Primetime Digital Program Insertion Signaling Specification 1.2

Primetime Digital Program Insertion Signaling Specification Version 1.2

Published July 2016

Introduction

This document describes the signaling used by Adobe Primetime to manage digital program insertion (DPI) into live and linear streams. Primetime DPI capabilities include both ad insertion/replacement as well as blackouts and insertion of alternate content. Signaling is defined for RTMP, HDS, and HLS.

This document does not describe the expected behaviors of a client processing the DPI signals, it only defines the formats used and how they should be embedded into different types of streams.

Updates

Updates in Version 1.1

Version 1.1 of this specification contained mainly editorial corrections, clearing up the data types and string formats used in the specification. No technical changes we issued in this version of the document.

Updates in Version 1.2

Version 1.2 of this specification adds support for carrying DPI signals in MPEG-DASH streams, aligning this specification to similar signaling specifications described by SCTE and the DASH Industry Forum. This version also details how this specification relates to the recently updated SCTE 67 guidelines for carrying SCTE 35 in various streaming protocols. Clarifications have been made to forbid carrying both Simple Mode and SCTE 35 Mode cues in the same content stream.

A number of corrections have been made to the HLS cue syntax description and examples to fix discrepancies between the normative syntax description and the examples, including adding correct quotation marks to ID strings. New language has been added to the HLS section to describe requirements around timeline synchronization in ABR sets and the use of continuation tags for live stream tune-in.

Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

“MUST”, “REQUIRED” or “SHALL”, mean that the definition is an absolute requirement of the specification.

“MUST NOT” or “SHALL NOT” means that the definition is an absolute prohibition of the specification.

“SHOULD” or “RECOMMENDED” mean that there may be valid reasons to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

“SHOULD NOT” or “NOT RECOMMENDED” mean that there may be valid reasons when the particular behavior is acceptable, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

“MAY” or “OPTIONAL” mean the item is truly optional.

References

[RFC2119] IETF RFC 2119, Keywords for use in RFCs to Indicate Requirements Levels, S. Bradner, March 1997, (http://www.ietf.org/rfc/rfc2119.txt?number=2119)

[SCTE35] “Digital Program Insertion Cueing Message for Cable”, https://www.scte.org/documents/pdf/Standards/SCTE%2035%202013.pdf

[SCTE67] "Recommended Practice for SCTE 35 Digital Program Insertion Cueing Message for Cable", http://www.scte.org/documents/pdf/standards/SCTE%2067%202014.pdf

[SCTEDASH] "MPEG DASH for IP-Based Cable Services Part 1: MPD Constraints and Extensions", In Progress

[AMF0] "Action Message Format – AMF 0", http://download.macromedia.com/pub/labs/amf/amf0_spec_121207.pdf

[XML10] "Extensible Markup Language (XML) 1.0, Fifth Edition", http://www.w3.org/TR/REC-xml/

[XMLS2] "XML Schema Part 2 Data Type, Second Edition", http://www.w3.org/TR/xmlschema-2/

[HLS] IETF Informational Draft, "HTTP Live Streaming, version 12", http://tools.ietf.org/html/draft-pantos-http-live-streaming-12

[ISO23009-1] ISO/IEC 23009-1 "Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats"

Definitions

Distributor Advertisement

A segment of the presentation that carries an advertisement placed by the distributor.

Placement Opportunity

A segment of the presentation that represents one or more advertisements. The ads within the placement opportunity may or may not be further segmented into provider ads and distributor ads.

Program

A segment of the presentation that represents a unique program, as described by its airing ID.

Provider Advertisement

A segment of the presentation that carries an advertisement placed by the provider.

Segment

A semantically meaningful sequence of time within a linear presentation. Examples are programs and placement opportunities. Multiple segments of different types may be active at one time, but only one segment of a single type may be active at once. For example, a placement opportunity may be active within a program, but one placement opportunity may not overlap another placement opportunity.

Splice

A generic sequence of time within a linear presentation where advertising or alternate content may be "spliced" in.

Signaling Modes

The Primetime DPI format support two major signaling modes:

- **Simple**. In this mode whole-pod ad replacement is achieved using a single "splice out" signal that contains the pod duration. Other metadata about the pod may be provided to facilitate ad decisioning. No return signal is required.
- **SCTE-35**. This mode supports the entire SCTE-35 cue set defined in [SCTE35]. This mode of operation supports both splice signals and segmentation signals provided by SCTE-35. Unlike the "simple" mode we do not assume that the duration of a splice or segment are known up front. Instead, the semantics of the SCTE-35 definition of splices and the various segment types (programs, chapters, placement opportunities, etc.) are taken into account to manage the entry and exit point of the splice or segment.

Content MAY contain either Simple Mode or SCTE 35 Mode signals, but MAY NOT carry both simultaneously. This restriction is intended to prevent any ambiguity as to whether multiple, simultaneous cues are intended as different source signals, or as alternate representations of the same source signal.

Splices and Segments

The Primetime DPI model supports two types of time sequences to be described within the presentation, "splices" and "segments". Splices represent a generic time sequence within the presentation where alternate content may be spliced in. Splices usually, but not always, signal a time sequence where advertising content should be placed into the presentation. Segments represent semantically meaningful sequences of time within a linear presentation, such as a program or ad placement opportunity.

While the signaling for splices and segments are different, they share a common set of properties and behaviors. All splices and segments have the following properties:

- ID. Each splice or segment has a unique identifier that allows it to be differentiated from other splices or segments. These identifiers may not be unique across the set of splices and the set of segments.
- Start Time. Each splice or segment has a specific start time within the presentation timeline.
- Duration. Each splice or segment has a duration. This duration may not be known when the splice or segment begins.
- Elapsed Time. As each splice or segment advances it has a known elapsed time from when the splice or segment began.

More details about splices and segments are described below.

> **Important Note**: The term "segment' in this document refers to a description of a portion of the content stream, as it is used in [SCT35]. This **should not** be confused with HLS segments. In this document "segment" shall refer to the SCTE-35 definition of the term. When referring to HLS segments we will use the term "HLS segment".

Splices

Splices are the simpler of the two signal types. Only a single splice may be active at one time. In "simple" mode the duration of the splice is assumed to be known when the splice begins. In "SCTE-35" mode the end point of the splice must be signaled explicitly.

Mapping to SCTE-35

Splices correspond to the time sequences described by the SCTE-35 splice_insert() command type (splice_command_type = 0x05), as defined in [SCTE35]. The splice begins when a splice_insert() command with the out_of_network indicator set to '1' is received. In SCTE-35 mode the break duration is optional, and the end of the splice is signaled when a splice_insert() command with the out_of_network indicator set to '0' is received. In simple mode the break_duration must be set, and any splice_insert() commands received with the out_of_network indicator set to '0' will be ignored.

The "time" attribute shall correspond to the splice_time() field of the SCTE-35 cue, mapped into the appropriate units and bitspace. As the range of the "time" attribute may be larger than the splice_time() field of the cue processors shall allow for rollover of the splice_time() field. Processors shall apply any pts_offset specified in the SCTE-35 cue when interpreting the splice command.

Segments

While there is only a single type of splice, there are multiple types of segments that may be signaled at one time. Only one segment of each type may be signaled at once. The following segment types are supported:

- Program
- Chapter
- Provider Advertisement
- Distributor Advertisement
- Placement Opportunity

Mapping to SCTE-35

Segments correspond to the equivalent time sequences described by a segmentation descriptor carried within a SCTE-35 time_signal() command (splice_command_type = 0x09), as defined by [SCTE35]. The following table describes this mapping in more detail.

Segment	Start Signal (segmentation_type_id)	End Signal (segmentation_type_id)
Program	0x10	0x11
Program Overlap	0x17	0x11
Chapter	0x20	0x21
Provider Advertisement	0x30	0x31
Distributor Advertisement	0x32	0x33
Placement Opportunity	0x34	0x35

The "time" attribute shall correspond to the splice_time() field of the SCTE-35 cue, mapped into the appropriate units and bitspace. As the range of the "time" attribute may be larger than the splice_time() field of the cue processors shall allow for rollover of the splice_time() field. Processors shall apply any pts_offset specified in the SCTE-35 cue when interpreting the splice command.

When multiple segmentation descriptors are signaled within a single SCTE-35 cue, a unique splice or segment cue shall be placed into the stream or manifest, using the appropriate "ID" and "time" values for each splice or segment. This requirement ensures there is no ambiguity for any of the attributes in the output signal. The binary SCTE-35 cue carried in the "cue" attribute may include all segmentation descriptors carried in the original cue. It is up to the client to select the appropriate descriptor out of the binary data when parsing the binary SCTE-35 cue.

Relationship of this Specification to SCTE 67

The latest revision of [SCTE67] describes the carriage of SCTE 35 signaling in various streaming protocols. The ad signaling defined in this specification is, in general, compatible with the signaling defined in SCTE 67. Several of the fields or attributes that are defined as optional in SCTE 67 are required by Primetime. It is recommended that implementors of SCTE 67 support these additional fields in order to create content that is compatible both with SCTE 67 and with Primetime.

Signals in Adaptive Bitrate Sets

All renditions of an adaptive bitrate set shall carry the same set of signals for the presentation. Signal identifiers for the corresponding signal in each rendition MUST be equal.

RTMP Signal Format

RTMP ad signals are sent as AMF cue messages embedded within the RTMP stream. The cue messages may be sent some time before the actual splice or segment event. To accommodate this actual time of the splice or segment is sent within the cue message.

Each cue message may be repeated in order to accommodate viewers that tune in after the start of the splice or segment. Repeated cue messages can be identified by the client using the id field present on each message. In addition the optional elapsed time field can be used to gauge how far the splice or segment has progressed.

Signal Syntax

The syntax of the signal message conforms to the object and field types defined in [AMF0].

Simple Mode

For RTMP simple mode we define a single AMF cue message called "onAdCue" with the following format:

Field Name	Field Type	Required?	Description
type	String	Required	Shall be "SpliceOut" to designate a simple mode splice.
id	String	Required	Shall be a unique identifer describing the simple mode splice.
duration	Number	Required	Shall be the duration of the splice. Units are fractional seconds.
elapsed	Number	Optional	When the signal is being repeated in order to support tune in, this field shall be the amount of presentation time that has elapsed since the splice began. Units are fractional seconds. In simple mode this value should not exceed the original duration of the splice.
time	Number	Required	Shall be the time of the splice, in presentation time. Units are fractional seconds.

SCTE-35 Mode

For RTMP SCTE-35 mode we define a single AMF cue message called "onAdCue" with the following format:

Field Name	Field Type	Required?	Description
cue	String	Required	Shall be a string carrying the base64 binary SCTE-35 cue that this cue message represents.
type	String	Required	Shall be "scte35" to designate a SCTE-35 mode splice or segment.
id	String	Required	Shall be a unique identifer describing the splice or segment.
duration	Number	Required	Shall be the duration of the splice or segment, if known. This field shall be 0 when the duration is not known. Units are fractional seconds.
elapsed	Number	Optional	When the signal is being repeated in order to support tune in, this field shall be the amount of presentation time that has elapsed since the splice began. Units are fractional seconds. In SCTE-35 mode this value may exceed the original specified duration of the splice or segment.
time	Number	Required	Shall be the time of the splice, in presentation time. Units are fractional seconds.

HDS Signal Format

HDS ad signals are sent as XML elements embedded in the HDS F4M. Optionally, AMF cue messages may be placed into the HDS data track. These cue messages shall be formed as "onAdCue" messages as defined in "RTMP SIgnal Format".

Note that while the HDS cue format is defined such that repeating the cue multiple time is semantically meaningful, just as it is for RTMP, it is not generally required since the <cue> element is described outside of the content stream and will never be "missed" during tune in. All cues that refer to the current content window may remain in the manifest until they are no longer needed.

Signal Syntax

The HDS signals are encoded as XML for carriage in the HDS F4M manifest. The structure of the signals conforms to [XML10].

Attribute type values used in this specification correspond to primitive XML Schema data types defined in [XMLS2] as follows:

Attribute Type	XML Schema Data Type
Number	double

String	string

Root Element

The **cueInfo** element is embedded as a child of the F4M **manifest** element as defined in [F4M].

cueInfo element

Child Element	Element Type	Multiplicity	Description
cue	cue	0..*	Zero or more cue elements, ordered by time

Simple Mode

cue element

Attribute Name	Attribute Type	Required?	Description
type	String	Required	Shall be "SpliceOut" to designate a simple mode splice.
id	String	Required	Shall be a unique identifer describing the simple mode splice.
duration	Number	Required	Shall be the duration of the splice. Units are fractional seconds.
elapsed	Number	Optional	When the signal is being repeated in order to support tune in, this field shall be the amount of presentation time that has elapsed since the splice began. Units are fractional seconds. In simple mode this value should not exceed the original duration of the splice.
time	Number	Required	Shall be the time of the splice, in presentation time. Units are fractional seconds.

SCTE-35 Mode

cue element

Attribute Name	Attribute Type	Required?	Description
cue	String	Required	Shall be a string carrying the base64 binary SCTE-35 cue that this cue message represents.
type	String	Required	Shall be "scte35" to designate a SCTE-35 mode splice or segment.
id	String	Required	Shall be a unique identifer describing the splice or segment.
duration	Number	Required	Shall be the duration of the splice or segment, if known. This field shall be 0 when the duration is not known. Units are fractional seconds.
elapsed	Number	Optional	When the signal is being repeated in order to support tune in, this field shall be the amount of presentation time that has elapsed since the splice began. Units are fractional seconds. In SCTE-35 mode this value may exceed the original specified duration of the splice or segment.
time	Number	Required	Shall be the time of the splice, in presentation time. Units are fractional seconds.

HLS Signal Format

HLS ad signals consist of an M3U8 extension tag embedded into the HLS M3U8 stream manifest. The tag shall be embedded into the manifest preceding the HLS segment that contains the indicated splice time.

In order to facilitate tune in and the HLS segment rolling window the cue tag shall be repeated often enough that the splice or segment is always fully described in the M3U8. The ELAPSED attribute shall be used to indicate the amount of time the splice or segment has been active.

The position of the cue tag shall always be either immediately before the first HLS segment (for splice or segment start) or immediately after the last HLS segment (for splice or segment end).

Time Synchronization for Adaptive Bitrate Sets in HLS

Unlike RTMP, HDS, and MPEG-DASH, HLS does not require a global presentation time (one which spans all adaptive bitrate renditions) to be present in the M3U8 manifest. Primetime ad insertion in general requires some way to determine a global presentation time from the M3U8.

The RECOMMENDED way to provide the ad insertion system time information is via the #EXT-X-PROGRAM-DATE-TIME tag defined in [HLS]. If the #EXT-PROGRAM-DATE-TIME tag is present in the M3U8, it SHALL be synchronized across all renditions (as is required by HLS) and SHALL be accurate to within 1 millisecond. In this case, accuracy means that if the HLS segment durations (as signaled in the #EXTINF tag) are added to the date/time value in the #EXT-X-PROGRAM-DATE-TIME tag, the derived time value will match across renditions. This implies that the program date/time clock moves forward at the same rate as the media presentation clock.

If the #EXT-X-PROGRAM-DATE-TIME tag is not present in the M3U8, the ad insertion system SHOULD utilize the ID field of the ad cue to determine the relationship of ad cues across renditions. In order for this to work with a rolling HLS segment window, the ad cue signal SHALL be repeated with an appropriate ELAPSED value (a "continuation tag") if the initial signal rolls out of the window, as described in the section "Handling Tune-In for HLS".

For correct operation of the ad insertion system, one of these conditions MUST be true:

- The stream signals a global presentation time, using the #EXT-X-PROGRAM-DATE-TIME tag, as described in this section.

or

- The stream continuation tags (repeated ad signals), using the ELAPSED field, as described in this section.

Handling Tune-In for HLS

Ad cues are associated with the HLS stream presentation timeline due to their placement immediately before the corresponding HLS segment in the M3U8. This positional dependency means that an ad cue signal may roll out of the stream window during the lifetime of the live stream. As the ad signal may present a portion of time that is still in the window, this behavior can lead to incorrect operation of the ad insertion system when a client tunes into the live stream after the initial ad signal has been removed from the M3U8.

In order to handle this case, it is RECOMMENDED that ad signals be repeated when the initial ad signal is removed from the window. This is known as a "continuation tag". When an ad signal is repeated, it must carry an ELAPSED attribute set to the amount of time that has elapsed in the splice or segment at the new signal position. For simplicity of implementation, a repeated ad signal MAY be repeated for every HLS segment in the ad segment/splice.

See "Time Synchronization for Adaptive Bitrate Sets in HLS" for additional requirements on the use of continuation tags.

Signal Syntax

The HLS signals are encoded as M3U tags with all attributes encoded as part of an attribute list as defined in [HLS].

Attribute type values used in this specification correspond to AttributeValue data types defined in [HLS] as follows:

Attribute Type	AttributeValue Data Type
Number	decimal-floating-point
String	quoted-string

Simple Mode

Tags

Tag	Attributes	Required?

#EXT-X-CUE	TYPE ID DURATION ELASPED TIME	Y

Attributes

Attribute Name	Attribute Type	Required?	Description
TYPE	String	Required	Shall be "SpliceOut" to designate a simple mode splice.
ID	String	Required	Shall be a unique identifer describing the simple mode splice.
DURATION	Number	Required	Shall be the duration of the splice. Units are fractional seconds.
ELAPSED	Number	Optional	When the signal is being repeated in order to support tune in, this field shall be the amount of presentation time that has elapsed since the splice began. Units are fractional seconds. In simple mode this value should not exceed the original duration of the splice.
TIME	Number	Required	Shall be the time of the splice, in presentation time. Units are fractional seconds.

SCTE-35 Mode

Note that since version 1.2 of this specification, HLS SCTE-35 mode cues may be signaled using either the original tag name (#EXT-X-CUE) or the SCTE 67 compatible tag name (#EXT-X-SCTE35). Content conforming to this specification MAY use either tag name. Clients MUST support both tag names. Content MUST NOT mix usage of the two styles of tags.

Tags

Tag	Attributes	Required?
#EXT-X-CUE or #EXT-X-SCTE35	TYPE ID DURATION ELASPED TIME	Y

Attributes

Attribute Name	Attribute Type	Required?	Description
CUE	String	Required	Shall be a string carrying the base64 binary SCTE-35 cue that this cue message represents.
TYPE	String	Required	Shall be "scte35" to designate a SCTE-35 mode splice or segment.
ID	String	Required	Shall be a unique identifer describing the splice or segment.
DURATION	Number	Required	Shall be the duration of the splice or segment, if known. This field shall be 0 when the duration is not known. Units are fractional seconds.
ELAPSED	Number	Optional	When the signal is being repeated in order to support tune in, this field shall be the amount of presentation time that has elapsed since the splice began. Units are fractional seconds. In SCTE-35 mode this value may exceed the original specified duration of the splice or segment.
TIME	Number	Required	Shall be the time of the splice, in presentation time. Units are fractional seconds.

MPEG-DASH Signal Format

MPEG-DASH ad signals SHALL be carried as event messages embedded in the MPEG-DASH MPD using the *EventStream* element, as

described in the DASH Industry Forum guidelines for ad insertion [DASHIFADIN].

While the DASH IF guidelines allow for in-stream signals, content conforming to this specification SHALL carry ad signals directly in the MPD, allowing for compatibility both with Primetime's client-based as well as server-based ad insertion systems. Content MAY carry in-stream signals in addition to the in-MPD signals.

Signal Syntax

The syntax of the signal message SHALL conform to the definition of in-MPD events, as described in [ISO23009-1].

Simple Mode

Simple mode cues SHALL be signaled using a scheme URI of "urn:com:adobe:dpi:simple:2015". Scheme-specific attributes for each *Event* elem ent are described in the following table.

Attribute Name	Attribute Type	Required?	Description
duration	Number	Required	Shall be the duration of the splice. Units are fractional seconds.
id	String	Required	Shall be a unique identifer describing the simple mode splice.
time	Number	Required	Shall be the time of the splice, in presentation time. Units are fractional seconds.

SCTE-35 Mode

SCTE 35 mode cues SHALL be signaled using the scheme ""urn:scte:scte35:2014:xml+bin", as described in the the SCTE profile for MPEG-DASH [SCTEDASH]. The semantics of the signal are consistent with the app-driven interoperability point specified in [DASHIFADIN]. This signaling is also consistent with the DASH signaling described in [SCTE67], with the exception that the referenced version of SCTE 67 describes usage of an XML-based representation for of the SCTE 35 cue. The DASH IF ad insertion guidelines allow for either a binary or an XML representation to be used. This specification restricts the cue to the binary form, for consistency with the other streaming formats supported by Primetime.

While the binary SCTE cue is carried in an XML envelope, the intention of selection of a binary format is actually to avoid the need for a full XML parser. This is possible by restricting all signaling information to that which is carried in the binary payload. A compliant processor MAY ignore any additional elements or attributes outside the <Binary> element. Processing the XML using a regular expression of the form *<*Binary*>(*)</* MUST result in a valid Base64 encoded SCTE 35 SpliceInfoSection being captured.

Appendix A - RTMP Examples

Example of RTMP Stream with Simple Mode Cues

Example shows a snapshot of a live stream with a 30 second ad break, followed later by a 60 second ad break. Note that each cue arrives (indicated by "timestamp") with 4 seconds of pre-roll before the presentation time of the signaled splice.

AMF Message 1 - method: "onAdCue", timestamp: 596

Field	Field Type	Value
type	String	"SpliceOut"
id	String	"1"
time	Number	600
duration	Number	30

AMF Message 2 - method: "onAdCue", timestamp: 656

Field	Field Type	Value

type	String	"scte35"
id	String	"2"
time	Number	660
duration	Number	60

Example of RTMP Stream with SCTE-35 Mode Cues

Example shows a snapshot of a live stream. The first cue, ID "1" signals a program start with no duration available. Cue "2" signals the beginning of an ad break. The cue is repeated once during the span of the break, with the "elapsed" field updated appropriately. The break is terminated with an end market, cue "3".

Note that the actual base64 binary data is sample data and does not correctly reflect the values in the in-manifest cues. Also note that each cue arrives (indicated by "timestamp") with 4 seconds of pre-roll before the presentation time of the signaled splice or segment.

AMF Message 1 - method: "onAdCue", timestamp: 596

Field	Field Type	Value
type	String	"scte35"
id	String	"id:1"
time	Number	600
cue	String	"/DAIAAAAAAAAAAAQAAZ/I0VniQAQAgBDVUVJQAAAAH+cAAAAAA=="
duration	Number	0

AMF Message 2 - method: "onAdCue", timestamp: 610

Field	Field Type	Value
type	String	"scte35"
id	String	"id:2"
time	Number	614
cue	String	"/DAIAAAAAAAAAAAQAAZ/I0VniQAQAgBDVUVJQAAAAH+cAAAAAA=="
duration	Number	30

AMF Message 3 - method: "onAdCue", timestamp: 626

Field	Field Type	Value
type	String	"scte35"
id	String	"id:2"
time	Number	614
cue	String	"/DAIAAAAAAAAAAAQAAZ/I0VniQAQAgBDVUVJQAAAAH+cAAAAAA=="
duration	Number	30
elapsed	Number	16

AMF Message 4 - method: "onAdCue", timestamp: 640

Field	Field Type	Value
type	String	"scte35"
id	Number	"id:3"

time	Number	644
cue	String	"/DAIAAAAAAAAAAAQAAZ/I0VniQAQAgBDVUVJQAAAAH+cAAAAAA=="
duration	Number	0

Appendix B - HDS Examples

Example of HDS F4M with Simple Mode Cues

Example shows a snapshot of a live stream with a 30 second ad break, followed later by a 60 second ad break.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns="http://ns.adobe.com/f4m/1.0" version=”2.0”>
    <id>stream1</id>
    <streamType>live</streamType>
    <baseURL>http://www.example.com/data/</baseURL>
    <bootstrapInfo profile="named" id="boot1">
        BASE64 encoding of bootstrap information
    </bootstrapInfo>
    <cueInfo>
        <cue type="SpliceOut" id="1" time="600" duration="30" />
        <cue type="SpliceOut" id="2" time="660" duration="60" />
    </cueInfo>
    <media url="stream1" bootstrapInfoId=”boot1”/>
</manifest>
```

Example of HDS F4M with SCTE-35 Mode Cues

Example shows a snapshot of a live stream. The first cue, ID "1" signals a program start with no duration available. Cue "2" signals the beginning of an ad break. The cue is repeated once during the span of the break, with the "elapsed" field updated appropriately. The break is terminated with an end market, cue "3"

Note that the actual base64 binary data is sample data and does not correctly reflect the values in the in-manifest cues.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns="http://ns.adobe.com/f4m/1.0" version=”2.0”>
    <id>stream1</id>
    <streamType>live</streamType>
    <baseURL>http://www.example.com/data/</baseURL>
    <bootstrapInfo profile="named" id="boot1">
        BASE64 encoding of bootstrap information
    </bootstrapInfo>
    <cueInfo>
        <cue type="scte35" id="id:1" time="600"
cue="/DAIAAAAAAAAAAAQAAZ/[I0VniQAQAgBDVUVJQAAAAH]\+cAAAAAA==" duration="0"
/>
        <cue type="scte35" id="id:2" time="614"
cue="/DAIAAAAAAAAAAAQAAZ/[I0VniQAQAgBDVUVJQAAAAH]\+cAAAAAA==" duration="30"
/>
        <cue type="scte35" id="id:2" time="614"
cue="/DAIAAAAAAAAAAAQAAZ/[I0VniQAQAgBDVUVJQAAAAH]\+cAAAAAA==" duration="30"
elapsed="16" />
        <cue type="scte35" id="id:3" time="644"
cue="/DAIAAAAAAAAAAAQAAZ/[I0VniQAQAgBDVUVJQAAAAH]\+cAAAAAA==" duration="0"
/>
    </cueInfo>
    <media url="stream1" bootstrapInfoId=”boot1”/>
</manifest>
```

Appendix C - HLS Examples

Example of HLS M3U8 with Simple Mode Cues

Example shows a snapshot of a live stream with a 30 second ad break, followed later by a 60 second ad break.

```
#EXTM3U
#EXT-X-TARGETDURATION:10
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:0300
#EXT-X-PROGRAM-DATE-TIME:2015-04-22T13:00:20.000+08:00
#EXT-X-CUE:TYPE="SpliceOut",ID="1",TIME=600.0,DURATION=30.0
#EXTINF:4,
stream_01.ts
#EXTINF:10,
stream_02.ts
#EXTINF:10,
stream_03.ts
#EXTINF:6,
stream_04.ts
#EXTINF:10,
stream_05.ts
#EXTINF:10,
stream_06.ts
#EXT-X-CUE:TYPE="SpliceOut",ID="2",TIME=660.0,DURATION=60.0
#EXTINF:10,
stream_07.ts
#EXTINF:10,
stream_08.ts
#EXTINF:10,
stream_09.ts
```

Example of HLS M3U8 with SCTE-35 Mode Cues

Example shows a snapshot of a live stream. The first cue, ID "1" signals a program start with no duration available. Cue "2" signals the beginning of an ad break. The cue is repeated once during the span of the break, with the "elapsed" field updated appropriately. The break is terminated with an end market, cue "3".

Note that the actual base64 binary data is sample data and does not correctly reflect the values in the in-manifest cues.

```
#EXTM3U
#EXT-X-TARGETDURATION:10
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:0300
#EXT-X-PROGRAM-DATE-TIME:2015-04-22T13:00:20.000+08:00
#EXT-X-CUE:TYPE="scte35",ID="id:1",TIME=600.0,CUE="/DAIAAAAAAAAAAAQAAZ/I0V
niQAQAgBDVUVJQAAAAH+cAAAAAA==",DURATION=0
#EXTINF:10,
stream_01.ts
#EXTINF:4,
stream_02.ts
#EXT-X-CUE:TYPE="scte35",ID="id:2",TIME=614.0,CUE="/DAIAAAAAAAAAAAQAAZ/I0V
niQAQAgBDVUVJQAAAAH+cAAAAAA==",DURATION=30.0
#EXTINF:6,
stream_03.ts
#EXTINF:10,
stream_04.ts
#EXT-X-CUE:TYPE="scte35",ID="id:2",TIME=614.0,CUE="/DAIAAAAAAAAAAAQAAZ/I0V
niQAQAgBDVUVJQAAAAH+cAAAAAA==",DURATION=30.0,ELAPSED=16.0
#EXTINF:10,
stream_05.ts
#EXTINF:4,
stream_06.ts
#EXT-X-CUE:TYPE="scte35",ID="id:3",TIME=644.0,CUE="/DAIAAAAAAAAAAAQAAZ/I0V
niQAQAgBDVUVJQAAAAH+cAAAAAA==",DURATION=0
#EXTINF:6,
stream_07.ts
#EXTINF:10,
stream_08.ts
#EXTINF:10,
stream_09.ts
```

Example of HLS M3U8 with a Continuation Tag

Example shows a snapshot of a live stream with a 60 second ad break, where the first 14 seconds of the break has been removed from the DVR window. The ELAPSED attribute signals that this break is still in progress, which allows for correct behavior when a client tunes-in to the stream. This example uses Simple Mode cues, but the usage is similar for SCTE 35 mode.

```
#EXTM3U
#EXT-X-TARGETDURATION:10
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:0302
#EXT-X-PROGRAM-DATE-TIME:2015-04-22T13:00:34.000+08:00
#EXT-X-CUE:TYPE="SpliceOut",ID="1",TIME=614.0,DURATION=30.0,ELAPSED=14.0
#EXTINF:10,
stream_03.ts
#EXTINF:10,
stream_04.ts
#EXTINF:10,
stream_05.ts
#EXTINF:10,
stream_06.ts
#EXTINF:6,
stream_07.ts
#EXTINF:10,
stream_08.ts
#EXTINF:10,
stream_09.ts
```

The same example, but with the continuation tag repeated on each HLS segment in the ad segment/splice. This method is more verbose, but may be easier to implement.

```
#EXTM3U
#EXT-X-TARGETDURATION:10
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:0302
#EXT-X-PROGRAM-DATE-TIME:2015-04-22T13:00:34.000+08:00
#EXT-X-CUE:TYPE="SpliceOut",ID="1",TIME=614.0,DURATION=30.0,ELAPSED=14.0
#EXTINF:10,
stream_03.ts
#EXT-X-CUE:TYPE="SpliceOut",ID="1",TIME=614.0,DURATION=30.0,ELAPSED=24.0
#EXTINF:10,
stream_04.ts
#EXT-X-CUE:TYPE="SpliceOut",ID="1",TIME=614.0,DURATION=30.0,ELAPSED=34.0
#EXTINF:10,
stream_05.ts
#EXT-X-CUE:TYPE="SpliceOut",ID="1",TIME=614.0,DURATION=30.0,ELAPSED=44.0
#EXTINF:10,
stream_06.ts
#EXT-X-CUE:TYPE="SpliceOut",ID="1",TIME=614.0,DURATION=30.0,ELAPSED=50.0
#EXTINF:6,
stream_07.ts
#EXTINF:10,
stream_08.ts
#EXTINF:10,
stream_09.ts
```

Appendix D - MPEG-DASH Examples

Example of MPEG-DASH MPD with Simple Mode Cues

Example shows a snapshot of a live stream with a 30 second ad break, followed later by a 60 second ad break.

```
<?xml version="1.0"?>
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="urn:mpeg:dash:schema:mpd:2011"
    xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
    type="dynamic"
    availabilityStartTime="2014-03-23T11:51:44"
    minimumUpdatePeriod="PT2S"
    timeShiftBufferDepth="PT600S"
    minBufferTime="PT4S"
    profiles="urn:mpeg:dash:profile:isoff-live:2011"
    >

    <!-- Live stream containing single period w/ embedded SCTE-35 ad
signals as EventStream -->
    <Period start="PT0S">
        <EventStream schemeIdUri="urn:com:adobe:dpi:simple:2015"
timescale="1">
   <Event id="1" presentationTime="600.0" duration="30.0"/>
   <Event id="2" presentationTime="660.0" duration="60.0"/>
        </EventStream>

        <AdaptationSet mimeType="video/mp4" contentType="video"
segmentAlignment="true" startWithSAP="1">
            <Representation id="1" width="320" height="240"
codecs="avc1.640828" bandwidth="250000">
              <BaseURL>../bitrate1/video/</BaseURL>
              <SegmentTemplate initialization="Init.mp4"
media="Segment$Number$.m4s" startNumber="0" timescale="1000">
                <SegmentTimeline>
                .....
              </SegmentTimeline>
            </Representation>
            <Representation id="2" width="640" height="480"
bandwidth="500000">
              <BaseURL>../bitrate2/video/</BaseURL>
            .....
            </Representation>
       </AdaptationSet>

       <AdaptationSet contentType="audio" id="2" segmentAlignment="1"
startWithSAP="1">
           <Representation bandwidth="64000" codecs="mp4a.40.02" id="1"
mimeType="audio/mp4">
              <BaseURL>../bitrate1/audio/</BaseURL>
                 <SegmentTemplate initialization="Init.mp4"
```

```
media="Segment$Number$.m4s" startNumber="0" timescale="1000">
                    <SegmentTimeline>
                    .....
                    </SegmentTimeline>
                </SegmentTemplate>
            </Representation>
        </AdaptationSet>
```

```
       </Period>
</MPD>
```

Example of MPEG-DASH MPD with SCTE-35 Mode Cues

Example shows a snapshot of a live stream. The first cue, ID "1" signals a program start with no duration available. Cue "2" signals the beginning of an ad break. The break is terminated with an end market, cue "3"

Note that the actual base64 binary data is sample data and does not correctly reflect the values in the in-manifest cues.

```
<?xml version="1.0"?>
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="urn:mpeg:dash:schema:mpd:2011"
    xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd"
    type="dynamic"
    availabilityStartTime="2014-03-23T11:51:44"
    minimumUpdatePeriod="PT2S"
    timeShiftBufferDepth="PT600S"
    minBufferTime="PT4S"
    profiles="urn:mpeg:dash:profile:isoff-live:2011"
    >

    <!-- Live stream containing single period w/ embedded SCTE-35 ad
signals as EventStream -->
    <Period start="PT0S">
        <EventStream schemeIdUri="urn:scte:scte35:2014:xml+bin"
timescale="1">
            <Event id="id:1" presentationTime="600">

<scte35:Binary>/DAIAAAAAAAAAAQAAZ/[I0VniQAQAgBDVUVJQAAAAH]\+cAAAAAA==</sc
te35:Binary>
   </Event>
            <Event id="id:2" presentationTime="614" duration="30">

<scte35:Binary>/DAIAAAAAAAAAAQAAZ/[I0VniQAQAgBDVUVJQAAAAH]\+cAAAAAA==</sc
te35:Binary>
   </Event>
            <Event id="id:3" presentationTime="644">

<scte35:Binary>/DAIAAAAAAAAAAQAAZ/[I0VniQAQAgBDVUVJQAAAAH]\+cAAAAAA==</sc
te35:Binary>
   </Event>
        </EventStream>

        <AdaptationSet mimeType="video/mp4" contentType="video"
segmentAlignment="true" startWithSAP="1">
            <Representation id="1" width="320" height="240"
codecs="avc1.640828" bandwidth="250000">
              <BaseURL>../bitrate1/video/</BaseURL>
              <SegmentTemplate initialization="Init.mp4"
```

```
media="Segment$Number$.m4s" startNumber="0" timescale="1000">
                <SegmentTimeline>
                .....
                </SegmentTimeline>
            </Representation>
            <Representation id="2" width="640" height="480"
bandwidth="500000">
              <BaseURL>../bitrate2/video/</BaseURL>
            .....
            </Representation>
       </AdaptationSet>

       <AdaptationSet contentType="audio" id="2" segmentAlignment="1"
startWithSAP="1">
           <Representation bandwidth="64000" codecs="mp4a.40.02" id="1"
mimeType="audio/mp4">
              <BaseURL>../bitrate1/audio/</BaseURL>
                 <SegmentTemplate initialization="Init.mp4"
media="Segment$Number$.m4s" startNumber="0" timescale="1000">
                     <SegmentTimeline>
                     .....
                     </SegmentTimeline>
                 </SegmentTemplate>
           </Representation>
       </AdaptationSet>
```

```
        </Period>
</MPD>
```