, the problem of defining how this time should be measured can be avoided. Any reasonable measuring method will suffice.

5.2 Splice Time Accuracy

Although precise splice times can be specified using both the Program Splice Mode and Component Splice Mode a server/splicer may choose to perform splices at other than these precise times. Since most splicing devices operate in the compressed domain they are typically constrained to only perform splices at specific locations in the stream if the splice is to meet the semantic and syntactic requirements of the MPEG-2 Systems specification [3] and the relevant video coding specification.

See Section 6 of the document for stream requirements for splicing MPEG-2 video. Stream requirements for splicing AVC video are described in DVS714. While the details of Out Point requirements differ, both video codecs require that In Points be intra-coded pictures with no stream dependencies on pictures prior to the In Point (a closed GOP in MPEG-2, an IDR in AVC). In time shifted environments (for example VOD), where ads may be inserted into content, the Out and In Points in the entertainment content may be coincident and, if coincident, needs to meet the requirements of both an In Point and an Out Point.

If the signaled splice times are not aligned with these locations, the server/splicer may choose nearby In or Out Points, using the splice times as guidelines, if the resulting splice is to be syntactically seamless.

Splicing between AVC streams with differing GOP structures and/or buffer delays may place further requirements on the selection of splice points. Please refer to DVS714 for details.

In the case of audio the video Presentation Time of the splice will not in general align with the audio Presentation Time(s) and, therefore, corresponding audio splices may be executed at times that are "close" to the nearest video Presentation Time. Consequently, it may be necessary for a server/splicer to replace existing audio around the splice point with silence to ensure a clean transition.

Note that specific server/splicer implementations may place further, or fewer, restrictions on In and Out Points depending upon their capabilities. Section 6 of this document discusses various aspects of server/splicer capabilities and performance.

In order for a splice to be visually seamless, that is occurring at the desired location in the content with no visible transition (for example, black frames) the specified splice points should align with the GOP structure of the content. This can be accomplished by various means:

- Encoders that insert splice points under control of an automation system may adjust the GOP structure around the splice point so that it meets In or Out Point requirements without impacting video quality
- Systems that insert SCTE 35 Cue Messages into existing content may re-encode the GOP surrounding the desired splice point so that it meets In or Out Point requirements
- Since splice points will often coincide with scene Breaks or changes, encoders that employ scene change detection will often generate streams where GOP structures naturally align with splice points (for example, beginning each new scene with a closed GOP or IDR would enable scene Breaks to be used as splice points after encoding).

If the content cannot be encoded such that the desired splice points align with the GOP structure, other techniques can be used such as encoding a number of black frames around the time of the desired splice point to minimize the visual effect if the server/splicer performs the splice at the nearest In or Out Point.

Some server/splicers may be capable of inserting black or repeat frames, and corresponding audio silence, between the nearest valid In or Out Point and the specified splice location to minimize visual effect of adjusting the splice point. If this is used in an environment where an ad is replacing an existing ad in the network feed the server/splicer is required to ensure that the combined length of the black frames and the inserted content does not exceed the length of the replaced ad. Similarly, there is the possibility that the length or GOP structure of the replacement ad may not precisely align with the original. In that event a splicer may insert black frames to align with the network feed GOP structure when returning from the ad.

Thus, behavior when requested splice points do not align with actual splice points will be one factor that differentiates between splicers.

5.3 Splice_event_id Usage and Uniqueness

SCTE 35 Cue Messages can be created by several means: from information embedded in the original audio / video source, by uplink event trigger systems or by headend event trigger systems. When all of these sources are combined within a service there is potential for collisions in the splice_event_id value chosen for the SCTE 35 Cue Message. The 32-bit splice_event_id should therefore be partitioned in the following way to allow each source a unique range of splice_event_id values:

Syntax	Bits	Type
Event_source	4	uimsbf
Event_number	28	uimsbf

Event_source – A user assigned number for the source of the SCTE 35 Cue Message

Event_number – A number chosen by the Event source to identify an instance of the SCTE 35 Cue Message.

The splice_event_id serves to identify a particular instance of an opportunity to change the multiplex. At least two distinct splice_events are required to perform one insertion, one to initiate the insertion and one to end it (unless the duration is included in the Message signaling the splice-out).

Each Event is required to have a unique splice_event_id. A splice_event_id cannot be re-used before the splicing opportunity it describes (fully or in part) has completely passed. A splice_event_id is regarded to be in-use from the first SCTE 35 Cue Message where it appears until about one second after the splice time that it is associated with. It may be reused for a new Splice Event immediately afterwards. It is not possible to use the same splice_event_id to signal an Avail's start and stop if the stop is signaled before the start has been executed. However, it is possible to reuse the same splice_event_id if the first stop Message is sent after the start has been executed.

The advertising industry uses additional information besides the splice_event_id to identify the actual material being played and the time of its insertion. This information includes: service name, time, ad-agency and spot number.

As long as the Event_source is unique for each point at which SCTE 35 Cue Messages can be inserted in the following chain the Event identifiers will not collide:

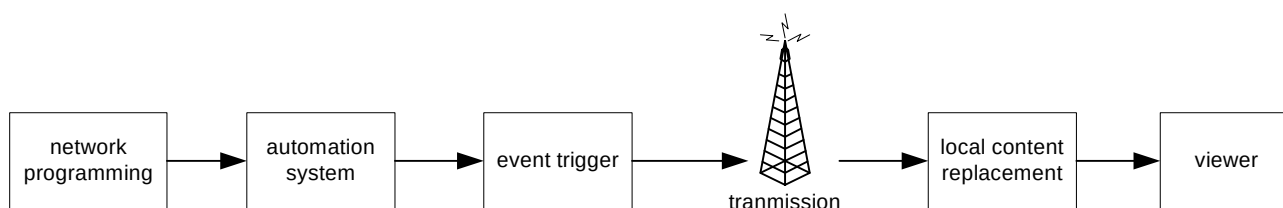


Figure 2 – SCTE 35 Cue Message Insertion Points

The following values for Event_source are suggested:

Source of SCTE 35 Cue Message	Event_source Value	Example splice_event_id ranges
Cue embedded in original source material	0	0x00000000, 0x00000001 0x0fffffff
Cue created by automation system switching	4	0x40000000, 0x40000001 0x4fffffff
Cue created by live Event trigger system	6	0x60000000, 0x60000001 0x6fffffff
Cue created by local content replacement system	12	0xC0000000, 0xC0000001 0xCfffffff

5.4 Use of splice_schedule() Command

The splice_schedule() command is intended for announcements of large schedules. It is not intended for the announcement of single Events. Current implementations of SCTE 35[1] focus on the support of single Events that do not rely on the use of the splice_schedule() command.

5.5 Component Splice Mode

The primary reason for the Component Splice Mode is to allow the replacement of or the passage of individual elementary streams of any type in a manner consistent with the program content and the advertiser's intent.

For example, during a local commercial Break, it might be useful to allow informational data (e.g. market returns) that is being streamed from the network provider to continue, although the viewer is being shown a locally inserted commercial. Conversely, with an advanced set-top box implementation, one might offer the viewer information to be downloaded for later review as part of the commercial. In this case, if the data content exceeds the duration of the commercial, the advertising data might be continued, although the program has resumed.

The methodology defined in SCTE 35[1] allows some data types to be passed while allowing others to be blocked or replaced during a Break. The decision which to pass and which to block is made by the Message inserter. A splicing device may choose to behave differently if it knows better or is commanded to do so by an entity within the headend.

It is the intention of the standard, in both the Program Splice Mode and the Component Splice Mode, that the splicer should have a great deal of freedom in choosing the actual Access Units (both video and audio) on which to splice. Some equipment will provide frame accurate splices and some may not.

The signaled splice_time may or may not be on an anchor frame or I-frame and the splicer may need to round off to a frame that makes sense for its capabilities.

There appears to be industry consensus that in Program Splice Mode it makes the most sense to give a Presentation Time of a video frame and then let the splicer choose whichever audio frame is closest or which makes sense for its methods. This wasn't, however, specified in the standard. It is believed that the industry will come up with the best way to use the tools (especially if it differs from what is stated here).

In the Component Splice Mode, things work the same way. A single splice_time should be used (called the default splice_time). It is optional to give a splice_time for each component. There is some concern in the industry that the creator of a Message will try to "help" the splicer by giving the exact time of all components and possibly complicate the job of the splicer (or the splicer may need to ignore the splice_time for some components).

It is understood that SMPTE 312M[6] requires a time per component to define the exact Access Unit for each component to be spliced. The cable industry is looking at a different usage for the time per component approach, however. SCTE 35[1] allows individual components to start or end at very different times from the normal beginning or ending of the Break. For instance, a "data" component such as a Java applet may have to be downloaded to the set-top box a number of seconds prior to the ad in order to support the ad. This is the primary usage of the unique splice_time for each component.

5.5.1 Erroneous Component Splice Commands

When a splice Message (in component mode) specifies invalid component tags, the splice should be executed for all correctly identified components. The intent is to let insertion succeed if possible, even if the complete chain of devices results in a violation of the standard. Several potential cases are given below.

In case a device removes one of several audio streams after the splice Message has been inserted, but fails to update the splice Messages (either because it is unaware of them or other reasons), the combination of stream and Message at the server/splicer will be invalid. It is, however, still possible for the server/splicer to perform a valid insertion.

In case a device adds a component (e.g. a data channel), but fails to update the splice Message (same possible reasons as above), it is still possible to perform a valid insertion.

Whenever a discrepancy between actual stream and splice Message exists, a server/splicer should perform the insertion, in order to add error resilience to the chain.

5.6 Pre-Roll Functionality - Accomplishing a Pre-Roll Function

A Pre-Roll function was necessary in the days of analog ad insertion to allow time for a tape machine to get up to speed by the beginning of the ad Avail. So Analog Cue Tones were often sent about 5 to 8 seconds prior to the Avail (each network used its own chosen time and the consistency was varying). The early arrival of the cue tone has remained unchanged during the days of hybrid ad insertion where tape machines are replaced by MPEG servers and decoders, but the insertion is still analog. This early cue tone is useful for digital ad insertion as well. It gives the ad insertion equipment time to access its ad database, to determine which ad to play next, to start accessing its disk drives and to fill the MPEG decoder pipeline.

The splice_insert() command is constructed such that a pre-roll time can be used. It is also possible to repeat a splice_insert() command to increase the error resilience of the system.

The simplest variant is to send the splice_insert() command once about 8 seconds prior to an Avail. This is similar to the analog insertion case.