SCTE | STANDARDS

Digital Video Subcommittee

AMERICAN NATIONAL STANDARD

ANSI/SCTE 214-1 2022

MPEG DASH for IP-Based Cable Services
Part 1: MPD Constraints and Extensions

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.

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 214-1 2015	*8/10/2015*
SCTE 214-1 2016	*2/22/2016*

Table of Contents

Title **Page Number**

NOTICE 2
Document Types and Tags 3
Document Release History 3
Table of Contents 4
1. Introduction 7
1.1. Executive Summary 7
1.2. Scope 7
1.3. Benefits 7
1.4. Intended Audience 7
1.5. Areas for Further Investigation or to be Added in Future Versions 7
2. Normative References 7
2.1. SCTE References 8
2.2. Standards from other Organizations 8
3. Informative References 10
3.1. SCTE References 10
3.2. Standards from other Organizations 10
3.3. Published Materials 11
4. Compliance Notation 11
5. Abbreviations and Definitions 11
5.1. Abbreviations 11
5.2. Notation 12
6. MPD Restrictions 13
6.1. Restrictions on MPD elements 13
6.2. Restrictions on Period elements 13
6.3. Restrictions on Adaptation Set elements 14
6.4. Restrictions on ContentComponent elements 18
6.5. Restrictions on Representation elements 18
6.6. Restrictions on use of XLink 19
6.7. Use of events 19
6.7.1. Inband events 19
6.7.2. MPD Events 20
6.7.3. Carriage of SCTE 35 as inband event messages 22
6.8. MPD Updates 23
6.9. MPD handling of missing segments 24
7. Signaling service metadata 24
7.1. Associated services 24
7.1.1. General 24
7.1.2. Media language defaults 24
7.1.3. Roles 25
7.1.4. Full and partial audio services 27
7.2. Caption service metadata in video media components 28
7.2.1. Introduction 28
7.2.2. Signaling CEA-708 caption service metadata 28
7.2.3. Signaling CEA-608 caption service metadata 29
7.2.4. Caption service examples 30
7.2.5. Derivation of caption service metadata from MPEG-2 TS 30
8. Signaling Asset Identification 30
8.1. General 30
8.2. UPID Content Identification scheme 31
8.2.1. ContentIdentifier element semantics 31

8.2.2. XML syntax 32
8.2.3. Example 32
8.3. Use of EIDR and Ad-ID 32
9. Generic restrictions on media segments 33
9.1. Terminology 33
9.2. Duration 33
9.2.1. Segments 33
9.2.2. Subsegments 34
9.2.3. Segment duration patterns (Deprecated) 34
9.3. Bandwidth, size, and buffering 36
9.3.1. Introduction 36
9.3.2. Segments 36
9.3.3. Video aspects 37
10. Codec-Specific Aspects 37
10.1. Video 37
10.1.1. Supported video codecs 37
10.1.2. Resolutions and frame rates 37
10.1.3. SAP values 37
10.1.4. Multiplexed segments 37
10.1.5. Colorimetry 38
10.1.6. Supplemental Codecs 38
10.2. Audio 39
10.2.1. Supported codecs 39
10.2.2. SAP values 39
10.3. Trick Modes 40
10.3.1. Introduction 40
10.3.2. Trick mode representations 40
10.3.3. I-Frame track representations 40
11. Multi-period assets 41
11.1. Period continuity 41
11.2. Asset boundaries 41
11.3. Stream restrictions 42
12. URL query parameters for XLink 43
12.1. Introduction 43
12.2. Notation 43
12.3. Carriage of SCTE 35 in a query parameter 43
12.4. Carriage of geographical information in a query parameter 45
13. Annotation 46
13.1. Generator identification 46
14. Content Protection 48
14.1. Multi-key encryption 48
Annex A URNs 49

List of Figures

Title **Page Number**

FIGURE 1 - HANDLING OF SCTE 35 INFORMATION 20
FIGURE 2 - INBAND CARRIAGE OF SCTE 35 CUE MESSAGE 23

List of Tables

Title	Page Number
TABLE 1 - AUDIO SERVICES	26
TABLE 2 - ACCESSIBILITY ASSOCIATED SERVICES	26
TABLE 3 - CAPTION SERVICE EXAMPLES	30
TABLE 4 - UPID CONTENTIDENTIFIER ELEMENT SEMANTICS	31
TABLE 5 - UPID CONTENT IDENTIFICATION XML SYNTAX	32
TABLE 6 - UPID CONTENT IDENTIFICATION EXAMPLE	32
TABLE 7 - COLORIMETRY	38
TABLE 8 - SUPPLEMENTAL XML ATTRRIBUTES	38
TABLE 9 - CARRIAGE OF SCTE 35 IN A QUERY PARAMETER	43
TABLE 10 - REMOTE PERIOD ELEMENT WITH SCTE 35 EMBEDDED IN XLINK URL	44
TABLE 11 - CARRIAGE OF GEOGRAPHICAL INFORMATION IN A QUERY PARAMETER	45
TABLE 12 - GENERATOR IDENTIFICATION	46
TABLE 13 - REMOTE ENTITY WITH GENERATOR INFORMATION	47

1. Introduction

1.1. Executive Summary

This document is part of a suite of documents related to MPEG DASH for IP-Based cable services and is referred to SCTE 214. Part 1 describes general MPD constraints and common features supported by both the DASH TS profile and DASH ISO-BMFF profile. Part 2 contains further constraints for the DASH TS profile. Part 3 (*deprecated*) contains further constraints for the DASH ISOBMFF profile. Part 4 provides an instance MPD template for the DASH TS profile. Part 5 which replaces Part 3 (*deprecated*) extends the features and constraints but only for ISO-BMFF DASH Profiles.

1.2. Scope

This document describes general media presentation description (MPD) constraints and common features supported by both the DASH TS profile and DASH ISOBMFF profile. This will allow a common feature parity between DASH Profile and ISOBMFF Profile versions of the service and includes multiplexed segments. SCTE 214-1 and [SCTE 214-2] are used together to support DASH TS Profile delivery which is beneficial while transitioning from traditional broadcast MPEG-2 TS delivery structures using an ATS structured stream. Additional features developed in later DASH editions and needed for CABLE IP Services will be supported in SCTE 214-5 but only for constrained DASH ISO-BMFF Profiles with non-multiplexed segments.

Profile URNs for DASH/TS and DASH/FF appear in SCTE 214-2 and SCTE 214-5.

1.3. Benefits

This document assists in the transition between a broadcast cable infrastructure to a unicast IP based cable services while allowing for a gradual transition of backend systems supporting these services.

1.4. Intended Audience

This document is intended for the development, operations, and maintenance of IP based cable services. It is useful for those creating, ingesting, and delivering content for adaptive streaming services.

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

Additional developments in this document are limited to including additional features that can co-exist with features and constructs define in DASH edition 5 that enable parity between DASH MPEG-TS and DASH ISOBMFF profiles which includes the use of multiplexed segments.

2. Normative References

The following documents contain provisions which, through reference in this text, constitute provisions of this document. The editions indicated were valid at the time of subcommittee approval. 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

[SCTE 35] ANSI/SCTE 35 2020, Digital Program Insertion Cueing Message for Cable

[SCTE 128-1] ANSI/SCTE 128-1 2020, AVC Video Constraints for Cable Television Part 1: Coding

[SCTE 130-10] ANSI/SCTE 130-10 2020, Digital Program Insertion – Advertising Systems Interfaces, Part 10 – Stream Restriction Data Model (SRDM)

[SCTE 193-1] ANSI/SCTE 193-1 2020, MPEG-4 AAC Family Audio System – Part 1: Coding Constraints for Cable Television

[SCTE 193-2] ANSI/SCTE 193-2 2020, MPEG-4 AAC Family Audio System – Part 2: Constraints for Carriage over MPEG-2 Transport

[SCTE 194-1] ANSI/SCTE 194-1 2018, DTS-HD Audio System – Part 1: Coding Constraints for Cable Television

[SCTE 194-2] ANSI/SCTE 194-2 2018, DTS-HD Audio System – Part 2: Constraints for Carriage over MPEG-2 Transport

[SCTE 215-1] ANSI/SCTE 215-1 2020, HEVC Video Constraints for Cable Television, Part 1 – Coding

[SCTE 215-1-1] ANSI/ SCTE 215-1-1 2020b, HEVC Video Constraints for Cable Television, Part 1- 1 HDR10 Coding

[SCTE 223] ANSI/SCTE 223 2018, Adaptive Transport Stream

2.2. Standards from other Organizations

[ATSC A/52] ATSC A/52: 2018 Digital Audio Compression (AC-3) (E-AC-3) Standard

[ATSC A/53] ATSC A/53 Part 5: 2014 ATSC Digital Television Standard

[ATSC A/65] ATSC A/65:2013 ATSC Standard: Program and System Information Protocol for Terrestrial Broadcast and Cable

[DASH] ISO/IEC 23009-1:2021 5th Ed., Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats.

[CMAF] ISO/IEC 23000-19: 2020, Common media application format (CMAF) for segmented media, AMENDMENT 1: Additional CMAF HEVC media profiles

[H.264/VC] ITU-T Recommendation H.264: 2020 "Advanced video coding for generic audio-visual services" | ISO/IEC 14496-10: "Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding".

[ISOBMFF] ISO/IEC 14496-12: 2020, Information technology – Coding of audio-visual objects – Part 12: ISO base media file format.

[ISOBMFF-NAL] ISO/IEC 14496-15: 2019, Information technology – Coding of audio-visual objects – Part 15: Carriage of network abstraction layer (NAL) unit structured video in ISO base media file format.

[HEVC] ITU-T Recommendation H.265: 2021, "Advanced video coding for generic audio-visual services" | ISO/IEC 23008-2: " High Efficiency Coding and Media Delivery in Heterogeneous Environments – Part 2: High Efficiency Video Coding"

[CICP-Video] ISO/IEC 23091-2: 2019, “Information technology – MPEG systems technologies – Part 8: Coding-independent code points”

[CEA 608-E] ANSI/CTA-608-E S-2019: Line 21 Data Services. Note that for backwards compatibility this standard is referred to as a CEA document even though the official name is now CTA.

[CEA 708-E] ANSI/CTA-708-E R-2018: Digital Television (DTV) Closed Captioning. Note that for backwards compatibility this standard is referred to as a CEA document even though the official name is now CTA.

[CEA 708.1-E] ANSI/CTA-708-1 R-2017: Digital Television (DTV) Closed Captioning: 3D Extensions. Note that for backwards compatibility this standard is referred to as a CEA document even though the official name is now CTA.

[ISO 639-3] ISO 639-3:2007, Codes for the representation of names of languages-Part3: Alpha-3 code for comprehensive coverage of languages – First Edition

[RFC 7826] IETF RFC 7826, Real Time Streaming Protocol (RTSP) Version 2.0, December 2016

[RFC 3339] IETF RFC 3339, Date and Time on the Internet: Timestamps, July 2002

[RFC 8141] IETF RFC 8141, Uniform Resource Names (URNs), April 2017

[RFC 6874] IETF RFC 6874, Representing IPV6 Zone Identifiers in Address Literals and Uniform Resource Identifiers, February 2013

[RFC 4648] IETF RFC 4648, The Base16, Base32, and Base64 Data Encodings, October 2006

[RFC 7405] IETF RFC 7405, Case-Sensitive String Support in ABNF, December 2014

[RFC 6381] IETF RFC 6381, The ‘Codecs’ and ‘Profiles’ Parameters for ‘Bucket’ Media Types, August 2011

[RFC 7230] IETF RFC 7230, Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing, June 2014

[RFC 7231] IETF RFC 7231, Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content, June 2014

[RFC 7232] IETF RFC 7232, Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests, June 2014

[RFC 7233] IETF RFC 7233, Hypertext Transfer Protocol (HTTP/1.1): Range Requests, June 2014

[RFC 7234] IETF RFC 7234, Hypertext Transfer Protocol (HTTP/1.1): Caching, June 2014

[RFC 5646] IETF RFC 5646, Tags for Identifying Languages, September 2009

[DASH-IF IOP] DASH-IF Implementation Guidelines: Interoperability Points; Version 4.3, 2018, http://dashif.org/wp-content/uploads/2015/10/DASH-IF-IOP-v4.3.pdf

[XML] Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Recommendation 26 November 2008, available at http://www.w3.org/TR/REC-xml/

[XLINK] XML Linking Language (XLink) Version 1.0, W3C Recommendation 27 June 2001, available at http://www.w3.org/TR/xlink/

[XML Schema] XML Schema Part 2: Datatypes Second Edition, W3C Recommendation 28 October 2004, available at http://www.w3.org/TR/xmlschema-2/

[ISOBMFF-TT] ISO/IEC 14496-30: 2018, Information technology- Coding of audio-visual objects- Part 30: Timed text and other visual overlays in ISO base media file format- Second Edition.

[IMSC 1] W3C TTML Profiles for Internet Media Subtitles and Captions 1.0 (IMSC1), 2020

3. Informative References

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

3.1. SCTE References

[SCTE 243-1] SCTE 243-1 Next Generation Audio Carriage Constraints for Cable Systems: Part-1 _Common Transport Signaling.

[SCTE 243-2] SCTE 243-2 Next Generation Audio Carriage Constraints for Cable Systems: Part-2 _AC-4 Audio Carriage Constraints.

[SCTE 243-3] SCTE 243-3 Next Generation Audio Carriage Constraints for Cable Systems: Part-3 _MPEG-H Audio Carriage Constraints. SCTE 243-4 Next Generation Audio Carriage Constraints for Cable Systems: Part-4 DTS-UHD Audio Carriage constraints.

3.2. Standards from other Organizations

[ATSC T/300] ATSC Technology Group Report: ATSC 3.0 Launch – DASH Timeline and IMSC1.

[ETSI TS 103 285] ETSI TS 103 285 V1.3.1 (2020-02): "MPEG-DASH Profile for Transport of ISO BMFF Based DVB Services over IP Based Networks"

[NorDig] NorDig Unified Requirements for Integrated Receiver Decoders for use in cable, satellite, terrestrial and managed IPTV based networks, NorDig.

3.3. Published Materials

[HLS I-D] IETF RFC 8216bis, HTTP Live Streaming.

4. Compliance Notation

shall	This word or the adjective "***required***" means that the item is an absolute requirement of this specification.
shall not	This phrase means that the item is an absolute prohibition of this specification.
forbidden	This word means the value specified shall never be used.
should	This word or the adjective "*recommended*" 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.
should not	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.
may	This word or the adjective "*optional*" 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.
deprecated	Use is permissible for legacy purposes only. Deprecated features may be removed from future versions of the standard. Implementations should avoid use of deprecated features.

5. Abbreviations and Definitions

5.1. Abbreviations

AAC	advanced audio coding
ABNF	Augmented Backnus-Naur Form
AC-3	Audio Codec 3 or Advanced Codec 3 (also Dolby Digital)
AES-CBC	Advanced Encryption Standard cipher block chaining
ANSI	American National Standards Institute
ATSC	Advanced Television Systems Committee
AVC	advanced video coding
BMFF	base media file format
CC	closed captioning
CEA	Consumer Electronics Association, former name of the Consumer Technology Association
CPB	coded picture buffer
CTA	Consumer Technology Association
DASH	[MPEG] dynamic adaptive streaming over HTTP
DTS	trademark for DTS, Inc. audio (originally Digital Theater Systems, Inc.)
DTV	digital television
DVB	Digital Video Broadcasting [Project]
DVS	[SCTE] Digital Video Subcommittee

e.g.	for example (*exempli gratia*)
ETSI	European Telecommunications Standards Institute
FF	file format
HEVC	high efficiency video coding
HI	hearing impaired
HLS	HTTP live streaming
HTTP	hypertext transfer protocol
i.e.	that is (*id est*)
IEC	International Electrotechnical Commission
IP	Internet protocol
ISO	International Organization for Standardization
ISO-BMFF	ISO Base Media File Format
MBT	minimum buffer time
MPD	media presentation description
MPEG	Moving Picture Experts Group
MPEG-2 TS	MPEG-2 transport stream
NAL	network abstraction layer
PMT	program map table
SAP	stream access point(s)
SCTE	Society of Cable Telecommunications Engineers
SEI	supplemental enhancement information
TS	MPEG-2 transport stream
UPID	unique program identifier
URI	uniform resource identifier
URN	universal resource name
VI	visually impaired
XLink	external link
XML	extensible markup language

5.2. Notation

This document uses notation similar to the one of ISO/IEC 23009-1.

Extensible markup language (XML) elements are written in bold face, e.g. **`Element1`**.

Child XML elements are separated from parent elements by a dot ('.'), e.g. **`Element2.Element1.`**

XML attributes are prefixed by an at-sign ('@'), e.g. `@attribute`. Attributes of an element are separated from the name of the containing element by at-sign, e.g. **`Element`**`@attribute.`

ISO-BMFF boxes are written as box names enclosed in backquote ('`') signs, e.g. `` `box0` ``

Fields in ISO-BMFF boxes are separated from box names by a dot ('.'), e.g. `` `box0`.field0 ``

In cases where an element has the same name as a concept it describes, when the name is written in bold face, it refers to the syntactic element. For example, **`Representation`** refers to an XML element

named "Representation", while "representation" refers to the concept of a representation as defined in ISO/IEC 23009-1.

XML elements and attributes defined in SCTE 214 are prefixed with `scte214:`

URLs that do not fit into a single line are escaped with `\` character.

6. MPD Restrictions

6.1. Restrictions on MPD elements

1. **`MPD`**`@minBufferTime` ***shall*** be present. Its value *should* be equal or larger than maximum segment (live profile) or subsegment (on demand profile) duration.

2. If the **`MPD`**`@type` is `"dynamic"`:

    a. **`MPD`**`@minimumUpdatePeriod` ***shall*** be present.

    b. **`MPD`**`@maxSegmentDuration` ***shall*** be present.

Note: It is unsafe to base player buffer allocation on the attributes above whenever XLink is used, or MPD type is `"dynamic"`. See sec. 8.2.3 for more details.

    c. **`MPD`**`@availabilityStartTime` ***shall*** be present and *should* satisfy the conditions below:

        i. **`MPD`**`@availabilityStartTime` *should* not change across MPD updates

        ii. **`MPD`**`@availabilityStartTime` *should* be set to a fixed time in the past (e.g. the Epoch time)

    d. **`MPD`**`@suggestedPresentationDelay` *should* be present.

    e. If it is expected that at some point in the future a media segment *may* become unavailable, then the `@timeShiftBufferDepth` attribute ***shall*** be present, or `@timeShiftBufferDepth` ***shall*** be present as a part of segment information.

    f. **`MPD.Location`** element *should* be present

    g. **`MPD.PatchLocation`** *may* be present, and **`MPD.Location`** ***shall*** be present in this case. The patch URL *should* contain the value of **`MPD`**`@publishTime` of the original MPD (verbatim) as a query parameter named `publishTime`.

6.2. Restrictions on Period elements

1. The **`Subset`** element ***shall not*** be present.

2. The **`Period.SegmentList`** element ***shall not*** be present

3. At least one audio and one video **AdaptationSet** element ***shall*** contain a **Role** element with `@schemeIdUri="urn:mpeg:dash:role:2011"` and `@value="main"` Perceptually equivalent audio adaptation sets differing by non-linguistic characteristics such as codecs or number of channels *should* be prioritized for the players using the `@selectionPriority` attribute across the equivalent adaptation sets with a higher value indicating higher preference. The **Role** descriptor ***shall*** be used only to express linguistic differences.

4. If a **Period** element represents a part of a multi-period asset, this **Period** *should* contain an **AssetIdentifier** element.

5. **Period**`@start` ***shall*** be present

6. **Period**`@id` ***shall*** be present

6.3. Restrictions on Adaptation Set elements

1. Every adaptation set ***shall*** use consistent addressing. Exactly one of the following restrictions ***shall*** be met:

   a. *Deprecated*: Every Representation within this Adaptation Set has a **Representation.SegmentTemplate** element, and **AdaptationSet.SegmentTemplate** is not present.

   b. **AdaptationSet.SegmentTemplate** element is present, and no **Representation.SegmentTemplate** elements are present.

   c. *Deprecated*: The **Representation.SegmentList** element is present in every representation in this **AdaptationSet**, and all media segments are MPEG-2 TS segments.

   d. Every **Representation** element within this Adaptation Set consists of a single segment and its URL is stated in the **Representation.BaseURL** element. The MPEG DASH ISO-BMFF On Demand profiles satisfy this requirement.

2. All Representations within an Adaptation Set ***shall*** use the same codecs, but not necessarily the same profiles and levels. Therefore, exactly one of the following restrictions ***shall*** be met:

   a. The **AdaptationSet**`@codecs` attribute is present and signals the maximum profile, tier, and level of any representation contained in the adaptation set, and **Representation** elements within this **AdaptationSet** *may* contain the full `@codecs` attribute including profile, level, or tier pertaining to that representation if these differ from the one signaled in **AdaptationSet**`@codecs`.

      Note: As an example, no adaptation set is permitted contain both `` `avc1` `` (AVC) and `` `hev1` `` (HEVC) video, however `avc1.64Y01F` (Progressive High@L3.1) and `avc1.64Y028` (Progressive High@L4.0) can be in the same AVC adaptation set.

b. **AdaptationSet**@codecs is present as code point (4cc) only, and **Representation** elements within this **AdaptationSet** contain the full @codecs attribute including profile, level, or tier pertaining to that representation.

3. **AdaptationSet**@segmentAlignment attribute ***shall*** be present and have a value of `true`.

Note: Older versions of this specification allowed value of 1, which has been *deprecated* in the latest editions of ISO/IEC 23009-1

4. **AdaptationSet**@startWithSAP ***shall*** be present and its value ***shall*** be 1 or 2.

5. If indexing (`sidx`) is used, then

   a. @subsegmentAlignment ***shall*** be present and have value of `true`.

   b. **AdaptationSet**@subsegmentStartsWithSAP ***shall*** be present and ***shall*** have a value of `1` or `2`. Values larger than 2 are allowed if **AdaptationSet**@segmentAlignment is also present.

      Note: The above *may* happen in cases such as low-latency deployment relying on CMAF chunks (subsegments)

6. For any adaptation set that contains video the following attributes ***shall*** be present:

   a. @maxWidth (or @width if all Representations have the same width)

   b. @maxHeight (or @height if all Representations have the same height)

   c. @maxFrameRate *should* be an integer multiple of each @frameRate in this adaptation set. If all representations have the same frame rate, @frameRate rather than @maxFrameRate *should* be present

   d. @scanType *should not* be present for progressive content.

   e. If one or more of the video frames are coded as interlaced. @scanType ***shall*** be present and have value "interlaced"

      Note: The above implies that if an adaptation set is interlaced, all representations in it are interlaced. There is no need to signal progressive content – this is the default value of the @scanType attribute when not present.

   f. @sar ***shall*** be present and, consequently, all representations within this adaptation set ***shall*** have the same aspect ratio.

   g. @lang attribute ***shall*** be used with video sign language media components to indicate which type of sign language is used. These code points are defined in ISO 639-3

   h. If a representation contains at least one interlaced picture, this representation is considered interlaced. Interlaced and non-interlaced representations ***shall not*** be mixed in the same adaptation set.

Note: Alignment between interlaced and non-interlaced adaptation sets can be expressed by using adaptation set switching

i. For any adaptation set, colorimetric properties of video representations, ***shall*** be the same. When these properties are known (e.g. can be derived from the VUI as defined in AVC, HEVC, and VVC), they *should* be signaled as defined in sec 10.1.5.

7. *Deprecated*: There ***shall*** be at most one video media component in a single **`AdaptationSet`**.

8. Supplementary video media components (e.g. video sign language) ***shall*** be in separate adaptation sets. They ***shall*** have the **`Role`** descriptor with value other than "main".

9. For any adaptation set containing audio, the following elements and attributes ***shall*** be present (and thus ***shall*** apply to all representations):

    i. `@lang` ***shall*** be present. ISO 639-3 (3-character code) ***shall*** be used in the `@lang` attribute. If the information expressed in the code is insufficient to determine the linguistic content, BCP-47 extensions ***shall*** be used for the purpose [RFC 5646].

    Note 1: the above signaling complies with both BCP-47 and `xs:lang` syntax and restricts the number of possible options for encoding a single language

    Note 2: BCP-47 extensions *may* be needed in cases such as regional dialects.

    ii. When language can change and timing of such a change is unknown, the `` `mul` `` language code ***shall*** be used.

    iii. When no language is used, then `` `zxx` `` language code *should* be used.

    iv. When the language has no code, then `` `mis` `` language code *should* be used.

    b. `@codecs`, which contains sub-parameters as defined in RFC 6381. **Note** implies that only the option described in 2.b above is acceptable for audio adaptation sets.

    c. `@audioSamplingRate`

    d. `AudioChannelConfiguration` which ***shall*** be at **`AdaptationSet`** level. This implies that all audio representations within an adaptation set shall have same channel configuration.

    e. There ***shall*** be same number of audio and video segments

    f. Audio and video segment duration ***shall*** be very close and the earliest presentation time of any i^{th} audio and i^{th} video segment *should* be within 70ms from each other

    Note: when **`SegmentTimeline`** and `$Time$`-based addressing are used for video, the above makes it possible to use **`SegmentTemplate`** with `$Number$`−based addressing and **`SegmentTimeline`** for audio. This *may* significantly reduce the overall MPD size.

10. If media segments contain CEA 608/708 closed captioning carried in video elementary stream (as defined in SCTE 128-1 and SCTE 215-1), this ***shall*** be reflected in the MPD using **`AdaptationSet.Accessibility`**, as described in 7.2.

11. For any adaptation set carrying text (such as closed captioning or subtitles) the following ***shall*** hold:

    a. IMSC1 text profile of TTML ***shall*** be used;

    b. The above provide a continuous timeline and avoid the creation of sparse tracks by using empty segments to create a continuous set of segments. Please refer to [ATSC T/300] for recommended practices.

    c. Text adaptation sets *should* be aligned to video adaptation sets, and the earliest presentation time of any i^{th} text and i^{th} video segment *should* be within 100ms from each other

    d. `@lang` ***shall*** be present. ISO 639-3 (3-character code) ***shall*** be used in the `@lang` attribute. If the information expressed in the code is insufficient to determine the linguistic content, BCP-47 extensions ***shall*** be used for the purpose.

       Note 1: the above signaling complies with both BCP-47 and `xs:lang` syntax and restricts the number of possible options for encoding a single language

       Note 2: BCP-47 extensions *may* be needed in cases such as regional dialects or different scripts used for the same language.

12. If segments comply with one or more CMAF profiles defined in ISO/IEC 23000-19 or in CTA-5001, these ***shall*** be signaled in the **`AdaptationSet`**`@segmentProfiles` attribute using the ISO-BMFF brands defined in the above standards

13. Segments *should* be referenced from one and only one **`SegmentTemplate`** element.

14. Timescale values for ISO-BMFF representations ***shall be*** selected such that any sample duration would be an integer. **Note** that the MPEG-2 Systems 90KHz timescale does not satisfy this requirement in cases of video frame rates such as 24000/1001 or 60000/1001 fps. One of the two practices below are recommended:

    a. Use different timescales in different adaptation sets. In particular, use frame rate (video) or sampling rate (audio) as timescale; For example, timescale of 24 can be used for 24fps content, while value of 60000 can be used for 60000/1001 (i.e., each sample duration is 1001).

    b. Use a common timescale across adaptation sets, allowing for integer sample duration for all adaptation sets. In particular, 240KHz timescale satisfies these requirements in most cases

15. **`AdaptationSet`** attributes for Supplemental Media Properties such as codecs ***shall*** be used for enhanced media experiences such as SCTE 215-1-1. Supplemental media properties are not essential properties of the content but can be used to enhance the media experience upon playout but are not essential to the playout of the content. An scte214 namespace ***shall*** be used to create additional **`AdaptationSet`** attributes for this purpose.

    Note: By signalling enhanced viewing properties in the form of **`AdaptationSet`** attributes can be used to avoid separate **`AdaptationSet`** elements pointing to the same content segments.

    Supplemental media properties defined for these purposes are:

    a. **`AdaptationSet`**`@scte214:supplementalProfiles`

Values for this list are ISO-BMFF optional brands (registered in mp4ra.org) with whitespace separated multiple values being listed in no particular order.

b. **AdaptationSet**@scte214:supplementalCodecs

Values for this list of codec strings for "enhancement" codecs with whitespace-separated multiple codec strings being listed in no particular order

6.4. Restrictions on ContentComponent elements

ContentComponent element ***shall not*** be used with MPEG-2 TS segments, i.e., unless the **AdaptationSet**@mimeType equals video/mp2t, audio/mp2t, or application/mp2t.

For MPEG-2 TS segments, the below conditions apply:

1. **AdaptationSet** elements ***shall*** contain a single **ContentComponent** element per each media component in a multiplexed representation.
2. If more than one audio content component is present, each one of them ***shall*** be signaled using a separate **ContentComponent** element. @lang attribute ***shall*** be present for each audio component.
3. **ContentComponent**@contentType attribute ***shall*** be present in any **ContentComponent** element.

6.5. Restrictions on Representation elements

1. The following attributes and elements ***shall not*** appear at **Representation** level within an adaptation set containing audio:

    a. **AudioChannelConfiguration**;

    b. @audioSamplingRate;

    c. @lang;

    d. @codecs

Note: the above implies that audio representations within an adaptation set always have identical codec, sampling rate and number of channels.

2. For any **Representation** element within an adaptation set containing video the following attributes ***shall*** be present:

    a. @width, if and only if **AdaptationSet**@width is not present in this adaptation set

    b. @height, if and only if **AdaptationSet**@height is not present in this adaptation set

    c. @frameRate, if and only if **AdaptationSet**@frameRate is not present in this adaptation set

    d. @codecs, which ***shall*** contain complete sub-parameter string as defined in ISO/IEC 14496-15 Annex E. **Note** that this means that 2.a in 6.3 applies to video adaptation sets.

3. **Representation**@id value ***shall*** be unique within the scope of the **Period** to which it belongs.

4. **Representation**@bandwidth value ***shall*** be unique within its parent **AdaptationSet** element.

5. **Representation.ContentProtection** element ***shall not*** be used.

6. **Representation.SegmentList** element ***shall not*** be used.

7. In case of variable bitrate content, **Representation.ExtendedBandwidth** element *should* be used to express average bandwidth.

    a. **ExtendedBandwidth**@vbr ***shall*** be true

    b. For assets conforming to the live profile, **ModelPair**@bufferTime *may* be infinite if the model describes target bitrate as set in the encoder and not all segments are available. It is recommended to use the value P999D for the purpose, for compatibility purposes, as infinity cannot be properly expressed with xsd:duration. If all segments are available, **ModelPair**@bufferTime ***shall*** be set to the total duration of the period.

    Note: For assets conforming to the VOD profile, all bitrate information can be trivially derived from the `sidx` box, so extended signaling is redundant. With that said, there *may* be operational circumstances where its use is needed.

6.6. Restrictions on use of XLink

The use of the XLink (only the subset defined in ISO/IEC 23009-1) is supported in SCTE profiles with the following restrictions:

1. The @xlink:href attribute *may* appear only in **Period** elements;

2. If the **Period**@xlink:href attribute is present, the value of **Period**@xlink:actuate ***shall*** be 'onLoad'

6.7. Use of events

6.7.1. Inband events

1. Event message boxes with same event scheme ***shall*** be aligned, per definition of event alignment in ISO/IEC 23009-1.
2. Version 0 of the EventMessageBox (`emsg`) ***shall not*** be used.

6.7.1.1. Declaring events

1. **InbandEventStream** element ***shall not*** be present either at **Representation** or at **SubRepresentation** level.

2. If inband events are used, their presence ***shall*** be signaled in **AdaptationSet.InbandEventStream** element. A client cannot be expected to process undeclared events, though this specification does not disallow processing them.

6.7.1.2. DASH events

MPD Patch and **MPD** Update events ***shall not*** be used, either inband or in **MPD**.

6.7.2. MPD Events

6.7.2.1. Carriage of SCTE 35 as MPD events

SCTE 35 cue messages can be encapsulated into a DASH **Event** element hereby called in this document as 'SCTE 35 DASH Event'**.** SCTE 35 DASH Events can be published in an **MPD** when SCTE 35 messages in the stream are ingested and processed by systems responsible for **MPD** creation. SCTE 35 DASH Events are contained in one or more **EventStream** elements which are hereby called in this document as 'SCTE 35 DASH Event Stream'.

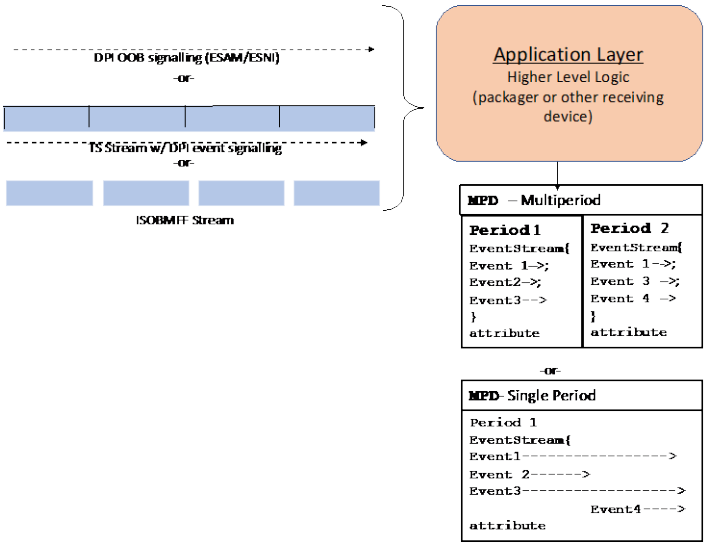

Figure 1 - Handling of SCTE 35 information

SCTE 35 DASH Events ***shall*** operate in an "on-receive" dispatch mode. Events will be dispatched at or before the event time to the Application [as defined in DASH 5th edition and DASH-IF 5.0]. This implies that the message will be passed to the application upon arrival with devices subscribing to the DASH Events.

SCTE 35 DASH Event elements contained in an SCTE 35 DASH Event Stream element with `@schemeIdUri="urn:scte:scte35:2013:xml"` ***shall*** contain an XML representation of an SCTE 35 cue message as either a **Signal.SpliceInfoSection** element or a **Signal.Binary** element as defined in SCTE 35.

SCTE 35 DASH Event elements contained in an SCTE 35 DASH Event Stream element with `@schemeIdUri ="urn:scte:scte35:2014:xml+bin"` ***shall*** contain an XML representation of an SCTE 35 cue message a **Signal.Binary** element as defined in SCTE 35.

For both `"urn:scte:scte35:2013:xml"` and `="urn:scte:scte35:2014:xml+bin"` schemes, SCTE 35 DASH Event Stream elements ***shall*** satisfy the following constraints,

1. If more than one SCTE 35 DASH Event Stream element with identical value of **`EventStream`**`@schemeIdUri` is present, then the **`EventStream`**`@value` attribute ***shall*** be present and *should* be set to the decimal PID value of the elementary stream or to a URI.

2. **`EventStream`**`@timescale` is an integer value for a common timescale that is granular enough to accommodate or allow at least all frame-accurate access and possibly frame-accurate access to audio. To achieve the former video timescale can be used, to achieve the latter a common time scale is needed. A timescale of 240000 is known to work with typical video and audio configurations used in US cable systems, but this standard does not require its use.

3. **`EventStream`**`@presentationTimeOffset` *should* be present. When **`EventStream`**`@presentationTimeOffset` is present and its value *should* ***be*** set such that **`Event`**`@presentationTime` of an event starting before the start of the current period would still be a non-negative number.

   Note 1: the above allows expressing events happening in the past without keeping periods for this purpose. As a result, the receiver *may* receive the same SCTE 35 DASH Event element twice, in different periods, and is expected to handle this.

   Note 2: It is possible to do "epoch locking" and start the event timing from the POSIX Epoch time. This implies that **`EventStream`**`@presentationTimeOffset` set to the number of ticks since the Epoch till the start of the period, and **`EventStream`**`@presentationTime` is set to the number of ticks from the Epoch till the event start as calculated from the SCTE 35 cue message. Note that the condition **`EventStream`**`@presentationTimeOffset > ` **`Event`**`@presentationTime` is legal and indicates an event starting prior to the start of the current period.

4. SCTE 35 DASH Event are considered essential, per definition of event essentiality in ISO/IEC 23009-1 [DASH].
   Note: SCTE 35 DASH Event*s* ***shall*** be received immediately and sent to the application.

For both `"urn:scte:scte35:2013:xml"` and `="urn:scte:scte35:2014:xml+bin"` schemes, SCTE 35 DASH Event elements ***shall*** satisfy the following constraints,

1. Each SCTE 35 DASH Event carrying an SCTE 35 Event shall contain only one SCTE 35 SpliceInfoSection. The SCTE 35 DASH Event shall include a reference to either the Splice Insert splice_event_id or Segmentation Descriptor segmenation_event_id that is applicable to the SCTE 35 DASH Event . See SCTE 35 on the specific constraints on which descriptors are applicable to each splice command.

2. **`Event`**`@duration` *should* be the expected duration of the SCTE cue message. If the event is "closed" by a different event that event *should* have a duration of zero and ***shall not*** be infinite.

   Note 1: Duration of zero implies that the application subscribed to SCTE 35 messages will get it immediately on its reception. If there is an expectation for a DASH client joining after the event start (e.g. during an ad break) to still reliably receive the event **`Event`**`@duration` *should* be sufficient for any such clients to join during its duration. For example, a duration can be set to an expected duration of the ad break

Note 2: An SCTE 35 cue message can be terminated by receiving a paired SCTE 35 event end message for the event, going beyond the assigned duration in an SCTE 35 event start message. Event extending beyond the period boundary will be ignored by clients starting from the next period unless copied into that period. Such copied event would have `presentationTime` smaller than **`EventStream`**`@presentationTimeOffset`

3. The **`Event`**`@messageData` attribute ***shall not*** be used.

4. SCTE 35 DASH Event elements *should* be retained as long as one or more media segments containing media with presentation times during the event are available.

5. There *should* not be more than one SCTE 35 DASH Event in the SCTE 35 DASH Event Stream with the same **`Event`**`@id` having identical values for the **`Event`**`@presentationTime` attribute. Note that subsequent events with same **`Event`**`@id` value *may* be dropped by the DASH client.

6.7.3. Carriage of SCTE 35 as inband event messages

1. SCTE 35 events ***shall*** be used with emsg v1 and use scheme `"urn:scte:scte35:2013:bin".`

2. For events with the above scheme, entire SCTE 35 `splice_info_section` starting at the `table_id` and ending with the `CRC_32` ***shall*** be carried in `` `emsg`.message_bytes[]. ``

3. `` `emsg`.presentation_time `` ***shall*** equal the time distance between the splice time and *PeriodStart* in the time units specified in `` `emsg`.timescale ``. These units ***shall*** be same as the timescale of the segment carrying the `` `emsg` `` box.

Note: epoch locking allows expressing events in the past. Since both `` `emsg` `` and the media segments carrying it share the value of `@presentationTimeOffset`, epoch locking will be needed for the media segments as well.

4. `` `emsg`.event_duration `` ***shall*** be a translation of `segmentation_duration` or `break_duration` into time units specified in `` `emsg`.timescale ``. If not known, the value ***shall*** be 0xFFFF or 0xFFFFFFFF (not known).

5. `` `emsg`.value `` ***shall*** be the value of either the SCTE 35 PID, or a URI, or may be left empty.

Note: `` `emsg`.value `` *may* be used by the application to distinguish between cue messages from different sources.

6. `` `emsg`.id `` ***shall*** be a unique integer within the scope of the Period (within the scope of events defined in the **`InbandEventStream`** element with same value of **`InbandEventStream`**`@value`). Simple ways of achieving it is using random numbers, CRC of the descriptor, or/and current time. Event messages with `` `emsg`.id `` values that already appeared *may* be discarded by the DASH client so id value *may* only be shared with repeated messages carrying no new information.

7. Presence of inband SCTE 35 events in media segments ***shall*** always be signaled using **`AdaptationSet.InbandEventStream`** element with `@schemeIdUri` value of `"urn:scte:scte35:2013:bin".` SCTE 35 events are essential as defined in SCTE 214-1 6.8.3

Note: As `emsg` boxes are aligned, all `emsg` boxes will be read if media from an adaptation set is played out irrespective of representation selections.

Figure 2 below shows the content of an `emsg` box at the beginning of a segment with earliest presentation time *T*. There is a 6-sec warning of an upcoming splice – delta to splice time is indicated as 6 seconds – and duration is given as 1 minute. This means that an ad will start playing at time *T* + *6* till *T* + *66*.

Figure 2 - Inband carriage of SCTE 35 cue message

6.8. MPD Updates

MPD updates *may* only extend the timeline, unless this is an MPD reset. This means that information provided in a previous version of the MPD *should not* be invalidated in an updated MPD. Hence the expected changes are typically addition or removal of **`Period`** elements, addition of **`Event`** elements, or addition or removal of **`S`** elements in **`SegmentTimeline`**.

In live scenarios **`MPD`** updates can add new **`MPD`** events but ***shall not*** remove existing **`MPD`** events in order to provide system consistency. Consequently, a cancellation of a previous event *should* be done via **`MPD`** update adding a new event. With that said, preserving events is not the reason to keep an old period when its segments are no longer available – relevant (e.g. “open”) SCTE 35 events *should* be copied to the current **`Period`**, and **`Period`** element removed via an **`MPD`** update.

Note: **`MPD`** Patches as defined in ISO/IEC 23009-1 5th edition are typically a more efficient mechanism then frequent **`MPD`** updates. Use of patches reduces the traffic and parsing overhead of **`MPD`** updates. The restrictions above apply to changes introduced either by a full **`MPD`** update or by a patch

6.9. MPD handling of missing segments

FailoverContent element *should* be used for missing segments at the time of **MPD** creation or updates. If missing segments are later found, then **MPD** update can remove the indicated segment from **FailoverContent** element at the next **MPD** update or through **MPD** patches.

7. Signaling service metadata

7.1. Associated services

7.1.1. General

In many cases an audio component is not intended for a general presentation, but for a more specialized purpose (e.g., audio description for the visually impaired). Moreover, in some cases (known as "receiver mix"), two audio elementary streams need to be combined for the same service.

This section defines signaling for such services. If signaling is present both in the media segments and in **MPD**, the two ***shall not*** contradict each other.

This section uses the **Role** descriptor for two different purposes. In 7.1.3 **Role** is used to express the purpose of the audio component, while in 7.1.47.1.4 the value of the **Role** descriptor indicates whether an audio component represents a full or partial service.

7.1.2. Media language defaults

In some cases, it is unclear which language *should* the player select by in absence of explicit user selection. One or more **SupplementalProperty** elements defined below can be used to explicitly signal this.

The value of the `@schemeIdURI` attribute ***shall*** be `urn:scte:dash:default-language<#fragment>` where the optional fragment argument can take the values "video", "audio", "text", and "commentary". If no fragment parameter is supplied, the language applies to all content components within the scope of the parent element.

The value of the `@value` attribute in these elements ***shall*** be a language as described in sec 6.3 above.

Example:

```
<SupplementalProperty schemeIdUri="urn:scte:dash:default-
language#audio" value="eng"/>

 <SupplementalProperty schemeIdUri="urn:scte:dash:default-
language#text" value="eng"/>

 <SupplementalProperty schemeIdUri="urn:scte:dash:default-
language#video" value="ase" />

<SupplementalProperty schemeIdUri="urn:scte:dash:default-
language#commentary" value="esp"/>
```

These supplemental properties represent the default language of the media or content for audio, text, video signing language, commentary from the channel or content perspective. The client shall use the specified language in absence of explicit user preferences. In cases where program language may shift from program to program but all programs are in its native language, the channel language of "qaa" can be used to indicate the channel language is native but can vary in the channel. See [NorDig] see Table 12.6.

In applications at the **`MPD`** level, it can represent the default language of the channel and can be used to align the language choices for a program, ad, or promo that starts at the new period boundary.

In applications at the Period level, it can represent an override of the **`MPD`** level default language information if available. An example of this may be a forced English commercial on a Spanish language channel.

7.1.3. Roles

Personalized audio services ***shall*** be signaled using the **`Role`** descriptor with `@schemeIdUri ="urn:mpeg:dash:role:2011"`. The role descriptor can differentiate multiple audio streams associated with a video media adaptation set or video content component. This allows the client player to select the appropriate audio adaptation set or content component for the video component adding some personalization factor to the content experience. It can assist with creating a smooth content experience with channels or programs that have inserted ads to align the audio experience of the ad with the main channel or program.

Associated services, such as visually impaired (VI) and hearing impaired (HI), ***shall*** be signaled using the **`Accessibility`** descriptor with `@schemeIdUri="urn:mpeg:dash:role:2011"` or **`Role`** descriptor with `@schemeIdUri="urn:scte:dash:associated-service:2015".`

> **Note**: the earlier versions of this specification required **`Role`**, rather than **`Accessibility`** descriptor. The 2021 edition of this specification requires use of **`Accessibility`** descriptor in order to achieve better alignment with DASH-IF and DVB approaches

7.1.3.1. Derivation from MPEG-2 TS

The role values *should* be derived from MPEG-2 TS stream using the ISO_639_ language descriptor in the program map table (PMT)[1]. In absence of ISO_639_ language descriptor, Role / Accessibility can be derived from the audio elementary streams as discussed below.

The recommended process for deriving Role/Accessibility for audio elementary stream is as follows. Let service_type *ST* be an integer value instantiated as follows:

- For AC-3 and E-AC-3 elementary streams, *ST* takes the value of the `bsmod` field. The possible values for `bsmod` are defined in ATSC A/52 Table 5.7
- For advanced audio coding (AAC) elementary streams, *ST* takes the value of `AAC_service_type`, as defined in SCTE 193-2 Table 4.

[1] Note that the ISO_639_language descriptor could contain more than one pair of language/audio_type values.

- For DTS elementary streams, *ST* value is derived from `component_type` bit values `b3`, `b4` and `b5`, as follows: *ST* = `b5 << 2 + b4 << 1 + b3`. The values `b3`, `b4` and `b5` are defined in SCTE 194-2 Table 6.

7.1.3.2. DASH signaling

The value of the **`Role`**`@value` attribute ***shall*** be derived from equivalent audio type value as described in table below.

Table 1 - Audio Services

Type	**`Role`**`@value`	**`Accessibility`**`@value`
Audio default (audio_type = 0x00 | service_type [bsmod or equivalent] = 000)	main[2]	N/A
Clean effects (audio_type = 0x01 | service_type [bsmod or equivalent] = 001)	SCTE: Music & Effects	N/A
Primary Audio (audio_type = 0x80)	main[3]	N/A
Native Audio (audio_type = 0x81)	absence of dub[4]	N/A
Emergency (audio_type = 0x82 | service_type [bsmod or equivalent] = 110)	emergency	N/A
Primary Commentary (audio_type = 0x83 | service_type [bsmod or equivalent] = 101)	main[5], commentary	N/A
Alternate Commentary (audio_type = 0x84)	alternate, commentary	N/A
ST = 100 or 111	TBD	N/A

Table 2 - Accessibility Associated Services

Type	**`Role`**`@value`	**`Accessibility`**`@value`
Audio description	alternate	description

[2] The role of "main" can occur across multiple adaptationSets of the period. This can apply across several dimensions of the media such as audio channel order, or audio codec format. Selection between multiple adaptation sets with the same language and role should be done according to the value of `@selectionPriority`.

[3] The supplemental property for media language default for audio *should* be set at the **`MPD`** or **`Period`** Level

[4] In DASH, native audio is not signaled explicitly but can be determined be the absence of "**`role`** = dub" to describe the track. In cases where program language may shift from program to program but all programs are in its native language, the channel language of "qaa" can be used to indicate the channel language is native but can vary in the channel. See [NorDig] see Table 12.6.

[5] The supplemental property for media language default for commentary *should* be set at the **`MPD`** or **`Period`** Level

(audio_type = 0x03 | service_type [bsmod or equivalent] = 010)		
Clean audio (audio_type = 0x02 | service_type [bsmod or equivalent] = 011)	alternate	enhanced-audio-intelligibility
Closed Captions[6]	main	captions
Sign Language[7]	Supplementary	sign

The expected practice in North America is that an audio adaptation set having `@contentType="audio"` and **`Role`**`@value = "main"` is equivalent to the audio service "Complete Main," which is defined for audio standards such as AAC and DTS. In North America, the "Complete Main" audio service is an audio component that contains a complete audio program (which typically includes dialog, music, silence, and effects).

For any two audio adaptation sets belonging to the same period element with the role of main, the values of `@lang` ***shall*** be identical.

The expected practice in North America is that audio adaptation sets having **`Role`**`@value = "commentary"` are equivalent to the audio service "commentary", which is defined for audio standards such as AAC and DTS.

DASH **`Role`** scheme `"urn:mpeg:dash:role:2011"` ***shall*** be used if there is more than one audio component a client can select (i.e., multiple audio services within a single multiplex or multiple audio adaptation sets).

In case of *ST=0* and multiple audio content components (as described above), **`Role`** descriptor with `@schemeIdUri="urn:mpeg:dash:role:2011"` and `@value="main"` ***shall*** be used.

In case of *ST > 1*, **`Role`** descriptor with `@schemeIdUri="urn:mpeg:dash:role:2011"` and `@value="alternate"` ***shall*** be used in case of full service, and `"supplementary"` ***shall*** be used otherwise ("receiver mix").

7.1.4. Full and partial audio services

An audio service *may* be a full service suitable for presentation, or only a partial service which *should* be combined with another audio service before presentation ("receiver mix"). In case the partial and the full services are in different adaptation sets, it is necessary to signal such dependence in order to indicate to the client that two adaptation sets need to be downloaded prior to the presentation.

Note: There is no need to signal this for a multiplex containing both – inband signaling in this multiplex is sufficient.

Let *F* be a boolean value, which indicates whether a service is a full service ('true'), or the client will need to combine it with a different audio service ('false').

---

[6] Closed captioning is an accessibility component for a video or text track indicated by the caption service descriptor in the PSI. The equivalent audio_type value would be 0x00

[7] Sign language is identified through the @lang attribute (e.g. "ase" or "bfi"). The equivalent audio_type value would be 0x00.

For AC-3 and E-AC-3 elementary streams, *F* is true if and only if the `full_svc` bit in the `AC-3_audio_stream_descriptor` is set to `1`.

For AAC, *F* is true if and only if `receiver_mix_rqd` is set to '0' (see SCTE 193-2 Table 1).

For DTS, *F* is true if and only if `full_service_flag` bit in `component_type` field is set to `1` (see SCTE 194-2 tables 6 and 7).

If neither signaled nor known by other means, *F* is assumed to be true.

In case *F* is false for an audio service in adaptation set A, and it needs to be combined with a different audio service in a different adaptation set B, this will be signaled in adaptation set A using an **`EssentialProperty`** descriptor with `@schemeIdUri` attribute value of `urn:mpeg:dash:audio-receiver-mix:2015` . The `@value` attribute ***shall*** the value of **`AdaptationSet`**`@id` of B.

Note 1: this signalling is defined in sec. 5.8.5.7 of ISO/IEC 23009-1 and was introduced in ISO/IEC 23009-1 3rd edtion.

Note 2: AC-3, E-AC-3 and AAC full service is signalled in PMT descriptors, hence when ISO-BMFF segments are generated from an MPEG-2 TS source, such signalling is expected to be translated into signalling defined in this section by the entity performing the container format conversion.

7.2. Caption service metadata in video media components

7.2.1. Introduction

CEA-608 and CEA-708 caption services are carried embedded in the elementary streams. Carriage of CEA-608 and CEA-708 in supplemental enhancement information (SEI) messages is defined in SCTE 128-1 and SCTE 215-1. This section describes **`MPD`** signaling of caption service metadata for and applies to content with both MPEG-2 TS and ISO-BMFF segments.

Signaling is done using the **`Accessibility`** descriptors, one per each standard. The value string of each descriptor can be either list of languages or a complete map of services (or CC channels, in CEA-608 terminology).

Listing languages without service/channel information is strongly discouraged if more than one caption service is present. At any time language-channel (CEA-608) or language-service (CEA-708) is known at content generation time, it ***shall*** be used, as opposed to signaling mere presence or presence and language.

Note: Signaling described in this section is identical to DASH-IF IOP.

7.2.2. Signaling CEA-708 caption service metadata

If CEA-708 closed caption service is carried in the video elementary stream, the relevant metadata per CEA-708 sec. 4.5 will be expressed using **`ContentComponent.Accessibility`** or, if the latter is not used, **`AdaptationSet.Accessibility`** with `@schemeIdURI` set to `urn:scte:dash:cc:cea-708:2015`.

The `@value` attribute ***shall*** contain the Caption Service Metadata as provided in CEA-708 section 4.5, as a semicolon-separated string of service descriptions. Each service description is either a single language code or a list of colon-separated name-value pairs

```
@value            = service *15 [";" service]
service           = language / ( service-number "=" param )
service-number    = (%d1 - %d63) ; decimal numbers 1 through 63
param             = language[",”easy-reader][",”aspect-ratio] [",”3d]
language          = "lang" ":" 3ALPHA; language code per ISO 639-3
easy-reader       = "er" ":" BIT ; default value 0
aspect-ratio      = "war" ":" BIT / "?"
                     ; default value is 1 (16:9),
                     ; value '0' indicates 4:3,
                     ; value '?' if unknown
3d                = "3D" ":" BIT
                     ; 1 if caption disparity data is present (CEA-708.1)
                     ; default value 0 (no 3d support).
```

Note: `ALPHA` and `BIT` are as defined by IETF RFC 5234, Appendix B.1.

Each of the `service` parameters (except for language) *may* be present or not present. Default values can be assumed where specified. If more than one language is present, the `service-number` ***shall*** be present. If `service-number` is known to the entity generating the **`MPD`**, it *should* be present even when a single language is signaled.

The CEA-708 information supplied in the **`Accessibility`** descriptor ***shall not*** contradict information supplied in the `caption_service_descriptor` in the PMT. See 0 below for derivation.

7.2.3. Signaling CEA-608 caption service metadata

If CEA-608 closed caption service is carried in the video elementary stream, language metadata will be expressed using **`AdaptationSet.Accessibility`** with `@schemeIdURI` set to `urn:scte:dash:cc:cea-608:2015`.

The `@value` attribute ***shall*** contain description of caption service(s) provided in the stream, as either a semicolon-separated list of languages or of colon-separated channel-language pairs. The `@value` syntax ***shall*** be as described in the Augmented Backnus-Naur Form (ABNF) below.

```
@value            = language / channel 0*3 [";" channel]
channel           = channel-number "=" language
channel-number    = “CC1” / “CC2” / “CC3” / “CC4”
language          = 3ALPHA ; language code per ISO 639-3
```

If language is known more than one language is present, the `channel-number` ***shall*** be present. If `channel-number` is known to the entity generating the **`MPD`**, it ***shall*** be present even when a single language is signaled.

7.2.4. Caption service examples

Table 3 - Caption Service Examples

```
<!-- Simple signaling of presence of CEA-608 closed caption service -->
<!-- NOTE: not signaling languages is a discouraged practice        -->
<Accessibility
    schemeIdUri="urn:scte:dash:cc:cea-608:2015"/>


<!-- Signaling of presence of CEA-608 closed caption service
<!-- in English and German, with channel assignments -->
<Accessibility
    schemeIdUri="urn:scte:dash:cc:cea608:2015" value="CC1=eng;CC3=deu"/>

<!-- Signaling of presence of CEA-708 closed caption service -->
<!-- in English and easy reader English                      -->
<Accessibility
    schemeIdUri="urn:scte:dash:cc:cea708:2015"
    value="1=lang:eng;2=lang:eng,war:1,er:1"/>
```

Note: For captioning, ABNF channel syntax is preferred and channel-less syntax is *deprecated*.

7.2.5. Derivation of caption service metadata from MPEG-2 TS

When **MPD** and media segments are generated from MPEG-2 transport stream, the PMT *may* contain the `caption_service_descriptor()` descriptor, as defined in Sec. 6.9.2 of ATSC A/65. If this descriptor is present, **MPD** signaling of caption service ***shall*** be generated using the procedure described below.

If there is a service for which `cc_data.digital_cc` bit is '0', then **Accessibility** with URI `urn:scte:dash:cea-608:2015` ***shall*** be used to signal it. If languages or channel-language association is known (from any source), it *should* be provided, using syntax from 7.2.3.

If there is at least one service with `cc_data.digital_cc` bit set to '1', then **Accessibility** with URI `urn:scte:dash:cea-708:2015` ***shall*** be used to signal it. For each such service syntax defined in **7.2.2** ***shall*** be used, and at least service number and language ***shall*** be provided.

Note 1: Descriptors for both CEA 608 and CEA 708 often appear in the same scope.

Note 2: PSI, and, consequently, caption service descriptors *may* change at splice points. In case of a splice we expect a new period to be started and the process above will be applied to the new period.

Note 3: Use of CEA 708.1 is not (as of 2015) reflected in `caption_service_descriptor()`

8. Signaling Asset Identification

8.1. General

AssetIdentifier elements are used in ISO/IEC 23009-1 to uniquely identify content in periods. This section identifies schemes that *may* be used in content compliant to this specification.

There *may* be several alternative identifiers applicable to the same content. **`AssetIdentifier`** elements are used for grouping periods with the same content and indicates to DASH client that one is continuation of the other. Thus, choice of main identifier (carried in **`AssetIdentifier`**) needs to reflect continuity in terms of user interface, random access, and trick modes. Alternate identifiers (carried in **`SupplementalProperty`**) are informational and may be used for reporting purposes.

As an example, when the same advertisement is shown multiple times, its instances will have same value of Ad-ID, but different AiringID. Use of Ad-ID as **`AssetIdentifier`** will result in DASH client considering the second instance of the advertisement a continuation of the first. In this case, e.g. the UI *may* show at the start of the second instance progress bar at 50% and advertisement duration of twice its actual duration. Thus AiringID *should* be used as **`AssetIdentifier`**, while Ad-ID *should* be used as an alternative identifier carried in **`SupplementalProperty`** with the same value of `@schemeIdUri`

In the opposite case, when two periods carry same in-network content which has the same EIDR and a period between them is an advertisement, the two in-network periods are continuation of each other and thus EIDR *should* be used in **`AssetIdentifier`**, while AiringID of these periods will be different and, if used, it *should* be used in **`SupplementalProperty`**

8.2. UPID Content Identification scheme

The value of **`AssetIdentifier`**`@schemeIdUri` or **`SupplementalProperty`**`@schemeIdUri` for this scheme ***shall*** have the value "`urn:scte:dash:asset-id:upid:2015`". The content of this **`AssetIdentifier`** or **`SupplementalProperty`** descriptor ***shall*** contain one or more **`ContentIdentifier`** elements defined below.

Only **`ContentIdentifier`** elements with the same scope can appear in the same **`AssetIdentifier`** or **`SupplementalProperty`** element.

8.2.1. ContentIdentifier element semantics

Table 4 - UPID ContentIdentifier Element Semantics

Element or Attribute Name	Use	Description
`ContentIdentifier`		Represents a textual value
`@type`	M	Type corresponding to SCTE 35 UPID type as specified in table 9-7. The value of this attribute ***shall*** be same as the value of `segmentation_upid()` (i.e., 3rd column of the table). MID ***shall not*** be used – the structure ***shall*** be translated into multiple UPID elements.
`@value`	M	Textual representation of the UPID value. It ***shall*** correspond to the description in the Description column (i.e., 4th column) of table 9-7 in SCTE 35. In case of the UPID contains binary encoding (e.g., ADI and ISAN), and a full textual representation is specified by the applicable standard, this textual representation ***shall*** be used.

Element or Attribute Name	Use	Description
		Otherwise, binary encoding is represented as a byte string in hexadecimal format.
Legend: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>...<maxOccurs> (N=unbounded) Elements are **bold**; attributes are non-bold and preceded with and @.		

8.2.2. XML syntax

Table 5 - UPID Content Identification XML Syntax

```
<xs:complexType name="UPID">
    <xs:sequence>
        <xs:any namespace="##other" processContents="lax"
                minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="type" type="xs:string" use="required"/>
    <xs:attribute name="value" type="xs:string" use="required"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
```

8.2.3. Example

Table 6 - UPID Content Identification Example

```
<AssetIdentifier schemeIDUri="urn:scte:dash:asset-id:upid:2015">
    <!-- ADI of the asset -->
    <scte214:ContentIdentifier type="ADI" value="cablelabs.com/MOVE1234567890123456>
    <!-- Alternative ID using an opaque provider-specific scheme -->
    <scte214:ContentIdentifier type="MPU" value="CSP1DE12AB327FE312AF"/>
</AssetIdentifier>

<SupplementalProperty schemeIDUri="urn:scte:dash:asset-id:upid:2015">
    <!-- AiringID of the asset -->
    <scte214:ContentIdentifier type="AiringID" value="0xDEADBEEF"/>
</AssetIdentifier>
```

8.3. Use of EIDR and Ad-ID

In case only a single identifier is needed, it is recommended to use EIDR or Ad-ID.

DASH-IF IOP v4.3 sec 5.5.1 normatively describes use of Ad-ID and EIDR in **`AssetIdentifier`** descriptors. The text below describes that approach.

EIDR: The value of `@schemeIdUri` is set to `"urn:eidr"` to signal the use of EIDR. The value of the `@value` attribute ***shall*** be a valid canonical EIDR entry

Ad-ID: Use of Ad-ID for asset identification is signaled by setting the value of `@schemeIdUri` to `"urn:smpte:ul:060E2B34.01040101.01200900.00000000"`. The value of `@value` attribute ***shall*** be a canonical full 22 Ad-ID identifier as defined in SMPTE 2092-1

9. Generic restrictions on media segments

9.1. Terminology

For the purpose of this section the following variables are defined for any segment *S(n)* and its k^{th} subsegment *n[k]*:

EPT(n) := earliest presentation time of segment *n*. *EPT(0) = 0*;

EPT(n[k]) := earliest presentation time of its subsegment *n[k]*. *EPT(n[0]) = EPT(n)*.

SD(R) := signaled segment duration for representation R, as expressed e.g. in the `@duration` attribute or the **`S`**`@d` in the **`SegmentTimeline`**. While this applies to a specific representation *R*, segment alignment requirement in this specification requires this value to be identical for all representations in an adaptation set.

MSD := maximum segment duration, as indicated in **`MPD`**`@maximumSegmentDuration`

MSSD := maximum subsegment duration, as indicated in **`MPD`**`@maximumSubsegmentDuration`

D(n) := "real" presentation duration of segment *n*, i.e. *EPT(n+1) – EPT(n)*.

SD(n[k]) , the signaled subsegment duration, is same as *D(n[k])*, presentation duration of segment *k*, and is provided in `` `sidx`.subsegment_duration[k] `` of the `sidx` box indexing segment *n*.

BW[R] := value of **`Representation`**`@bandwidth` of a representation *R* to which segment n belongs. **Note** that this is the peak and not the average bandwidth.

MBT := value of **`MPD`**`@minBufferTime`

All durations are in seconds, and bandwidth is given in bits per second.

9.2. Duration

9.2.1. Segments

If representation contains more than one segment and is used for normal playback, the following restrictions ***shall*** be met:

1. Unless the **`SegmentTimeline`** element is used to express precise timing, Segments ***shall*** have almost equal "real" duration. The maximum tolerance of "real" segment duration *D(n)* ***shall*** be ±50% of the stated segment duration, and the accumulated drift ***shall not*** exceed 50% of the stated segment duration *SD*.

$$\boldsymbol{abs}\left(\sum_{\boldsymbol{i=0}}^{\boldsymbol{n-1}} \boldsymbol{D(i)} - \boldsymbol{(n+1)} * \boldsymbol{SD(R)}\right) \leq \boldsymbol{0.5} * \boldsymbol{SD(R)}$$

Note 1: This is done so that if seeking is done using stated duration, correct segment will be identified despite the accumulating drift.

Note 2: drift *may* develop due to mismatch between *D* and *SD* due to imprecision of the clock used to state SD. For example, if *SD*=2 sec and segments are 2002 ms each, ±50% drift will be exceeded in less than 10 minutes.

2. The "real" segment duration for representations containing more than one segment ***shall*** be between 1.5 and the defined value for MSD, except for the last segment

$$1.5 \leq D(n) \leq MSD;$$

Note: This is done in order to simplify client implementation and compatibility with existing clients

9.2.2. Subsegments

For representation used for normal playback and containing subsegments, the "real" subsegment duration ***shall*** be less than the defined value for MSSD.

$$D(n[k]) \leq SSD_{max} = MSSD;$$

9.2.3. Segment duration patterns (Deprecated)

This section and its subsections information are *deprecated*.

9.2.3.1. Syntax and Semantics

If segment durations follow a well-defined pattern, the segment duration specified in the MPD *should* be the average duration. In case of number-based addressing this *should* be average over the duration of the period, while in case of **`SegmentTimeline`** it *should* apply only to segments described in an **`S`** element.

Note: There is no requirement to specify a precise segment duration – an approximation is good enough as long as the restrictions in 9.2.1 are maintained.

If there is a requirement for higher precision for precise lookup purposes, the following attributes are defined in the SCTE DASH namespace:

Element or Attribute Name	Use	Description
`@offsetTimescale`	OD	specifies the timescale in units per seconds to be used for the derivation of precise duration values in the Segment Information. Default value is 1.
`@offsetPattern`	OD	specifies a repeating pattern of offsets. Each offset is a signed integer in units specified by `@offsetTimescale`. For a pattern with *N* offsets, segment *i* has offset O(i) = `offsetPattern[i%N]` The relation between real and stated duration of the segment is given by $D(n) = SD + O(n)$ Default value is 0 (i.e., a single offset of 0)
Legend: For attributes: M=Mandatory, O=Optional, OD=Optional with Default Value, CM=Conditionally Mandatory. For elements: <minOccurs>...<maxOccurs> (N=unbounded) Note that the conditions only holds without using `xlink:href`. If linking is used, then all attributes are "optional" and <minOccurs=0> Elements are **bold**; attributes are non-bold and preceded with an @.		

The attributes *may* be used in **`S`** element or in **`SegmentBase`** and elements derived from it.

Note 1: Offsets are intended for precision purposes and are purely informational. In particular they do not affect URL construction.

9.2.3.2. Example

Let us assume **`SegmentBase`**`@timescale = 1000` and `@scte214:offsetTimescale = 90000`

Let us further assume 2-sec segments with a pattern of 12 segments of 180480 90KHz clock ticks followed by a shorter 176640-tick segment, and *SD* = 2002.

In this case `@scte214:offsetPattern =”300 300 300 300 300 300 300 300 300 300 300 300 -3540”.`

In our case the number of offsets (i.e., number of elements in the `@scte214:offsetPattern` list) is 12. Therefore segment *i* has duration (in 90KHz clock ticks) of *D(n) = 2002×90 + offsetPattern[i%N]*. For *i=0…11* the result will be *2002×90 + 300 = 180480*.

9.3. Bandwidth, size, and buffering

9.3.1. Introduction

This section formalizes the relationship between the declared bandwidth *BW[R]*, *MBT*, and segment sizes.

The derivations below are a straightforward, albeit lengthy, translation of the requirement in ISO/IEC 23009-1 that if segments of representation *R* are delivered over a constant bitrate channel with bitrate equal to *BW[R]* attribute, then each sample with decoding time *DT* is available for decoding at the media engine by time *DT* + *MBT*.

Note: In many cases the latter *may* be a stricter limitation that the ones stated in the sections below, as the discussion below applies to complete (sub)segments, rather than samples.

While MBT does specify minimum time sufficient for ensuring continuous playout of a representation, it describes content encoding properties, rather than expected network behavior. Hence a player implementation has to account for realistic network conditions, and this specification provides neither restrictions nor any guidance on these issues.

9.3.2. Segments

Let SD_{max} be the maximum segment duration. For representations containing more than one segment it is defined as follows:

$$SD_{max} = min(1.5 * SD(R), MSD)$$

Let $SZ_R(n)$ be the size (in bits) of segment *n* from representation *R*.

Note: $SZ_R(n)$ is the size of the complete segment including all headers, `sidx` and `ssix` boxes (for ISO-BMFF) and inband events.

Let MBT_s be the minimum buffer time in units of segments, defined as $MBT_s = ceil\left(\frac{MBT}{SD_{max}}\right)$

Note: Buffer size of $BW[R] * SD_{max} * MBT_s$ is sufficient for playback of representation R under idealized network conditions (i.e., assuming constant download rate).

For any representation that contains *N>1* segments and is used for normal playback, the following restrictions ***shall*** be met:

1. Any segment *n* ***shall not*** exceed the buffer size, hence the following ***shall*** hold:

$$SZ_R(n) \leq BW[R] * SD_{max} * MBT_s$$

2. Combined size of any MBT_s consecutive segments ***shall not*** exceed the buffer size, hence for any $0 \leq k \leq \mathrm{N} - MBT_s$, the following ***shall*** hold:

$$\sum_{i=k}^{MBT_s} SZ_R(i) \leq BW[R] * \sum_{i=k}^{MBT_s} D(i)$$

Note 1: In case of inband events care *should* be taken to keep events small enough in order not to break the model above.

Note 2: For representations without subsegments it is often useful to set *MBT* to SD_{max}. For representations containing subsegments SSD_{max} *may* be a better alternative. This agrees with the recommendation in DASH-IF IOP 4.3.

9.3.3. Video aspects

***MBT shall** not* be less than coded picture buffer (CPB) removal delay.

*MBT*BW[R]* ***shall*** equal or exceed the size of CPB.

10. Codec-Specific Aspects

10.1. Video

10.1.1. Supported video codecs

The following video codecs ***shall be*** supported in SCTE DASH profiles:

1. AVC (ISO/IEC 14496-10, restrictions in SCTE 128-1)
2. HEVC (ISO/IEC 23008-2, restrictions in SCTE 215-1)

10.1.2. Resolutions and frame rates

This specification neither specifies nor requires support for specific operating points (i.e., combination of resolution, frame rate and aspect ratio).

10.1.3. SAP values

10.1.3.1. AVC video

Segments starting from an IDR picture in decoding order have SAP value of 1, unless this IDR picture is followed by a picture which precedes it in presentation order. In the latter case the segment has SAP value of 2.

10.1.3.2. HEVC video

Segments starting from pictures with nal_unit_type equal to IDR_N_LP or BLA_N_LP have SAP value of 1.

Segments starting from IDR_W_RADL or BLA_W_RADL have SAP value of 2.

10.1.4. Multiplexed segments

When a segment contains video and one or more audio elementary streams, its SAP value is the SAP value of the video elementary stream.

10.1.5. Colorimetry

`AdaptationSet.SupplementalProperty or AdaptationSet.EssentialProperty` descriptors ***shall*** be used to signal source signal information in the manifest to indicate color primaries, optoelectronic transfer characteristics, and matrix coefficients properties for derivation of luma and chroma signals. If all video adaptation sets contain only BT.709 colorimetry, the above signaling is recommended.

Note 1: If DVB compatibility is needed, **`AdaptationSet.EssentialProperty`** ***should*** be used. Please see ETSI TS 103 285 V1.3.1 sec 5.2.6 for backwards-compatible HLG signaling.

Note 2: In absence of the colorimetry signaling due to no indication in the high level syntax of the encoded stream, we can assume BT.709 SDR colorimetry.

The URNs and corresponding values are defined in ISO/IEC 23091-2 and are informatively provided in the table below.

Table 7 - Colorimetry

`@schemeIdUri`	**`@value`**
`urn:mpeg:mpegB:cicp:ColourPrimaries`	See ISO/IEC 23091-2 sec. 8.1
`urn:mpeg:mpegB:cicp:TransferCharacteristics`	See ISO/IEC 23091-2 sec. 8.2
`urn:mpeg:mpegB:cicp:MatrixCoefficients`	See ISO/IEC 23091-2 sec. 8.3

10.1.6. Supplemental Codecs

Supplemental codecs are defined as backwards-compatible codecs enhancing the experience of a base layer codec as described as **`AdaptationSet`** attributes in part 15 of section 6.3. Annexes A and B of SCTE 215-1-1 specify such codecs, DolbyVision profile 8.1 and HDR10+. In both cases, the "base layer" is HEVC HDR10, support for which is essential for rendering. If supported use of the optional DolbyVision or HDR10+ codecs enhances viewers' experience.

This specification defines two additional XML attributes used within the RepresentationBaseType element defined in ISO/IEC 23009-1 [DASH].

Table 8 - Supplemental XML Attrributes

Attribute Name	Use	Description
Common attributes and elements		
`@scte214:supplementalCodecs`	0	specifies a space-separated list of codec strings for supplemental codecs used in the associated Representation(s). These codec strings shall comply with the syntax defined in IETF RFC 6381 When multiple codec strings are present, the order of codec appearance is immaterial.

Attribute Name	Use	Description
`@scte214:supplementalProfiles`	O	specifies a space-separated list of supplemental segment profiles (expressed as ISO-BMFF brands or URIs) to which the associated Representation(s) conform.

For an HDR10 (chd1) content stream containing metadata for DolbyVision (e.g. dvh1.08.09 for 215-1-1) and HDR10+ (cdm4) the syntax would be:

a. **AdaptationSet**`@segmentProfiles="chd1"`
b. **AdaptationSet**`@scte214:supplementalProfiles = "cdm4"`
c. **AdaptationSet**`@scte214:supplementalCodecs= "dvh1.08.09"`

NOTE 1: the difference between `supplementalProfiles` and `supplementalCodecs` is technical: the latter specifies codec strings based on MP4RA-registered codec code points and containing additional capability-related information such as profiles and levels, while the former specifies MP4RA-registered brands (such as CMAF brands) or URIs.

NOTE 2: the concept of supplemental codecs is not limited to HDR dynamic metadata – same approach may work for video coding standards such as LC-EVC, SVC, or SHVC, which are currently unsupported in this specification.

10.2. Audio

10.2.1. Supported codecs

The following audio codecs are supported in SCTE DASH profiles:

1. (E-)AC-3 (ATSC A/52, restrictions in A/53 Parts 5-6), AC-3 is only supported for MPEG-2 TS
2. AAC (ISO/IEC 14496-3, restrictions in SCTE 193-1)
3. DTS-HD (ETSI TS 102 114, restrictions in SCTE 194-1)

The following next generation audio codecs could be supported in the SCTE DASH Profiles:

1. AC-4 (restrictions in SCTE 243-2)
2. MPEG-H (restrictions in SCTE 243-3)
3. DTS-UHD (restrictions in SCTE 243-4)

10.2.2. SAP values

For AC-3, E-AC-3, DTS and AAC, all segments ***shall*** have SAP value of 1.

AAC segments ***shall*** be start with a RAP AU (as defined in SCTE 193-1) and *should* be encoded according to the MPEG DASH Implementation Guidelines sec. 5.1.2 in order to ensure seamless bitstream switching.

10.3. Trick Modes

10.3.1. Introduction

Playback of media content at speed and / or direction other than the ones intended for normal playback of this asset is referred to as *trick modes*. Trick modes include modes like fast forward, slow motion, and rewind; and are used to emulate visual experience of rewinding analog videotapes.

Trick modes can be implemented in multiple ways, starting from fetching segments at a different speed, to maintaining special trick mode representations, to bringing only specific frames from the segment. This standard does not prescribe a particular implementation strategy or combination of strategies. ETSI TS 103 285 sec. 6.2 provides a long discussion about ways of implementing trick modes in DASH, while encoding techniques discussed in SCTE 128 provide a content preparation perspective.

Note: SCTE 128 and SCTE 215 discuss trick modes based on extraction of identifiable pictures that result in respective decodable sub-bitstreams, or conversely, on discarding identifiable pictures to obtain respective decodable sub-bitstreams. This functionality can be implemented using Subsegment Index ('`ssix`') boxes

Trick modes are not necessarily permitted in all content – sometimes certain modes will be disallowed. This restriction model is described in SCTE 130-10, and sec. 11.3 defines its integration into DASH **`MPD`**.

10.3.2. Trick mode representations

`Periods` *may* contain adaptation sets with representations intended for use in trick modes (e.g., representations with low frame rate). Such adaptation sets ***shall*** employ signaling as defined in DASH-IF IOP 4.3 with additional signalling defined below.

Trick mode adaptation sets ***shall*** be marked with a **`SupplementalProperty`** or **`EssentialProperty`** element with `@schemeIdUri` value of "`http://dashif.org/guidelines/trickmode`" and the `@value` the value of the **`AdaptationSet`**`@id` attribute of the adaptation set(s) to containing “normal” (non-trick-mode) representations of the same content.

10.3.3. I-Frame track representations

Fixed frame rate trick mode adaptation sets *may* create segment boundary mismatches unless all segments in the “normal” representation have precisely the same number of frames.

In this case it is possible to have a trick mode adaptation set aligned with “normal” adaptation set(s), where the segment boundaries of the trick mode representations match the segment boundaries of the “normal” representations. As a result, the frame rate *may* be variable due to variable duration of the “normal” representation segments.

The I-Frame representations defined in this clause are a subset of DASH-IF trick modes maintaining segment alignment for variable-duration segments and compatible with IETF RFC 8261 (HLS).

The following ***shall*** hold for I-Frame track representations:

1. Each media segment ***shall*** consist of a single IDR frame, and its EPT ***shall*** match the EPT of the corresponding “normal” media segment;

NOTE: The above implies that there can be more single-frame segments than “normal” segments – an I-frame track may be used for functionality such as “fast join”.

2. DASH-IF trick mode signaling ***shall*** be present as describe in 10.3.2 above.

3. There ***shall*** be a **`EssentialProperty`** element with `@schemeIdUri` value of "`urn:scte:dash:i-frame-track:2021”.`

4. The value of the `@frameRate` attribute ***shall*** be a long-term approximation of the frame rate (i.e., average frame rate per period or a reasonable expectation of it in case of live content). For example, for segments with target duration of 2 seconds and occasional short segments due to ad insertion, this value *should* be 1/2.

11. Multi-period assets

11.1. Period continuity

If multi-period content is offered (e.g., when some of the periods represent placement opportunities), periods with identical **`AssetIdentifier`** elements are considered as contiguous parts of the same asset.

If an asset spans over more than one period, **`Period.AssetIdentifier`** element *should* be present in each such period.

`Periods` with identical asset identifiers ***shall*** be *period-continuous* as specified in ISO/IEC 23009-1.

11.2. Asset boundaries

If multi-period content is offered in a dynamic **`MPD`**, periods can be removed and/or added during the presentation. In these cases, the author *may* want to preserve the information regarding the playback location in time in order to allow e.g. correct display of time in UI.

If a period is the last period of a given asset, this *may* be signaled using **`Period.SupplementalProperty`** with `@schemeIdUri="urn:scte:dash:asset-end".`

Correspondence of *PeriodStart* to the time of the asset *may* be signaled using **`Period.SupplementalProperty`** with `@schemeIdUri="urn:scte:dash:asset-time".` The value of `@value` attribute ***shall*** be the timestamp corresponding to *PeriodStart*, as NPT or SMPTE relative timestamp, as defined in RFC 2326.

Correspondence of *PeriodStart* to UTC time *may* be signaled using **`Period.SupplementalProperty`** with `@schemeIdUri="urn:scte:dash:utc-time".` The value of `@value` attribute ***shall*** be the timestamp corresponding to *PeriodStart*, in format defined in RFC 3339.

Note: The difference between the asset time and UTC time is that asset time is relative to the asset start, while UTC time is the UTC time corresponding to the acquisition time of the first sample of the period. Thus, asset time will show that a period starts at 42nd minute of an asset, while UTC time will show that the period starts with content sent to origin on October 21, 2015 at 4:29am.

11.3. Stream restrictions

Period elements *may* contain a **`SupplementalProperty`** element with **`SupplementalProperty`**`@schemeIdUri` value of "`urn:scte:scte130-10:2014 `". The content of the descriptor is the SCTE 130-10 **`StreamRestrictionList`** element.

`NptRange` in this descriptor ***shall*** be relative to *PeriodStart* and the restrictions ***shall*** be valid only for the duration of the period in which the **`SupplementalProperty`** element appears.

Note: Given `@nptstart` value of N_s, `@nptend` value of N_e, and period duration D, the restrictions in the **`StreamRestrictionList`** element are valid in the range [max(0:00.00, N_s), min(D,N_e)].

12. URL query parameters for XLink

12.1. Introduction

The only interface between an entity dereferencing XLink and the DASH client is the XLink URL. Passing avail metadata such as SCTE 35 cue messages *may* be useful when remote periods are used to signal upcoming avails. This usage is described in DASH-IF IOP and referred to as "server-based ad insertion".

The section below defines standardized names and values for parameters that can be used in XLink URLs. It defines the parameter name and the derivation of its value. Name and value ***shall*** contain only characters permitted by RFC 3986 and *may* need to be percent-coded.

The parameters defined in this section are not mandatory – i.e., the author is not required to use them. On the other hand, query parameter with the name defined in this section appears in a URL, its value ***shall*** be derived in a way defined in this section. The author ***shall not*** use query parameter names defined in this section with any different syntax and semantics.

The parameters in this section ***shall not*** be used in elements other than Period

12.2. Notation

The standardized parameters introduced in this section are <name>=<value> pairs embedded in a URL query string (as defined in RFC 3986). The <field> string in the above construct will be replaced with parameter name For example, for parameter named `param` and having value `42`, the URL will be http://example.com?param=42

Note: <name> is often referred to as "field" in different descriptions of URL query parameter syntax.

12.3. Carriage of SCTE 35 in a query parameter

Parameters defined in this section carry the complete SCTE 35 binary cue message, as well as message timing.

Table 9 - Carriage of SCTE 35 in a Query Parameter

<name>	<value>	format
`scte35-cue`	`splice_info_section()`	base64url-encoded string with padding. Pad character ('=') replaced with "%3D" See Note 1 and 2 below.

<name>	<value>	format
scte35-time	Offset of splice start from presentation start time. This is a translation of splice_time() into presentation timeline. The time *should* be provided with millisecond accuracy if feasible. See note 3 below	xsd:duration (defined in XML Schema Part 2, based on ISO 8601)

Note 1: Percent encoding is required to escape reserved characters in query string. Thus '=' character used as a padding character in base64 and base64url thus needs to be replaced with %3D. While splice_info_section() size is provided within the section, skipping padding per RFC 4648 sec 3.2 is explicitly disallowed in the definition above for simplicity and interoperability reasons.

Note 2: The format of the scte35-cue parameter is identical to the base64-coded format used in CUE attribute of #EXT-X-SCTE35 tag in SCTE 35 2020 sec. 12.2.3, with the following differences: (a) percent-coding of padding character, and (b) since base64url and not base64 alphabet is used, characters "+" and "/" in the CUE attribute are respectively replaced by "-" (minus) and "_" (underscore), per RFC 4648.

Note 3: The scte35-time value is equivalent to the value of *PeriodStart* corresponding to the start of the avail if an avail is represented by a Period element. It is equivalent to the TIME attribute of the #EXT-X-SCTE35 tag in SCTE 35 2016. In the latter case the TIME attribute gives absolute time, while scte35-time value is relative to start of the presentation.

Note 4: The definition in this section does not preclude use of SCTE 35 in DASH events (either inband or MPD), as defined in this standard.

Example

Table 10 - Remote Period element with SCTE 35 embedded in XLink URL

```
...
<Period duration="PT60.0S" id="ad break #1"
         xlink:href="https://adsrus.com/avail.mpd?scte35-cue=
                     DAIAAAAAAAAAAAQAAZ_I0VniQAQAgBDVUVJQAAAAH+cAAAAAA%3D%3D"
         xlink:actuate="onLoad" >

     <!-- Default content, replaced by elements from remote entity -->
     <AdaptationSet mimeType="video/mp4" codecs="avc1.640828"
                    frameRate="30000/1001"
                    segmentAlignment="true" startWithSAP="1">
        <BaseURL availabilityTimeOffset="INF">default_ad/</BaseURL>
        <SegmentTemplate timescale="90000" initialization="$Bandwidth%/init.mp4v"
                         media="$Bandwidth%/$Time$.mp4v"/>
            <Representation id="v0" width="320" height="240" bandwidth="250000"/>
            <Representation id="v1" width="640" height="480" bandwidth="500000"/>
            <Representation id="v2" width="960" height="720" bandwidth="1000000"/>
     </AdaptationSet>
</Period>
```

12.4. Carriage of geographical information in a query parameter

This section lists parameters can be used for carriage of geographical information in URLs.

Note that these parameters may have privacy implications, hence it is recommended to use HTTP over TLS (e.g., HTTPS) if the HTTP GET with these parameters is expected to be sent via an insecure channel.

Table 11 - Carriage of Geographical Information in a Query Parameter

<name>	<value>	format
`scte-dash-syscode`	`syscode`	Decimal number, assigned by NCC Media

13. Annotation

13.1. Generator identification

It is often necessary for debug purposes to indicate which software generated an MPD or responded to an XLink dereferencing request. In order to allow such information, the following schemes can be used in **`MPD.SupplementalProperty`** and/or in **`Period.SupplementalProperty`** elements:

Table 12 - Generator identification

`@schemeIdUri`	`@value`
`urn:scte:dash:powered-by:2016`	Human-readable string containing software name and version that generated this MPD or responded to the XLink dereferencing request
`urn:scte:dash:generation-info:2016`	Comma-separated list of name=value pairs with one or more of the following values: 1. `location=<location>`: same as location header in HTTP (RFC 7213), as an absolute URI 2. `date=[time]`: generation time (UTC) of the body of the HTTP response containing the , in format specified by ISO 8601. The time *should* be provided with millisecond accuracy if feasible.
`urn:scte:dash:generation-request:2016`	Request target string (per RFC 7230 sec 5.3) from the HTTP request which resulted in generating the current response (MPD or Period). Example: value of syscode, assuming `scte-dash-syscode` query parameter was used

Example

For remote period from example in Table 10, the remote entity (i.e., the contents of the HTTP response) containing generator identification information is described in Table 13 below.

In the example below the XLink URL from in Table 10 (sans host name) is reflected in the generation request information, while the arrival time of the XLink request and the server name of the responding server are reflected in the generation info.

Table 13 - Remote entity with generator information

```
<Period duration="PT60.0S" id="inserted ad #1" >

    <SupplementalProperty schemeIdUri="urn:scte:dash:generation-info:2016"
        value="location=adsrus.com date=2016-01-07T15:22:16-07:00"/>

    <SupplementalProperty schemeIdUri="urn:scte:dash:generation-request:2016"
        value="/avail.mpd?scte-dash-syscode=123456789&&scte35-
        cue=DAIAAAAAAAAAAAQAAZ_I0VniQAQAgBDVUVJQAAAAH+cAAAAAA%3D%3D"/>

    <!-- Replaced ad content, replacing by elements from remote entity -->
    <AdaptationSet mimeType="video/mp4" codecs="avc1.640828"
        frameRate="30000/1001"
        segmentAlignment="true" startWithSAP="1">
        <BaseURL availabilityTimeOffset="INF">ad/</BaseURL>
        <SegmentTemplate timescale="90000" initialization="$Bandwidth%/init.mp4v"
            media="$Bandwidth%/$Time$.mp4v"/>
        <Representation id="v0" width="320" height="240" bandwidth="250000"/>
        <Representation id="v1" width="640" height="480" bandwidth="500000"/>
        <Representation id="v2" width="960" height="720" bandwidth="1000000"/>
    </AdaptationSet>
</Period>
```

14. Content Protection

14.1. Multi-key encryption

There *may* be cases in which there is a need to encrypt content of the same type in the same **`MPD`** with different keys. This requirement often arises when the output protection requirements differ across different video resolutions.

In this case, subsets of representations sharing the same key will be combined into adaptation sets. For example, a 4K video ladder, *may* be broken into three adaptation sets, SD (up to 540p resolution), HD (720p and 1080p resolution) and UHD (above 1080p resolutions). These adaptation sets are aligned and are seamlessly switchable.

`AdaptationSet``@profiles` value of `scte:dash:multi-key:2021` *should* appear in all adaptation sets following the constraints in this clause.

Note: This implementation conforms to the DASH-IF IOP guidelines. Other implementations of multi-key encryption *may* be legal but discouraged unless needed for legacy devices.

1. All representations in an **`AdaptationSet`** ***shall*** share the same key;
2. Adaptation sets containing parts differently encrypted parts of perceptually identical content ***shall*** be aligned and switchable per definition of adaptation set switching in ISO/IEC 23009-1.
3. All adaptation sets in the above switchable adaptation sets ***shall*** have identical values of the following elements and attributes:

    a. **`AudioChannelConfiguration`**

    b. **`Accessibility`**

    c. **`Role`**;

    d. `@audioSamplingRate`;

    e. `@lang`;

    f. `@codecs`, unless codecs differ by profile/level.

4. If the use of different keys is due to robustness or/and output protection, then these ***should*** be explicitly stated:
    a. If a particular level of DRM robustness (e.g. hardware decryption and decoding) is required, this needs to be specified in the **`ContentProtection`**`@robustness` attribute. The value of the attribute is DRM-specific and is typically described in the DRM documentation.
    b. If a particular HDCP level is required for playback of a specific adaptation set, then the **`OutputProtection`** descriptor ***shall*** be used

Note: Use of **`ContentProtection`**`@robustness` and **`OutputProtection`** signaling is needed to prevent download of segments which are expected to be undecodable by the client. The source of truth is the DRM, and not this segment. This means that it is a DRM decision whether a client is or is not allowed to decode specific representations.

Annex A URNs

The following URNs are defined in this specification:

urn:scte:dash:2021
XML namespace for this specification.

urn:scte:dash:essential-event:2015
Event essentiality, see 6.7.3
urn:scte:scte35:2013:xml
XML representation of SCTE 35, see 6.7.2.1

urn:scte:scte35:2014:xml+bin
base64-coded representation of SCTE 35 wrapped in XML, see 6.7.2.1

urn:scte:dash:associated-service:2015
Roles for non-accessibility associated audio services, see 7.1

urn:scte:dash:cc:cea-608:2015
Signaling of CEA-608 closed captions, see 7.2.3

urn:scte:dash:cc:cea-708:2015
Signaling of CEA-708 closed captions, see 7.2.2

urn:scte:dash:asset-id:upid:2015
Asset identifier scheme based on SCTE 35 UPIDs, see 8.2

urn:scte:dash:asset-end
End of last period of an asset, see 11.2

urn:scte:dash:asset-time
NPT corresponding to asset time, see 11.2

urn:scte:dash:utc-time
UTC time corresponding to asset time, see 11.2

urn:scte:scte130-10:2014
Carriage of SCTE 130-10, see 11.3

urn:scte:dash:powered-by:2016
Information on MPD / XLink authoring software

urn:scte:dash:generation-info:2016
Time and place of MPD or XLink remote entity generation

urn:scte:dash:generation-query:2016
Time and place of MPD or XLink remote entity generation

urn:scte:dash:default-language
Information on default language in MPD see section 7.1.1.

urn:scte:dash:i-frame-track:2021
Information on I-Frame representation