ISO/IEC JTC 1/SC 29/WG 03 N0780

**ISO/IEC JTC 1/SC 29/WG 03  
MPEG Systems  
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output Document

**Title:** Text of ISO/IEC 23009-1 5th edition CDAM 2 EDRAP streaming, content steering and other extensions

**Status:** Approved

**Date of document:** 2023-02-24

**Source:** ISO/IEC JTC 1/SC 29/WG 03

**No. of pages:** 37 (with cover page)

**Email of Convenor:** young.L @ samsung . com

**Committee URL:** <https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg3>

**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 03 MPEG SYSTEMS**

**ISO/IEC JTC 1/SC 29/WG 03 N0780**

**January 2023, Online**

|  |  |
| --- | --- |
| **Title** | **Text of ISO/IEC 23009-1 5th edition CDAM 2 EDRAP streaming, content steering and other extensions** |
| **Source** | **WG 03, MPEG Systems** |
| **Status** | **Approved** |
| **Serial Number** | **22304** |

# Change 1: EDRAP Streaming

*In clause 2, add the following reference:*

*ISO/IEC 14496‑12:2021 AMD1[[1]](#footnote-1), Information technology — Coding of audio-visual objects — Part 12: ISO base media file format, AMD 1 Improved brand documentation and other improvements*

*In subclause 3.2, add the following abbreviations:*

|  |  |
| --- | --- |
| EDRAP | extended dependent random access point |
| ESR | external stream Representation |
| MSR | main stream Representation |

*Add subclause 5.8.5.15 as follows:*

***5.8.5.15 MSR and ESR descriptors***

**5.8.5.15.1 General**

An Adaptation Set may have an **EssentialProperty** descriptor with @schemeIdUri equal to urn:mpeg:dash:msr:2022. This descriptor is referred to as the MSR descriptor. This descriptor may only be present in an Adaptation Set level and its presence indicates that each Representation in that Adaptation Set is an MSR, which carries a video track containing a track reference of type 'aest' as specified in ISO/IEC 14496 12:2021 AMD1.

An Adaptation Set may have an **EssentialProperty** descriptor with @schemeIdUri equal to urn:mpeg:dash:esr:2022. This descriptor is referred to as the ESR descriptor. This descriptor may only be present in an Adaptation Set level and its presence indicates that each Representation in the Adaptation Set is an ESR, which carries a video track referenced by a track reference of type 'aest' as specified in ISO/IEC 14496 12:2021 AMD1. An ESR is only intended to be consumed or played back together with its associated MSR.

Each ESR shall be associated with an MSR through the Representation-level attributes @associationId and @associationType in the MSR as follows: the @id of the associated ESR shall be referred to by a value contained in the attribute @associationId for which the corresponding value in the attribute @associationType is equal to 'aest'. Each MSR shall have an associated ESR.

For an MSR and an ESR associated with each other, the following applies:

* For each media sample with a particular presentation time in the ESR, there shall be a corresponding media sample with the same presentation time in the MSR.
* Each media sample in the MSR that has a corresponding ESR media sample is referred to as an EDRAP sample.
* The first byte position of each EDRAP sample in the MSR shall be the ISAU of a SAP, which enables playback of the media stream in the MSR provided that the corresponding ESR media sample is provided to the media decoder immediately before the EDRAP sample.
* Each EDRAP sample in the MSR shall be the first sample in a Segment or Subsegment (i.e., each EDRAP sample shall start a Segment or Subsegment).
* For each Segment or Subsegment in the MSR that starts with an EDRAP sample, there shall be a Segment in the ESR with the same earliest presentation time as the MSR Segment or Subsegment. This Segment in the ESR is referred to as the corresponding ESR Segment of the MSR Segment or Subsegment and vice versa.
* The concatenation of any Segment in the ESR and the corresponding MSR Segment or Subsegment (i.e., the MSR Segment or Subsegment having the same earliest presentation time as the ESR Segment) and all subsequent MSR Segments or Subsegments shall result in a conforming bitstream.
* For each MSR Segment or Subsegment that does not start with an EDRAP sample, there shall be no corresponding ESR Segment having the same earliest presentation time as the MSR Segment or Subsegment.

**5.8.5.15.2 Example content preparation and client operations (informative)**

Below are example content preparation and client operations based on MSRs and their associated ESRs.

An example of content preparation operations is as follows:

1. A video content is encoded into one or more representations, each of which is of a particular spatial resolution, temporal resolution, and quality.
2. Each representation of the video content is represented by a pair of MSR and ESR associated with each other.
3. The MSRs of the video content are included in one Adaptation Set. The ESRs of the video content are included in another Adaptation Set.

An example of client operations is as follows:

1. A client gets the MPD of the Media Presentation, parses the MPD, selects an MSR.
2. When initializing a session or performing seeking, the client determines the starting presentation time from which the content is to be consumed, requests Segments or Subsegments of the MSR, starting from the Segment or Subsegment starting with a SAP and containing the sample having presentation time equal to (or earlier than but close enough to) the determined starting presentation time. For requesting Subsegments in a Segment, a Segment Index is requested beforehand to obtain information of the Subsegments and partial HTTP GET requests are used.
   1. If in the associated ESR there is a Segment having the same earliest presentation time as the starting MSR Segment or Subsegment, that ESR Segment is also requested, preferably before requesting of the starting MSR Segment or Subsegment. Otherwise, no Segment of the associated ESR is requested.
3. When switching to a different MSR, the client requests Segments or Subsegments of the switch-to MSR, starting from the first Segment or Subsegment having earliest presentation time greater than that of the last requested Segment or Subsegment of the switch-from MSR.
   1. If in the associated ESR there is a Segment having the same earliest presentation time as the starting Segment or Subsegment in the switch-to MSR, that ESR Segment is also requested, preferably before requesting of the starting Segment or Subsegment in the switch-to MSR. Otherwise, no Segment of the associated ESR is requested.
4. When continuously requesting and consuming subsequent Segments or Subsegments of an MSR after session initialization, seeking, or stream switching, no Segment of the associated ESR needs to be requested, including when requesting any subsequent MSR Segment or Subsegment starting with an EDRAP sample.

As can be seen from the above example client operations, the client needs to calculate the earliest presentation times of the MSR Segments and Subsegments as well as of the ESR Segments to figure out whether an MSR Segment or Subsegment has an associated ESR Segment.

# Change 2: Picture-in-picture support

*Add subclause 5.8.5.16 as follows:*

***5.8.5.16 Picture-in-picture descriptor***

A **SupplementalProperty** element with the @schemeIdUri attribute equal to urn:mpeg:dash:pinp:2022 is referred to as a picture-in-picture (PiP) descriptor.

At most one PiP descriptor may be present at Preselection level. The presence of a PiP descriptor in a Preselection indicates that the purpose of the Preselection is for providing a PiP experience.

PiP services offer the ability to include a video with a smaller spatial resolution within a video with a bigger spatial resolution. In this case, the different bitstreams/Representations of the main video are included in the Main Adaptation Set of the Preselection, and the different bitstreams/Representations of a supplementary video, also referred to as PiP video, are included a Partial Adaptation Set of the Preselection.

When a PiP descriptor is present in a Preselection, and the **picInPicInfo**@dataUnitsReplacable attribute is present and equal to 'true', the client may choose to replace the coded video data units representing the target PiP region in the main video with the corresponding coded video data units of the PiP video before sending to the video decoder. This way, separate decoding of the main video and the PiP video can be avoided. For a particular picture in the main video, the corresponding video data units of the PiP video are all the coded video data units in the decoding-time-synchronized sample in the supplemental video Representation.

The @value attribute of the PiP descriptor shall not be present. The PiP descriptor shall include a **picInPicInfo** element with its attributes as specified in Table 37.

**Table 37 — Semantics of picInPicInfo element**

| **Element or Attribute Name** | | | | **Use** | **Description** |
| --- | --- | --- | --- | --- | --- |
|  |  | **PicInpicInfo** | |  |  |
|  |  |  | @dataUnitsReplacable | OD  Default: 'false' | specifies whether the coded video data units representing the target PiP region in the main video can be replaced by the corresponding video data units of the PiP video.  When @dataUnitsReplacable is equal to 'true', the client may choose to replace the coded video data units representing the target PiP region in the main video with the corresponding coded video data units of the PiP video before sending to the video decoder for decoding. |
|  |  |  | @regionIds | O | specifies the IDs of the coded video data units representing the target PiP region, as a white space separated list.  When @dataUnitsReplacable is equal to 'false', this attribute shall not be present.  The concrete semantics of the region IDs need to be explicitly specified for specific video codecs. |
| **Key**  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 an @. | | | | | |

The XML syntax of the **PicInpicInfo** element is as follows:

<xs:complexType name="PicInpicInfoType">

<xs:annotation>

<xs:documentation xml:lang="en">

**Picnpic**

</xs:documentation>

</xs:annotation>

<xs:attribute name="dataUnitReplacable" type="xs:boolean" default="false" />

<xs:attribute name="regionIds" type="StringNoWhitespaceType"/>

<xs:anyAttribute namespace="##other" processContents="lax"/>

</xs:complexType>

Editors Note: An alternative solution for picture-in-picture support in DASH is to use the 'pip' value for **Role**, as well as to use **ContentComponent** along with **Role** and @tag to signal the subpicture IDs or any other IDs needed, as described in m58924, for which some discussions of the details are available herein: <http://mpegx.int-evry.fr/software/MPEG/Systems/DASH/spec/-/issues/265>.

Editor’s Notes:

1. The manipulation of the stream and the composition is out of the scope of the dash client.
2. the dash client provides the content properties and metadata to the application. It is the application's job to do any bitstream manipulation, location of pip and rendering.

# Change 3: Adding status to Event

*Replace Tables 38 and 39 with the following tables:*

Table 38 — Event Stream Semantics

| **Element or Attribute Name** | | | **Use** | **Description** |
| --- | --- | --- | --- | --- |
|  | EventStream | |  | specifies event Stream |
|  |  | @xlink:href | O | specifies a reference to an external EventStream element |
|  |  | @xlink:actuate | OD  default: onRequest | specifies the processing instructions, which can be either "onLoad" or "onRequest".  This attribute shall not be present if the @xlink:href attribute is not present. |
|  |  | @schemeIdUri | M | identifies the message scheme. The string may use URN or URL syntax. When a URL is used, it is recommended to also contain a month-date in the form mmyyyy; the assignment of the URL must have been authorized by the owner of the domain name in that URL on or very close to that date. A URL may resolve to an Internet location, and a location that does resolve may store a specification of the message scheme. |
|  |  | @value | O | specifies the value for the event stream element. The value space and semantics must be defined by the owners of the scheme identified in the @schemeIdUri attribute. |
|  |  | @timescale | O | specifies the timescale in units per seconds to be used for the derivation of different real-time duration values in the Event elements.  If not present on any level, it shall be set to 1. |
|  |  | @presentationTimeOffset | OD  Default: 0 | specifies the presentation time offset of this Event Stream that aligns with the start of the Period. Any Event contained in this Event Stream is mapped to the Period timeline by using the Event presentation time subtracted by the value of the presentation time offset.  This adjustment shall not be applied to Inband event message streams..  The value of the presentation time offset in seconds is the division of the value of this attribute and the value of the @timescale attribute. |
|  |  | Event | 0 ... N | specifies one event. For details see Table 39.  Events in Event Streams shall be ordered such that their presentation time is non-decreasing. |
| **Key**  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 an @. | | | | |

Table 39 — Event Semantics

| **Element or Attribute Name** | | | | | **Use** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Event | |  | specifies an Event and contains the message of the event. The content of this element depends on the event scheme. The contents shall be either:   * A string, optionally encoded as specified by @contentEncoding * XML content using elements external to the MPD namespace   For new event schemes string content should be used, making use of Base 64 encoding if needed.  NOTE The schema allows “mixed” content within this element however only string data or XML elements are permitted by the above options, not a combination. |
|  |  |  |  | @presentationTime | OD default: 0 | specifies the presentation time of the event relative to the start of the Period taking into account the @presentationTimeOffset of the Event Stream, if present.  The value of the presentation time in seconds is the division of the value of this attribute and the value of the @timescale attribute.  If not present, the value of the presentation time is 0. |
|  |  |  |  | @duration | O | specifies the presentation duration of the Event.  The value of the duration in seconds is the division of the value of this attribute and the value of the  @timescale attribute.  The interpretation of the value of this attribute is defined by the scheme owner.  If not present, the value of the duration is unknown. |
|  |  |  |  | @id | O | specifies an identifier for this instance of the event. Events with equivalent content and attribute values in the Event element shall have the same value for this attribute.  The scope of the @id for each Event is with the same @schemeIdURI and @value pair. |
|  |  |  |  | @status | O  default: none | specifies the status of event:   * none: no specific status * update: the event is an update of another event with identical values of @schemeIdUr, @value, and @id |
|  |  |  |  | @contentEncoding | O | specifies whether the information in the body and the information in the @messageData is encoded.  If present, the following value is possible:   * base64 the content is encoded as described in IETF RFC 4648 prior to adding it to the field.   If this attribute is present, the DASH Client is expected to decode the message data and only provide the decoded message to the application. |
|  |  |  |  | @messageData | O | specifies the value for the event stream element. The value space and semantics must be defined by the owners of the scheme identified in the @schemeIdUri attribute.  NOTE the use of the message data is discouraged by content authors, it is only maintained for the purpose of backward-compatibility. Including the message in the Event element is recommended in preference to using this attribute. This attribute is expected to be deprecated in the future editions of this document. |
| **Key**  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 an @. | | | | | | |

An **Event** with @status=’update’ is the updated instance of an event with identical @schemeIdUri, @value, and @id attributes that may have been previously processed by the DASH client. The DASH client may replace the previous event with the updated instance if the previous event has not been dispatched yet. An Event with @status=’update’ may differ from the previous event except in the following attributes: @schemeIdUri, @value, and @id.

*Replace 5.10 with the following:*

<xs:complexType name="EventStreamType">

<xs:annotation>

<xs:documentation xml:lang="en">

**Event Stream**

</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="Event" type="EventType" minOccurs="0" maxOccurs="unbounded"/>

<xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

<xs:attribute ref="xlink:href"/>

<xs:attribute ref="xlink:actuate" default="onRequest"/>

<xs:attribute ref="xlink:type" fixed="simple"/>

<xs:attribute ref="xlink:show" fixed="embed"/>

<xs:attribute name="schemeIdUri" type="xs:anyURI" use="required"/>

<xs:attribute name="value" type="xs:string"/>

<xs:attribute name="timescale" type="xs:unsignedInt"/>

<xs:attribute name="presentationTimeOffset" type="xs:unsignedLong" default="0"/>

</xs:complexType>

<xs:complexType name="EventType" mixed="true">

<xs:annotation>

<xs:documentation xml:lang="en">

**Event**

</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

<xs:attribute name="presentationTime" type="xs:unsignedLong" default="0"/>

<xs:attribute name="duration" type="xs:unsignedLong"/>

<xs:attribute name="id" type="xs:unsignedInt"/>

<xs:attribute name="status" type="xs:StatusType"/>

<xs:attribute name="contentEncoding" type="ContentEncodingType"/>

<xs:attribute name="messageData" type="xs:string">

<xs:annotation>

<xs:documentation xml:lang="en">

**Deprecated in favor of carrying the message information in the**

**value space of the event**

</xs:documentation>

</xs:annotation>

</xs:attribute>

<xs:anyAttribute namespace="##other" processContents="lax"/>

</xs:complexType>

<xs:simpleType name="StatusType">

<xs:annotation>

<xs:documentation xml:lang="en">

**Event Status**

</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:enumeration value="none"/>

<xs:enumeration value="update"/>

</xs:restriction>

</xs:simpleType>

<xs:simpleType name="ContentEncodingType">

<xs:annotation>

<xs:documentation xml:lang="en">

**Event Coding**

</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:enumeration value="base64"/>

</xs:restriction>

</xs:simpleType>

*Replace 5.10.3.3.3 with the following:*

##### Syntax

aligned(8) class DASHEventMessageBox extends FullBox('emsg', version, flags){

if (version==0) {

string scheme\_id\_uri;

string value;

unsigned int(32) timescale;

unsigned int(32) presentation\_time\_delta;

unsigned int(32) event\_duration;

unsigned int(32) id;

} else if (version==1) {

unsigned int(32) timescale;

unsigned int(64) presentation\_time;

unsigned int(32) event\_duration;

unsigned int(32) id;

string scheme\_id\_uri;

string value;

}

unsigned int(8) message\_data[];

}

*Replace 5.10.3.3.4 with the following:*

— scheme\_id\_uri: is a null-terminated ('C') string in UTF-8 characters that identifies the message scheme. The semantics and syntax of the message\_data[] are defined by the owner of the scheme identified. The string may use URN or URL syntax. When a URL is used, it is recommended to also contain a month-date in the form mmyyyy; the assignment of the URL must have been authorized by the owner of the domain name in that URL on or very close to that date. A URL may resolve to an Internet location, and a location that does resolve may store a specification of the message scheme.

— value: is a null-terminated ('C') string in UTF-8 characters that specifies the value for the event. The value space and semantics must be defined by the owners of the scheme identified in the scheme\_id\_uri field.

— timescale provides the timescale, in ticks per second, for the event duration and presentation\_time\_delta or presentation\_time fields. The value should be identical to the timescale of a track contained in the carrying Segment. Furthermore, the value should be identical for all events in one Event Stream.

— presentation\_time\_delta provides the Media Presentation time delta of the media presentation time of the event and the earliest presentation time in this segment. If the segment index is present, then the earliest presentation time is determined by the field earliest\_presentation\_time of the first 'sidx' box. If the segment index is not present, the earliest presentation time is determined as the earliest presentation time of any access unit in the media segment. The timescale is provided in the timescale field.

— presentation\_time provides the Media Presentation time of the event measured on the Movie timeline, in the timescale provided in the timescale field, and adjusted by **InbandEventStream**@presentationTimeOffset, in the time scale provided by **InbandEventStream**@timescale; the value shall not be less than the earliest presentation time of the carrying Segment.

— event\_duration provides the duration of event in media presentation time. The timescale is indicated in the timescale field. The value 0xFFFFFFFF indicates an unknown duration. The interpretation of this value must be defined by the owner of the event scheme.

— id: a field identifying this instance of the message. The scope of this identifier for each event is with the same scheme\_id\_uri and value pair. Messages with the same id within the scope of the same scheme\_id\_uri and value pair are equivalent , i.e. processing of any one event message box with the same id is sufficient.

— message\_data: body of the message, which fills the remainder of the message box. This may be empty depending on the above information. The syntax and semantics of this field must be defined by the owner of the scheme identified in the scheme\_id\_uri field.

The flags field is specified as follows:

* (flags & 1) equal to 1 indicates that the esmg is an update of another esmg with identical values of scheme\_id\_uri, value and id fields.

An emsg box with flags &1 =1 is the updated instance of an emsg box with identical scheme\_id\_uri, value, and id fields that may have been previously processed by the DASH client. The DASH client may replace the previous event with the updated instance if the previous event has not been dispatched yet. The updated emsg may differ from the previous emsg except in the following fields: scheme\_id\_uri, value, and id.

*Change A.13.8:*

It is assumed that the application is subscribed to a specific event stream identified by a (scheme/value) pair with a specific dispatch\_mode, either on-start or on-receive, as described in subclause A.13.7.

The processing model varies depending on the value of dispatch\_mode.

1. Common process
   1. The DASH Client implements the following process:
      1. The DASH Client sets up a Pending Event Table (PET) for each subscribed scheme\_uri/(value) in the case of dispatch\_mode = on\_start. The PET maintains a single list of event ids that are waiting to be dispatched. The DASH Client also sets up a Dispatched Event Table (DET) for each subscribed scheme\_uri/(value). The DET maintains a single list of 'emsg' ids that have been dispatched.
      2. Parse the 'emsg'/timed metadata sample and retrieve scheme\_uri/(value).
      3. If Application is not subscribed to the scheme\_uri/(value) pair, end the processing of this 'emsg'.
      4. Derive the event instance/metadata sample’s *ST*
      5. Derive the ending time *ET*= *ST* + *DU*.
2. On-receive processing
   1. The DASH Client implements the following process when dispatch\_mode = on\_receive:
      1. If the current presentation time value is greater than *ET*, then end processing.
      2. In the case of event: Compare the event’s *id* with the entries of the DET of the same scheme\_uri/(*value)* pair:
         * If an entry with the identical *id* value exists, end processing;
      3. Dispatch the event/timed metadata, including *ST*, *id*, *DU*, *timescale*, and message\_data as described in subclause A.13.6, and add the event to the DET.
3. on-start processing
   1. DASH ClientThe DASH Client implements the following process when dispatch\_mode = on\_start:
      1. If the event is an update of a previous event (signalled through @status or emsg flags), remove any existing event, if any, with identical scheme\_uri/(*value)* and *id* from the PET.
      2. Derive the event instance/metadata sample’s *ST*.
      3. If the current media presentation time value is smaller than *ST*, then go to step vi.
      4. Derive the ending time *ET*= *ST* + *DU*.
      5. If the current presentation time value is greater than *ET*, then end the processing.
      6. In the case of event: Compare the event’s *id* with the entries of the PET of the same scheme\_uri/(*value)* pair:
         * If an entry with the identical *id* value exists, end the processing.
         * If not, add 'emsg'’s *id* to the corresponding P[ET](#active-event-table).
      7. Dispatch the event/metadata message\_data at time ST, or immediately if the current presentation time is larger than ST, as described in subclause A.13.6, remove the event, if any exists, from the PET and add it to the DET.

# Change 4: Adding DASH specific events dispatch modes

*Change A.13.10 to:*

**A.13.10 Prose description of APIs**

The event/timed metadata API is an interface defined between a DASH client and a device application in the exchange of subscription data and dispatch/transfer of matching DASH Event or timed metadata information between these entities. The Event/timed metadata API is shown in Figure A.1.

NOTE 1 In this Annex, the term "DASH Player" is used.

The description of the API below is strictly functional, i.e. implementation-agnostic. For example, the subscribeEvent() method as defined below may be mapped to the existing on(type,listener,scope) method as defined for the dash.js under MediaPlayerEvents.

As part of this API and before any operations, the DASH Player provides a list of *scheme\_id*/(*value*) listed in the MPD when it receives it. This list includes all MPD and inband events as well as *scheme\_id* of all timed metadata tracks. At this point, the Application is aware of the possible events and metadata deliverable by the DASH Player.

NOTE 2 The DASH Player may provide the Application the list of DASH event schemes as a part of listed event schemes in the MPD and consequently, the Application may subscribe to one or more of these event schemes.

Change A.13.12 to the following:

**A.13.12 Dispatch modes for DASH-specific events**

In 5.10.4, several DASH-specific event schemes are defined. Table A.4 describes their dispatch modes.

**Change 4: Fixing the lang authoring and processing**

* + - 1. **Semantics**

**Table 5 — Semantics of** AdaptationSet **element**

| **Element or Attribute Name** | | | | **Use** | **Description** |
| --- | --- | --- | --- | --- | --- |
|  |  | **…** | |  |  |
|  |  |  | @lang | O | Declares the language code for this Adaptation Set. The syntax and semantics according to IETF ~~RFC 5646~~ BCP 47 shall be used.  Other subtags except the primary language subtag, e.g. region subtags such as used in “es-US”, should not be used unless essential in language disambiguation across an Adaptation Set..  Conformance tool should check the conformance of language according to 2.If not present, the language code may be defined for each media component or it may be unknown.  If the language is unknown, the 'und' code for undetermined primary language or the 'zxx' (Non-Linguistic, Not Applicable) code can be used.  Note 1: IETF BCP 47 is the combination of IETF RFC 5646 and IETF RFC 4647.  Note 2: Per IETF BCP 47, 2-character codes are to be used whenever possible, i. e. 3-character codes are not to be used when there is an equivalent 2-character code. |
|  |  |  | … |  |  |

…

* + - 1. **Overview**

…

The @lang attribute assigned to a label describes the language of that label. For example, the @lang attribute can be used by the DASH Client to filter labels according to user language preferences. When comparing language tag values of @lang, IETF BCP 47 shall be applied.

…

* + - 1. **Semantics**

**Table 24 — Semantics of** Label **element**

| **Element or Attribute Name** | | **Use** | **Description** |
| --- | --- | --- | --- |
| **Label** | |  | The label and the actual text of the label that annotates the element in the DASH Media Presentation |
|  | @id | OD default: 0 | An identifier for the label. Labels with the same value for this attribute belong to a label group. |
|  | @lang | O | Specifies the language of the label. If not present, the language of the label is unknown. See AdaptationSet@lang for more details. |

**Table 25 — Semantics of** GroupLabel **element**

| **Element or Attribute Name** | | **Use** | **Description** |
| --- | --- | --- | --- |
| **GroupLabel** | |  | The actual text of the group label that annotates the element in the DASH Media Presentation |
|  | @id | OD default: 0 | An identifier for the group label. The value for this attribute shall be unique for all group labels in the MPD. The value of this attribute shall be the same as the id attribute of all Label elements in the label group. |
|  | @lang | O | Specifies the language of the label. If not present, the language of the label is unknown. See AdaptationSet@lang for more details. |

…

* + 1. **Semantics**

**Table 31 — Program information semantics**

| **Element or Attribute Name** | | | **Use** | **Description** |
| --- | --- | --- | --- | --- |
|  | **ProgramInformation** | |  | specifies descriptive information about the program |
|  |  | @lang | O | Declares the language code(s) for this Program Information. The syntax and semantics according to ~~IETF RFC 5646~~ IETF BCP 47 shall be applied. See AdaptationSet@lang for more details.  If not present, the value is unknown. |

…

* + - 1. **Adaptation Set Constraints**

If the *DASH profile for CMAF content constraints* apply to an Adaptation Set, then the following holds for this Adaptation Set:

…

@lang shall set as follows:

To the extent that any ISOBMFF fields below do not conform to IETF BCP 47, they shall be converted before populating @lang. See AdaptationSet@lang for more details. [Ed. (MD): Further study is needed – the RFC has some text on this in section 4.1, item 5. Additionally, this intersects with “use” being set to “optional”.]

If an ExtendedLanguageBox (elng) in the MediaHeaderBox (mdia) of the CMAF Principal Header is present in the TrackBox, then it is set of the content of the extended\_language field

1. Table A.4 DASH-specific event schemes dispatch modes

| **scheme** | **Dispatch mode** |
| --- | --- |
| urn:mpeg:dash:event:2012 | on-receive |
| urn:mpeg:dash:event:callback:2015 | on-start |
| urn:mpeg:dash:event:ttfn:2016 | on-start |
| urn:mpeg:dash:event:period:2020 | On-receive |

# Change 5: Content Steering

*In clause 2, add the following reference:*

*ETSI TS 1XX XXX, DASH-IF: Content Steering for DASH*

NOTE: This specification is available as draft pre-publication as a community review version of DASH-IF CTS XX and can be accessed here (pdf)

*In subclause 5.2.3.5, change the title from:*

Elements and Attributes added in the fifth edition (ISO/IEC 23009-1:2021)

*to be the following:*

Elements and Attributes added in the fifth edition (ISO/IEC 23009-1:2022)

*Add subclause 5.2.3.6 as follows:*

**5.2.3.6 Elements and Attributes added in the sixth edition (ISO/IEC 23009-1:2022)**

This revision adds the following elements and attributes to the schema defined in Annex B compared to the 2022 revision (ISO/IEC 23009-1:2022) of this document:

— **MPD.ContentSteering**

— **Location**@serviceLocation

— **PatchLocation**@serviceLocation

*In subclause 5.3.1.2, replace Table 3 with the following:*

**Table 3 — Semantics of MPD element**

| **Element or Attribute Name** | | | | | | **Use** | **Description** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **MPD** | | | | | |  | The root element that carries the Media Presentation Description for a Media Presentation. |
|  | | | @id | | | O | specifies an identifier for the Media Presentation. It is recommended to use an identifier that is unique within the scope in which the Media Presentation is published.  If not specified, no MPD-internal identifier is provided. However, for example the URL to the MPD may be used as an identifier for the Media Presentation. |
|  | | | @profiles | | | M | specifies a list of Media Presentation profiles as described in 8.  The contents of this attribute shall conform to either the pro-simple or pro-fancy productions of IETF RFC 6381:2011, Section 4.5, without the enclosing DQUOTE characters, i.e. including only the unencodedv or encodedv elements respectively.  As profile identifier a restricted URI format as defined in 8.1 shall be used. |
|  | | | @type | | | OD  default: static | specifies the type of the Media Presentation. For static Media Presentations (@type="static"), all Segments are available between the @availabilityStartTime and the @availabilityEndTime. For dynamic Media Presentations (@type="dynamic"), Segments typically have different availability times. For details, refer to subclause 5.3.9.5.3.  In addition, the Media Presentation Description may be updated in dynamic Media Presentations, i.e. the  @minimumUpdatePeriod may be present.  NOTE   Static Media Presentations are typically used for On-Demand services, whereas dynamic Media Presentations are used for live services. |
|  | | | @availabilityStartTime | | | CM  shall be present for @type='dynamic' | For @type='dynamic', this attribute shall be present. In this case, it specifies the anchor for the computation of the earliest availability time (in UTC) for any Segment in the Media Presentation.  For @type=static” if present, it specifies the Segment availability start time for all Segments referred to in this MPD. If not present, all Segments described in the MPD shall become available at the time the MPD becomes available. |
|  | | | @publishTime | | | OD  shall be present for @type=′dynamic′ | specifies the wall-clock time when the MPD was generated and published at the origin server. MPDs with a later value of @publishTime shall be an update as defined in subclause 5.4 to MPDs with earlier @publishTime. |
|  | | | @availabilityEndTime | | | O | specifies the latest Segment availability end time for any Segment in the Media Presentation. When not present, the value is unknown. |
|  | | | @mediaPresentationDuration | | | O | specifies the duration of the entire Media Presentation. If the attribute is not present, the duration of the Media Presentation is unknown.  This attribute shall be present when neither the attribute **MPD**@minimumUpdatePeriod nor the **Period**@duration of the last Period are present. |
|  | | | @minimumUpdatePeriod | | | O | If this attribute is present, it specifies the smallest period between potential changes to the MPD. This can be useful to control the frequency at which a client checks for updates.  From a client perspective, after a client fetches an MPD, it specifies the minimum period during which the MPD remains valid. Validity is defined in subclause 5.4.  If this attribute is not present, it indicates that the MPD does not change.  If **MPD**@type is not 'dynamic', @minimumUpdatePeriod shall not be present.  Details on the use of the value of this attribute are specified in subclause 5.4. |
|  | | | @minBufferTime | | | M | specifies a common duration used in the definition of the Representation data rate (see @bandwidth attribute in subclauses 5.3.5.2 and 5.3.5.4). |
|  | | | @timeShiftBufferDepth | | | O | specifies the duration of the smallest time shifting buffer for any Representation in the MPD that is guaranteed to be available for a Media Presentation with type 'dynamic'. When not present, the value is infinite. This value of the attribute is undefined if the type attribute is equal to 'static'. |
|  | | | @suggestedPresentationDelay | | | O | When @type is 'dynamic', it specifies a fixed delay offset in time from the presentation time of each access unit that is suggested to be used for presentation of each access unit. For more details, refer to subclause 7.2.1. When not specified, then no value is provided and the client is expected to choose a suitable value.  When @type is 'static'the value of the attribute is undefined and may be ignored. |
|  | | | @maxSegmentDuration | | | O | specifies the maximum duration of any Segment in any Representation in the Media Presentation, i.e. documented in this MPD and any future update of the MPD. If not present, then the maximum Segment duration shall be the maximum duration of any Segment documented in this MPD. |
|  | | | @maxSubsegmentDuration | | | O | specifies the maximum duration of any Media Subsegment in any Representation in the Media Presentation. If not present, the same value as for the maximum Segment duration is implied. |
|  | | | **ProgramInformation** | | | 0…N | specifies descriptive information about the program. For more details, refer to the description in subclause 5.7. |
|  | | | **BaseURL** | | | 0…N | specifies a Base URL that can be used for reference resolution and alternative URL selection. For more details, refer to the description in subclause 5.6. |
|  | | | **Location** | | | 0…N | specifies a location at which the MPD is available.  A reference processing model is provided in Annex A.11. |
|  | | |  | | @serviceLocation | O | This attribute specifies a relationship between URLs sharing the same value for this attribute.  For more details refer to subclause 5.6.6. |
|  | | | **PatchLocation** | | | 0 ... N | specifies a location at which the MPD patch document is available. Details on the MPD patch document, this element, and expected processing models are available in subclause 5.15.  If this element is present, the **MPD**@id attribute and the **MPD**@publishTime shall be present.  When @type is 'static'or the @minimumUpdatePeriod attribute is not present, then value of the element is undefined and may be ignored.  If this element is not present, no MPD patch document is available.  If multiple elements are present, any **PatchLocation** element may be used. |
|  | | **ServiceDescription** | | | | 0 ... N | specifies the service description detailing how the service provider expects the service is to be consumed.  It shall provide a valid service description as described by Annex K.3 and Annex K.4.  For details refer to Annex K. |
|  | **InitializationSet** | | | | | 0 ... N | specifies a suitable initialization for a specific media type for the presentation. If present, at least one Period of the Media Presentation shall include at least one Adaptation Set that can be played when initialized by this Initialization Set.  For details see subclause 5.3.12. |
|  | **InitializationGroup** | | | | | 0 ... N | Specifies a white space separated list of ids of Initialization Sets of the same content type. This indicates that any Period in the Media Presentation has at least one Adaptation Set that conforms to one of the Initialization Sets referenced in this element.  For details see subclause 5.3.12. |
|  |  | | | @id | | M | specifies an unsigned integer identifier for this Initialization Group. The attribute shall be unique unique among all Initialization Set and Initialization Group ids in the MPD. |
|  |  | | | @profiles | | O | specifies the profiles which the listed Initialization Groups conform to. The value shall be a subset of the value in the **MPD**@profiles attribute.  If not present the value is inferred to be the same as the **MPD**@profiles attribute.  The same syntax is used as defined for the **MPD**@profiles attribute. |
|  |  | | | @contentType | | O | specifies the content type of Initialization Sets listed in the element. |
|  | **InitializationPresentation** | | | | | 0 ... N | specifies a white space separated list of ids of Initialization Sets and Initialization Groups to indicate a combination which creates a complete presentation. A client supporting all listed Initialization Sets and Initialization Groups of an Initialization Presentation is expected to be able to play the entire Media Presentation as intended by the service provider.  For details see subclause 5.3.12. |
|  |  | | | @id | | M | specifies a unique unsigned integer identifier for this Initialization Presentation. The attribute shall be unique among all Initialization Presentation ids in the MPD. |
|  |  | | | @profiles | | O | specifies the profiles which the listed Initialization Groups conform to. The value shall be a subset of the value in the **MPD**@profiles attribute.  If not present the value is inferred to be the same as the **MPD**@profiles attribute.  The same syntax is used as defined for the **MPD**@profiles attribute. |
|  |  | | | @contentType | | O | specifies the content type of initialization Groups listed in the element. |
|  | | | **ContentProtection** | | | 0 … N | specifies information about content protection and encryption schemes used in this Media Presentation. If present on this level, it shall include the @refId attribute.  For details, see subclauses 5.8.1 and 5.8.4.1. |
|  | | | **Period** | | | 1…N | specifies the information of a Period. For more details, refer to the description in subclause 5.3.2. |
|  | | | **Metrics** | | | 0 ... N | specifies the DASH Metrics.  For more details, see subclause 5.9. |
|  | | **EssentialProperty** | | | | 0 … N | specifies information about the containing element that is considered essential by the Media Presentation author for processing the containing element.  For details, see subclause 5.8.4.8. |
|  | | **SupplementalProperty** | | | | 0 … N | specifies supplemental information about the containing element that may be used by the DASH Client optimizing the processing.  For details, see subclause 5.8.4.9. |
|  | | **UTCTiming** | | | | 0 ... N | specifies information on ways to obtain a synchronization to wall-clock time as used in this Media Presentation. The order of the elements expresses a preference of choice by the Media Presentation author. For more details, refer to subclause 5.8.4.11. |
|  | | **LeapSecondInformation** | | | | 0 ... 1 | specifies leap second information affecting MPD timing calculations.  For details refer to subclause 5.13. |
|  | | **ContentSteering** | | | | 0 … 1 | A URL that can be used to access the Content Steering server. The URL points to a DASH Content Steering Manifest (DCSM) as defined in ETSI TS 1XX XXX. [Ed. (TS): This integration goes directly into the MPD. Preferred is usage of Service Description, see Annex K.] |
| NOTE: This option adds Content Steering directly to the MPD. This is one option, but not the preferred. Preferred is usage of Service Description, see Annex K. NBs are asked to comments whether to support one of the two options or both options. | | | | | | | |
| **Key**  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 an @. | | | | | | | |

*In subclause 5.3.1.3, replace the XML syntax with the following:*

<xs:complexType name="MPDtype">

<xs:annotation>

<xs:documentation xml:lang="en">

**MPD Type**

</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="ProgramInformation" type="ProgramInformationType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="BaseURL" type="BaseURLType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="Location" type="LocationType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="PatchLocation" type="PatchLocationType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="ServiceDescription" type="ServiceDescriptionType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="InitializationSet" type="InitializationSetType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="InitializationGroup" type="UIntVWithIDType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="InitializationPresentation" type="UIntVWithIDType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="ContentProtection" type="ContentProtectionType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="Period" type="PeriodType" maxOccurs="unbounded"/>

<xs:element name="Metrics" type="MetricsType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="EssentialProperty" type="DescriptorType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="SupplementalProperty" type="DescriptorType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="UTCTiming" type="DescriptorType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="LeapSecondInformation" type="LeapSecondInformationType" minOccurs="0"/>

<xs:element name="ContentSteering" type="ContentSteeringType" minOccurs="0" maxOccurs="unbounded"/> [Ed. (TS): This is the less preferred option, better to keep it in Service Description.]

<xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

<xs:attribute name="id" type="xs:string"/>

<xs:attribute name="profiles" type="ListOfProfilesType" use="required"/>

<xs:attribute name="type" type="PresentationType" default="static"/>

<xs:attribute name="availabilityStartTime" type="xs:dateTime"/>

<xs:attribute name="availabilityEndTime" type="xs:dateTime"/>

<xs:attribute name="publishTime" type="xs:dateTime"/>

<xs:attribute name="mediaPresentationDuration" type="xs:duration"/>

<xs:attribute name="minimumUpdatePeriod" type="xs:duration"/>

<xs:attribute name="minBufferTime" type="xs:duration" use="required"/>

<xs:attribute name="timeShiftBufferDepth" type="xs:duration"/>

<xs:attribute name="suggestedPresentationDelay" type="xs:duration"/>

<xs:attribute name="maxSegmentDuration" type="xs:duration"/>

<xs:attribute name="maxSubsegmentDuration" type="xs:duration"/>

<xs:anyAttribute namespace="##other" processContents="lax"/>

</xs:complexType>

<xs:complexType name="LocationType">

<xs:simpleContent>

<xs:extension base="xs:anyURI">

<xs:attribute name="serviceLocation" type="xs:string"/>

<xs:anyAttribute namespace="##other" processContents="lax"/>

</xs:extension>

</xs:simpleContent>

</xs:complexType>

<xs:simpleType name="PresentationType">

<xs:annotation>

<xs:documentation xml:lang="en">

**Presentation Type enumeration**

</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:enumeration value="static"/>

<xs:enumeration value="dynamic"/>

</xs:restriction>

</xs:simpleType>

<xs:complexType name="UIntVWithIDType">

<xs:annotation>

<xs:documentation xml:lang="en">

**UInt Vector With ID**

</xs:documentation>

</xs:annotation>

<xs:simpleContent>

<xs:extension base="UIntVectorType">

<xs:attribute name="id" type="xs:unsignedInt" use="required"/>

<xs:attribute name="profiles" type="ListOfProfilesType"/>

<xs:attribute name="contentType" type="RFC6838ContentTypeType"/>

<xs:anyAttribute namespace="##other" processContents="lax"/>

</xs:extension>

</xs:simpleContent>

</xs:complexType>

<xs:simpleType name="UIntVectorType">

<xs:annotation>

<xs:documentation xml:lang="en">

**Whitespace-separated list of unsigned integers**

</xs:documentation>

</xs:annotation>

<xs:list itemType="xs:unsignedInt"/>

</xs:simpleType>

<xs:simpleType name="ListOfProfilesType">

<xs:annotation>

<xs:documentation xml:lang="en">

**List of Profiles**

</xs:documentation>

</xs:annotation>

<xs:restriction base="xs:string">

<xs:pattern value="(&URN;|&URL;)(&comma\_sep;(&URN;|&URL;))\*"/>

</xs:restriction>

</xs:simpleType>

*In subclause 5.6.2, replace Table 30 with the following:*

**Table 30 — Semantics of BaseURL element**

| **Element or Attribute Name** | | | **Use** | **Description** |
| --- | --- | --- | --- | --- |
|  | **BaseURL** | |  | A URL that can be used as Base URL. The content of this element is a URI string as described in subclause 5.6.4. |
|  |  | @serviceLocation | O | This attribute specifies a relationship between URLs sharing the same value for this attribute.  For more details refer to subclause 5.6.6. |
|  |  | @byteRange | O | If present, specifies HTTP partial GET requests may alternatively be issued by adding the byte range into a regular HTTP-URL based on the value of this attribute and the construction rules in E.2.  If not present, HTTP partial GET requests may not be converted into regular GET requests.  NOTE   Such alternative requests are expected to not be used unless the DASH application requires this. For more details, refer to Annex E. |
|  |  | @availabilityTimeOffset | O | specifies an offset to define the adjusted segment availability time. For semantics, refer to Table 14.  If the value is present in **SegmentBase** then this attribute is additive to the one in **SegmentBase**. For details on processing, refer to 5.3.9.5.3. |
|  |  | @availabilityTimeComplete | O | specifies if all Segments of all associated Representation are complete at the adjusted availability start time. For semantics, refer to Table 14.  If the value is present in **SegmentBase** then this attribute should not be present. If present in **SegmentBase** and **BaseURL**, the value in **BaseURL** shall be ignored. |
|  |  | @timeShiftBufferDepth | O | specifies the duration of the smallest time shifting buffer for any Representation in the MPD that is guaranteed to be available for a Media Presentation with type 'dynamic'.  This value overrides **MPD**@timeShiftBufferDepth for the resources that use this BaseURL.  This value of the attribute is undefined if the **MPD**@type attribute is equal to 'static'. |
|  |  | @rangeAccess | OD  Default: FALSE | If set to true, partially available Segments may be accessed with byte range request. If a client is making a byte-range request against a partially available Segment and the first-byte position of that range request is non-zero and the client is expecting an aggregating response, then the client should signal that expectation which shall follow the convention of IETF RFC 8673. Specifically, it should use a last-byte value of 9007199254740991. This will signal the server to respond with a 206 aggregating response instead of waiting for the end of the Segment and responding with a 200 response code and a content-length response header.  If set to false, the client should not expect a response corresponding to the requested byte range. |
| **Key**  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 an @. | | | | |

*Add subclause 5.6.6 as follows:*

**5.6.6 Service location**

Several elements of type xs:anyURI may have associated a @serviceLocation attribute. If two resources share the same value for this attribute, then these URLs are likely to have their URLs resolve to services at a common network location, for example a common Content Delivery Network.

If the element does not include a @serviceLocation attribute, no relationship to any resource in the MPD is known.

The string value of the @serviceLocation should only contain characters from the set [a..z], [A..Z], [0..9], **'.'**, '-', and '\_'.

A client may for example use such information in order to correlate network statistics from the collected statistics when resolving to a URL at the same service location to predict the behaviour when resolving for another resource at the same service location.

Service locations may for example be used to annotate redundant content offerings. In this case, for example a content steering operation may use the values of service locations to steer the client towards a specific version of the redundant content offering.

*In subclause 5.15.2, replace Tables 48 and 49 with the following:*

**Table 48 — Semantics of PatchLocation element in MPD**

| **Element or Attribute Name** | | **Use** | **Description** |
| --- | --- | --- | --- |
| **PatchLocation** | |  | specifies a location at which the MPD patch is available. The referenced document shall conform to an MPD patch document as defined in subclause 5.15.3. |
|  | @serviceLocation | O | This attribute specifies a relationship between URLs sharing the same value for this attribute.  For more details refer to subclause 5.6.6. |
|  | @ttl | O | specifies the time period duration (in seconds) starting from **MPD**@publishTime until the MPD patch document is at least available at the indicated location above. For details refer to the processing model in subclause 5.14.4.  If not present, the value is unknown. |
| **Key**  For attributes: M=mandatory, O=optional  Elements are **bold**; attributes are non-bold and preceded with an @. | | | |

**Table 49 — XML Syntax of PatchLocation element in MPD**

<xs:complexType name="PatchLocationType">

<xs:annotation>

<xs:documentation xml:lang="en">

**Patch Location Type**

</xs:documentation>

</xs:annotation>

<xs:simpleContent>

<xs:extension base="xs:anyURI">

<xs:attribute name="serviceLocation" type="xs:string"/>

<xs:attribute name="ttl" type="xs:double" use="optional"/>

<xs:anyAttribute namespace="##other" processContents="lax"/>

</xs:extension>

</xs:simpleContent>

</xs:complexType>

*Add subclause I.2.4.5 as follows:*

**I.2.4.5 Example 5: Content Steering**

The intent is to re-use the URL parameters of the MPD URL in the Content Steering URL as defined in clause K.3.6.

Assuming the DASH MPD is accessible through:

http://www.example.com/dash/urlparam1.mpd?token=1234&sessionID=h48djn

Then

1) Computation of an initial query string

initialQueryString="token=1234&sessionID=h48djn"

2) Computation of a final query

finalQueryString="token=1234&sessionID=h48djn"

and the corresponding MPD looks as follows:

|  |
| --- |
| <MPD  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns="urn:mpeg:dash:schema:mpd:2011"  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd urn:mpeg:dash:schema:urlparam:2014 DASH-MPD-UP.xsd"  type="static" mediaPresentationDuration="PT3256S" minBufferTime="PT1.2S" profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">  <EssentialProperty schemeIdUri="urn:mpeg:dash:urlparam:2014" xmlns:up="urn:mpeg:dash:schema:urlparam:2014">  <up:UrlQueryInfo includeInRequests="7" queryTemplate="$querypart$" useMPDUrlQuery="true"/>  </EssentialProperty>  <Period>  ...  </Period>  <ServiceDescription>  ...  <ContentSteering defaultServiceLocation="beta" queryBeforeStart="true"> <https://steeringservice.com/app/instance1234</ContentSteering>>  </ServiceDescription>  </Period>  </MPD> |

*In subclause I.4.2, replace Tables I.3 with the following:*

**Table I.3 — Semantics of ExtendedUrlInfoType element**

| **Element or Attribute Name** | | | **Use** | **Description** |
| --- | --- | --- | --- | --- |
|  | **ExtendedUrlInfoType** | |  | provides information for derivation of parameter string. This is an extension of **UrlQueryInfoType** element defined in Table I.1. |
|  |  | @includeInRequests | OD  (default: "segment") | specifies which HTTP GET requests shall carry parameters. Value is a white spaced concatenated list of the following keys: [Ed. (TS): We should add references to the specific elements that request. We also need to check if we have other requests such as patch.]  1) "segment" (all segment requests)  2) "xlink" (all XLink resolution requests),  3) "mpd" (all MPD requests),  4) "callback" (all requests triggered by DASH callback events),  5) "chaining" (requests for chained-to MPDs),  6) "fallback" (requests for the alternative MPDs),  7) "steering" (requests for Content Steering servers as defined in clause K.3.6).  8) a URN may be used where the value and the semantics assigned to the URN are defined by the owner of the URN.  9) "\*" any requested URL by the DASH client, including all of the above.  Default value is "segment", i.e. parameters will be only sent with segment requests  NOTE   Depending on the actual element used, parameter output goes either to query parameters (for **ExtUrlQueryInfo**) or HTTP headers (for **ExtHttpHeaderInfo**) |
|  |  | @headerParamSource | OD  (default: "segment") | specifies HTTP responses from which HTTP header values, identified by the template $*header:header-name*$, should be extracted from. Value is a white spaced concatenated list of the following keys :  1) "segment" (all segment requests)",  2) "xlink" (all XLink resolution requests),  3) "mpd" (all MPD requests),  4) "callback" (all requests triggered by DASH callback events).  Default value: empty string (no header parameters inspected)  If this attribute is present then: (a) @queryTemplate attribute shall be present and shall contain the $header:<header-name>$ identifier, and (b) neither @useMPDUrlQuery nor @queryString attribute shall be present. |
|  |  | @sameOriginOnly | OD | specifies that parameters must only be sent to the same origin they were instantiated from. In case of HTTP headers as source, the origin is defined as the origin of the HTTP request identified by the attribute @headerParamSource. In case the parameters are instantiated from the MPD or from the MPD URL, the origin is defined in both case by the MPD URL.  Two origins are the same as defined by IETF RFC 6454, i.e. same scheme/host/port triple (see 5. Comparing Origins)  Default value: false (no origin restrictions) |
| **Key**  For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory.  For elements: <minOccurs>...<maxOccurs> (N=unbounded)  The conditions only hold 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 @. | | | | |

*Add subclause K.3.6 as follows:*

**K.3.6 Content Steering**

In case the content is redundantly available at multiple service locations as defined in clause 5.6.5, content steering may support the static and dynamic selection of service locations.

**Table K.5 — Content Steering**

|  |  |  |
| --- | --- | --- |
| **Key** | **Type** | **Description** |
| ContentSteeringServer | URI | A URL that can be used to access the Content Steering server. The URL shall reference to a DASH Content Steering Manifest (DCSM) as defined in ETSI TS 1XX XXX.  If the resolved resource is not a Content Steering Manifest (DCSM), the resource is ignored. |
| defaultServiceLocation | string | This attribute specifies a space-separated list of Service Locations as defined in subclause 5.6.6 that the client should use to access the selected resources, in case multiple access exist. This for example applies, when no content steering server is available, or before a valid response from a content steering server is available. |
| queryBeforeStart | Boolean | If true, indicates that the player is expected to resolve the response from the Steering Server prior to starting playback.  Default value is false. |
| clientRequirement | Boolean | If true, indicates that the client, if it is in the context of the Service Description shall follow the content steering rules as defined in ETSI TS 1XX XXX.  If false, indicates to client that it is its own decision to make use of content steering operation or not.  Default value is true. |

*In subclause K.4.2.1, replace Tables K.5 with the following:*

**Table K.5 — Semantics of ServiceDescription element**

| **Element or Attribute Name** | | | | **Use** | **Description** |
| --- | --- | --- | --- | --- | --- |
|  |  | **ServiceDescription** | |  | Service Description |
|  |  |  | @id | M | specifies a unique identifier for this Service Description. The attribute shall be a unique unsigned integer value amongst **ServiceDescription** elements in the scope of the MPD. |
|  |  |  | **Scope** | 0 … N | specifies the scope of the Service Description. If present, this Service Description only targets DASH Clients identified by this Scope descriptor. DASH Clients not in scope, i.e. not recognizing any of the scope descriptor elements, are expected to ignore this service description. If no Scope element is present the Service Description applies to all clients. |
|  |  |  | **Latency** | 0 … N | specifies the latency targets for the service. The details are provided in subclause K.4.2.2, Table K.6. |
|  |  |  | **PlaybackRate** | 0 … N | specifies the playback rate targets for the service. The details are provided in subclause K.4.2.3, Table K.7. |
|  |  |  | **OperatingQuality** | 0 … N | specifies the operating quality targets for the service. The details are provided in subclause K.4.2.4, Table K.8. |
|  |  |  | **OperatingBandwidth** | 0 … N | specifies the operating quality targets for the service. The details are provided in subclause K.4.2.5, Table K.9. |
|  |  |  | **ContentSteering** | 0 … 1 | Specifies content steering operation for the service. The details are provided in subclause K.4.2.6, Table K.10. |
| **Legend:**  For attributes: M=mandatory, O=optional, OD=ptional with default value, CM= conditionally mandatory, F=fixed.  For elements: <minOccurs>...<maxOccurs> (N=unbounded)  The conditions only hold 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 @, List of elements and attributes is in ***italics bold*** referring to those taken from the Base type that has been extended by this type. | | | | | |

*Add subclause K.4.2.6 as follows:*

**K.4.2.6 Content Steering**

**Table K.10 — Semantics of ContentSteering element**

| **Element or Attribute Name** | | | | **Use** | **Description** |
| --- | --- | --- | --- | --- | --- |
|  |  | **ContentSteering** | |  | See **ContentSteeringServer** in clause K.5. For details, see subclause K.3.6. |
|  |  |  | @defaultServiceLocation |  | See **defaultServiceLocation** in Table K.5 |
|  |  |  | @queryBeforeStart | OD default: false | See **queryBeforeStart** in Table K.5 |
|  |  |  | @clientRequirement | OD default: true | See **clientRequirement** in Table K.5 |
| **Legend:**  For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory, F=fixed.  For elements: <minOccurs>...<maxOccurs> (N=unbounded)  The conditions only hold 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 @, List of elements and attributes is in ***italics bold*** referring to those taken from the Base type that has been extended by this type. | | | | | |

*Replacce subclause K.4.3.1with the following:*

**K.4.3.1 General**

<xs:complexType name="ServiceDescriptionType">

<xs:annotation>

<xs:documentation xml:lang="en">

**Service Description**

</xs:documentation>

</xs:annotation>

<xs:sequence>

<xs:element name="Scope" type="DescriptorType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="Latency" type="LatencyType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="PlaybackRate" type="PlaybackRateType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="OperatingQuality" type="OperatingQualityType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="OperatingBandwidth" type="OperatingBandwidthType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="ContentSteering" type="ContentSteeringType" minOccurs="0" maxOccurs="unbounded"/>

<xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

<xs:attribute name="id" type="xs:unsignedInt"/>

<xs:anyAttribute namespace="##other" processContents="lax"/>

</xs:complexType>

*Add subclause K.4.3.6 as follows:*

**K.4.3.6 Content Steering**

<!--ContentSteering -->

<xs:complexType name="ContentSteeringType">

<xs:simpleContent>

<xs:extension base="xs:anyURI">

<xs:attribute ref="defaultServiceLocation"/>

<xs:attribute name="queryBeforeStart" type="xs:boolean" default="false"/>

<xs:attribute name="clientRequirement" type="xs:boolean" default="true"/>

</xs:extension>

</xs:simpleContent>

</xs:complexType>

<xs:attribute name="defaultServiceLocation">

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:pattern value="[a-zA-Z0-9\_.-,]"/>

</xs:restriction>

</xs:simpleType>

</xs:attribute>

1. Under preparation. [↑](#footnote-ref-1)