|  |  |
| --- | --- |
|  |  |

 ISO/IEC JTC 1/SC 29/WG 03 N0833

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

**MPEG Systems   
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output Document

**Title: Defects under Investigation (DuI) on DASH**

**Status:** Approved

**Date of document:** 2023-04-28

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

**No. of pages:** 14 (with cover page)

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

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

**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

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

**ISO/IEC JTC 1/SC 29/WG 03 N833**

**April 2023, Antalya, TR**

|  |  |
| --- | --- |
| **Source** | **MPEG Systems working group** |
| **Status** | **Output** |
| **Title** | **Defects under Investigation (DuI) on DASH** |
| **Editor** | **Iraj Sodagar** |
| **Serial Number** | **22575** |

# Table of Contents

[Table of Contents 1](#_Toc102127658)

[1. Introduction 3](#_Toc102127659)

[2. Xlink usage, conformance to W3C, and remote entity element specification (m53957) 4](#_Toc102127660)

[2.1 Type, href, show and actuate are supported inconsistently on different elements 4](#_Toc102127661)

[2.2 As constrained, type and show do not appear to be useful 4](#_Toc102127662)

[2.3 Actuate asserts a default value of “onRequest” in Table 4 4](#_Toc102127663)

[2.4 Actuate only mentions “onLoad” and “onRequest” values 4](#_Toc102127664)

[2.5 Xlink support is only on a few elements 4](#_Toc102127665)

[2.6 All normal element attributes are optional when href is present 4](#_Toc102127666)

[2.7 Remote element entity contents vary 4](#_Toc102127667)

[2.8 The XML conformance of the remote element entities are not clear 5](#_Toc102127668)

[2.9 Nested href is permitted except when actuate=”onLoad” 5](#_Toc102127669)

[2.10 Xlink processing is profile-dependent and apparently optional 5](#_Toc102127670)

[3. On usage of edit list + audio codec (aac) + multi-period with continuity/connectivity signalling (m55420) 5](#_Toc102127671)

[4. Defects from DVB liaison (m55478) 6](#_Toc102127672)

[4.1 Period Continuity and Connectivity 6](#_Toc102127673)

[4.2 Track role signalling and its intended usage and usage in ETSI 103 285 7](#_Toc102127674)

[5. Event related defects 8](#_Toc102127675)

[5.1 Questions and potential defects on DASH-specific events (m57570) 8](#_Toc102127676)

[5.2 Explicit timing constraint attributes for (Inband) EventStream (m57406 and updated with m58114) 8](#_Toc102127677)

[5.2.1 Client processing model 9](#_Toc102127678)

[6. Some Observations on Update Events and Early Termination Events scheme (m59694) 9](#_Toc102127679)

[7. @codecs attribute and optional refinement codecs ( based on m58262) 10](#_Toc102127680)

[7.1 Introduction 10](#_Toc102127681)

[7.2 Proposal 11](#_Toc102127682)

[7.3 Discussion at MPEG #136 (http://mpegx.int-evry.fr/software/MPEG/Systems/DASH/spec/-/issues/223) 11](#_Toc102127683)

[7.3.1 Comments 1 11](#_Toc102127684)

[7.3.2 Comments 2 11](#_Toc102127685)

[8. [DuI] Clarification on use of SegmentBase.FailoverContent (m59704) 11](#_Toc102127686)

[8.1 Introduction 11](#_Toc102127687)

[8.2 Proposal 13](#_Toc102127688)

# Introduction

This documents the defects under investigation for 23009-1, 4th Ed (DASH). It contains the substance of the following documents.

# Xlink usage, conformance to W3C, and remote entity element specification ([m53957](http://wg11.sc29.org/doc_end_user/documents/130_Alpbach/wg11/m53957-v1-m53957-TidyingUpXlink.zip))

This section has not been reviewed by the DASH AHG.

This section makes reference to the DASH normative reference to the W3C XLINK standard:

[XML Linking Language (XLink) Version 1.1](https://www.w3.org/TR/xlink11/)

## Type, href, show and actuate are supported inconsistently on different elements

These xlink attributes are documented in Section 5.5. But not all DASH elements permit xlink:type and xlink:show. They should be used consistently (but see next issue below).

## As constrained, type and show do not appear to be useful

These attributes are “fixed” to the values “simple” and “embed” respectively – that is if used, they can \*only\* be set to this one value each. For xlink:type, “simple” is the W3C default value, so the addition of this seems redundant. There is no defined semantics in DASH for xlink:show, which was intended by W3C to control the visual presentation in e.g. HTML which frame a new resource is displayed. I’m not sure what DASH would do with this semantic use.

## Actuate asserts a default value of “onRequest” in Table 4

There is no W3C default value for xlink:actuate so this deviates from W3C which would otherwise require xlink:actuate to be present. Table 4 just asserts a statement of fact about this which is not technically true. Is this deviation necessary? Why not just require xlink:actuate if xlink:href is present?

## Actuate only mentions “onLoad” and “onRequest” values

No problem, but there is no conformance language in Table 4, just a statement of fact about this which is not technically true.

## Xlink support is only on a few elements

Xlink is explicitly supported by the spec syntax tables only for the following elements:

* Period
* AdaptationSet
* SegmentList
* InitializationSet
* EventStream

Although there is not explicit support for xlink on the MPD element, Section 5.5.3 discusses the handling of this case which is inconsistent.

## All normal element attributes are optional when href is present

All syntax tables have this note:

Note that the conditions [column] only holds without using xlink:href. If linking is used, then all attributes are "optional" and <minOccurs=0>

This violates the provisions of 5.5.3. Further 1) that would make the element invalid WRT to the schema and 2) the xlink resolution could fail in which case processing of the original element is expected as described in 5.5.3.

## Remote element entity contents vary

For elements that support xlink, the content referenced by xlink:href varies by element:

* MPD – one MPD element (maybe – not explicitly enabled in the syntax, but see 5.5.3)
* Period – one or more Period elements
* AdaptationSet – one AdaptationSet element
* SegmentList – one or more SegmentList elements
* InitializationSet – one InitializationSet element
* EventStream – one EventStream element (worded unclearly)

No problem, but Period and SegmentList are unique and there is no statement about why the other elements do not support xlink.

## The XML conformance of the remote element entities are not clear

The spec is silent about conformance of the remote element entities except for the type and number of elements. Should it conform to an “MPD”, i.e. it has an MPD as the main element and conforms to the MPD schema? Or is it just the element(s), e.g. just one or more Period elements?

The 4th Edition schema added Period as a top-level element, thus enabling Period alone to be conformant. The other elements are not currently supported.

Example G.11 is an example of a Period xlink, but the remote element entity is a valid MPD (i.e. with an MPD element).

We should consider a new namespace for a new schema that enforces the supported xlink remote element entities; and remove the new stand-alone Period from the MPD schema.

## Nested href is permitted except when actuate=”onLoad”

Section 5.5.3 declares this case to be “invalid circular references”. This is not explained especially when apparently the nested xlink construction is OK for xlink:actuate=”onRequest”.

## Xlink processing is profile-dependent and apparently optional

Section 8.3.2, 8.4.2 and 8.5.2 dealing with the On Demand, Live and Main profiles say:

Elements using the @xlink:href attribute may be ignored from the MPD. The Representations conforming to this profile are those not accessed through an Adaptation Set that uses an @xlink:href.

Section 8.6.2 for MP2TS profile says:

It shall be possible to present a presentation conforming to this profile without resolving @xlink:href in AdaptationSet or SegmentList elements. Any initial Period elements using @xlink:href may be ignored, and the first non-excluded Period shall have an explicit @start attribute. After the first non-excluded Period, there shall be no Period using @xlink:href.

# On usage of edit list + audio codec (aac) + multi-period with continuity/connectivity signalling ([m55420](http://wg11.sc29.org/doc_end_user/current_document.php?id=76477&id_meeting=184))

aac preroll

(1 or 2 aac packets)

First Period boundary

Second period boundary

Period continuity

Segment 1

K +

2 samples

Segment 2

K samples

Segment 3

K samples

Segment 4

K samples

Segment 2-1

K samples ??

Segment 2-2

K samples

Segment

K samples2-3

Third period boundary

No period contintinuity

Segment 3-1

K samples + 2

Segment 2-1

K + 2samples ??

Segment 3-1

K samples ??

Option 2 apply edit list again

Option 3 don’t apply edit list

aac preroll

(1 or 2 aac packets)

Segment duration in mpd = K \* d\_sample

Apply edit list again

Description of the issue:

1. First period may be different case from 2nd or 3rd period when handling content with editlist
2. For period continuity, preroll may not be needed as playback is continuous, but what if a player starts from the second period, what would be the right strategy to start playback ?
3. Either different segments need to be created (a segment with K samples vs K + 2 samples),
4. or duration of first segment timeline is shorter e.g. (K-2) \* d\_sample , but this is ambiguous based on wether editlist would need to be applied again after the period boundary or not
5. Still differences may exist between first period, a continuous period and a non-continuous period with regard to applying the edit again
6. Problem may be worse for codecs with more delay
7. Doing nothing is easiest but check audio quality aspect and consider codecs with more delay
8. Consider documenting issue and possibly the best practices around this

Issues:

1. Is having a different edit list allowed for period continuity/connectivity?
2. How severe is not using a preroll sample (a sample removed by an edit)?
3. What would be the processing model for continuous playback of representations with different edit?

Specific defect from DVB, more detail on "logical continuation", especially regarding the fields in the manifest.

# Defects from DVB liaison ([m55478](http://wg11.sc29.org/doc_end_user/current_document.php?id=76535&id_meeting=184))

# Period Continuity and Connectivity

DVB developed its own descriptors for period continuity, and would like to adopt the ones defined in MPEG 3rd and 4th editions while keeping the specific requirements from DVB. While DVB thinks the intention of both texts are similar, it would like to assert that indeed a DVB-DASH period continuous period aligns with what was intended in MPEG and consider additional requirements to be adopted (in MPEG). This is because specific text was found missing in MPEG-DASH specification regarding specific requirements used in DVB-DASH.

Here are the requirements from current MPEG DASH for period continuity

“ *Media Presentations should signal period-continuous Adaptation Sets by using a supplemental descriptor on Adaptation Set level with @schemeIdUri set to "urn:mpeg:dash:period-continuity:2015" with*

*— the @value of the descriptor matching the value of an @id of a Period that is contained in the MPD, and*

*— the value of the AdaptationSet@id being the same in both Periods. MPD should signal period-continuous Adaptation Sets if the MPD contains Periods with identical Asset Identifiers.*

*There exist special cases for which the media in one Adaptation Set is a continuation of the previous one, but the timestamps are not continuous. Examples are timestamp wrap around, encoder reset, splicing, or other aspects. Two Adaptation Sets in one MPD are period-connected if all conditions from period-continuity from above hold, except that the timestamps across Period boundaries may be non-continuous but adjusted by the value of the @presentationTimeOffset at the Period boundary. However, for example the Initialization Segment is equivalent within the two Adaptation Sets. Media Presentations should signal period-connected Adaptation Sets by using a supplemental descriptor on Adaptation Set level with @schemeIdUri set to "urn:mpeg:dash:period-connectivity:2015". Period continuity implies period connectivity.* ”

DVB-DASH adds to the requirement of being Associated as below.

“ If Adaptation Sets in two different Periods are associated, then the following parameters shall be identical for the two Adaptation Sets:

• the language as described by the @lang attribute;

• the media component type described by the @contentType attribute;

• the picture aspect ratio as described by the @par attribute;

• any role properties as described by the Role elements;

• any accessibility property as described by the Accessibility elements;

• any viewpoint property as described by the Viewpoint elements;

• for audio Adaptation Sets, all values and presence of all attributes and elements: @mimeType, @codecs, @audioSamplinigRate, @AudioChannelConfiguration

NOTE: Additional requirements may apply to specific content protection schemes, the details of which are outside the scope of the present document. “

The key question is if this matches the “logical continuation” of the media described in MPEG-DASH, and if so if some of these aspects more explicitly be mentioned in the MPEG-DASH specification?

If not, is what DVB specified in line with what was intended as appropriate for MPEG period continuous periods. Last, would MPEG endorse DVB to apply to these additional constraints, considering that DVB content should playback on typically (all) DASH conforming players as it is intended by DVB-DASH, or are some requirements missing.

Another question/concern raised by a DVB member was regarding the usage of edit lists, by having equivalent initialisation segments, this would imply having the same editlist applied (i.e. not a new edit). This would mean that for example for aac segments in a continuous/connected period no different edit can be applied, and e.g. audio priming samples would be played by the client. DVB would like to understand if 1. This is the intended behaviour and 2. If MPEG thinks there are any perceptual effects caused by this.

Please let us know if

1. The DVB requirements map with what was intended in MPEG-DASH
2. If any updates are planned to the MPEG-DASH specification to include some of the specific requirement for DVB, this allows DVB to update its own specification to introduce less specific requirements for period continuity
3. If the behaviour with edit lists was intended and if MPEG things there are any perceptual effects to not having for example no priming samples for audio in a continuous period.

# Track role signalling and its intended usage and usage in ETSI 103 285

MPEG-DASH 3rd and 4th define new rack roles such as forced subtitles and several others. DVB used EBU TT D which is based on ttml, and in ttml such signatures can be signalled aswell. In addition, many DVB-DASH clients cannot render multiple timed text adaptationsets simultaneously. DVB would like to understand better the intended usage of the track roles by MPEG. It seems that for DVB-DASH the best option is:

1. Signalling multiple track roles per timed text track/Adaptationset while keeping the number of AdaptationSets limited, serving subtitles with different track roles in a single track/AdaptationSet
2. For some special cases, if deemed necessary a separate AdaptationSet with only specific track roles may be created, i.e. hard hearing subtitles, forced subtitles

Could MPEG elaborate if this usage aligns with what was intended by the MPEG-DASH specification?

# Event related defects

# Questions and potential defects on DASH-specific events ([m57570](https://dms.mpeg.expert/doc_end_user/current_document.php?id=79762&id_meeting=187))

* In **clause 5.10.4.2** 0xFFFF should be 0xFFFFFFFF

# Explicit timing constraint attributes for (Inband) EventStream ([m57406](https://dms.mpeg.expert/doc_end_user/current_document.php?id=79598) and updated with m58114)

For content authors, authoring Events in DASH is challenging, especially, native DASH clients may not take into consideration intended processing be a specific schemeIdUri (e.g. on-receive, on-start) or how long in advance an Event should be announced in a dynamic MPD. To enable robust server and client-side support for event processing we recommend explicit signalling of intended timing properties. This way an encoder/packager or client can exhibit the correct behaviour, which is important, especially in live (@type=”dynamic”) cases.

Table 38 — Event Stream Semantics

|  |  |  |
| --- | --- | --- |
| @preAnnounceTicks | O D=0 | Events targetting media presentation time T corresponding to wallclock time W will be available in an MPD update with @publishTime at least @preAnnounceTicks/@timescale seconds before W. In case of InbandEventStream, the Event will be available in a segment that starts at least @preAnnounceTicks/@timescale seconds before the intended event presentation time. This attribute enables clients to take pre-announcement into account when events require a lot of (pre-) processing. |
| @removeTicks | O D=0 | The Event may be removed from the manifest @removeTicks/@timescale seconds after no segments with a presentation time overlapping the Event are available. |
| @dispatchMode | O D=”on-recieve” | the client processing is “on-receive” or “on-start” DASH client is expected to dispatch events in on-receive or on-start mode |

## Client processing model

There is no need to change the text in Annex A, but it will become implementable for the case the @dispatchMode is signalled in the MPD and set to on-start. The default remains on receive, but on-start has some important applications as well that are enable by this modification.

# Some Observations on Update Events and Early Termination Events scheme (m59694)

This document presents some observations and problems that we see arising for “update” and/or early termination or extension of events.

In the last meetings there was some notion introduced allowing for an update of an event. Also, we see in some application schemes of events, where the termination of stopping of an event is done in the syntax of the scheme.

Something to be careful with, that we have seen, is say we have an event with a duration of 30 minutes, and we stop it after 10 minutes with an update, with a 10 minute update event (in case DASH introduces a syntax to update an event), or via application level signal (an event with a duration zero that updates the event in case no native event update is included in DASH), an event can be stopped.

An example in common practice of such as case is a cue-out event, e.g. a SCTE-35 splice\_insert with out\_of\_network\_inidicator set to 1 and a duration of 30 minutes, followed in this case by a cue-in a splice\_insert with out\_of\_network\_indicator set to 0 is received and set much earlier (after 10 minutes).

The problem we like to emphasize is the following. In case an such an update event is used (either native or specified by the scheme such as in the example above), there is a possibility that the update/cue-in event will move out of the active segment window/manifest before the original cue-out or the original event.

In this case a newly joining client interpreting the MPEG-DASH presentation may interpret the event as still active. We recommend any solution to this problem to take this into account.

|  |  |
| --- | --- |
|  |  |
| Inband event is stopped earlier (native) | What if a client only received the original but missed the update event ? Segments in the past cannot be changed. It seems like the update event needs to be repeated. |
| Inband event is stopped earlier (non-native) | What if a client only received the original but missed the update event ? It seems like the update event needs to be repeated. |
| MPD Event is stopped earlier (native) | What if the original Event is still in the active window but the update event is not ? DASH can natively support removing such events from the MPD/Manifest if such native update event would exist, but currently no guideline or recommendation for this is defined. |
| MPD Event is stopped earlier (non-native) | What if the original Event is still in the active window but the update event is not ? for non-native events and the logic decided at the application level, the update event must exist in the MPD as long as the original event exist in the MPD. This would limit/prohibit updates to the duration of an event. This makes update events tricky to implement in a non-native manner. |
| Event is stopped early (metadata track) | New samples no longer carry the event/metadata, but in this case the correlation between the update/cancel/stop is not explicit. In case explicit signalling is applied similar requirement of repetition may arise. Updates can be implemented by using same id/scheme but different payload in an event message track. |

Note: If no further evidence is shown, then it will be considered an application domain issue and will be removed from DuI.

# @codecs attribute and optional refinement codecs ( based on m58262)

# Introduction

In many cases a regular AVC or HEVC stream is accompanied with optional metadata. If the metadata is not understood, the underlying stream can still be decoded. Whenever it is understood, the consumer’s experience will be enhanced.

There are multiple examples of such codecs. Dynamic metadata technologies such as DolbyVision Profile 8 and HDR10+, film grain modeling, and few others are using the approach. In most cases SEI messages are used for the purpose, but e.g. many DolbyVision implementations uses RPU NAL units instead of SEI messages. It is quite possible to have streams combining multiple technologies – e.g a stream decodable as HDR10 (“base layer”), can also be decodable as DolbyVision 8.1 and (simultaneously) as HDR10+, and also having film grain SEI message.

The current way of expressing this is using a codec string and CMAF media profile. Unfortunately, the result is sometimes confusing. For example, the current CMAF cdm1 profile requires use of dvhe.08.09 or dvh1.08.09 as **AdaptationSet**@codec string. As a result, the media capability tested would be decoding Dolby Vision profile 8 (per SCTE 215-1-1) which will render the representation unplayable for devices supporting only HDR10 or HDR10+. This restricts the reach of a representation or requires multiple adaptation sets pointing to same set of segments. Moreover, hard-coded operating point in the cdm1 media profile means that cdm1 can support only up to 4Kp60 content. In our opinion neither of these options as a good and scalable solution, especially once there is a combination of different enhancement mechanisms.

The example above also demonstrates that CMAF brands don’t scale in this case. Brands need to be defined for different profiles/levels of each codec and its peculiarities, leading to an explosion of brands.

The newly introduced @segmentProfiles attribute seemingly allows both codec strings and brands, however its definition starts with essentiality of the profiles for a representation. Ability to process PQ10 is the only hard requirement for the example above – HDR10, HDR10+, and DolbyVision are all optional and requiring them would again prevent equipment capable of only PQ10 or HDR10 from being presented.

In our opinion, there is a need for a simple solution to tame this complexity and avoid workarounds.

# Proposal

Firstly, we propose to add an extra attribute, **RepresentationTypeBase**@optionalProfiles, listing a space-separated list of ISO-BMFF brands or RFC 6381 codec strings. These profiles will be used to list optional codecs or metadata present in the stream. Note that this is a simple list – i.e., it does not describe whether two enhancements are expected to be combined or are alternatives or one of the two is preferred. These details are implementation-dependent, while listing options is to allow capability negotiation prior to downloading segments.

Secondly, in order to allow expressing presence of important SEIs we propose to define a codec string for the H.274 VSEI specification. The goal would then to be able to e.g. express the existence of a frame grain characteristics SEI message (defined in sec. 8.5 of H.274) in a frequency filtering mode e.g. as vsei.05.00, where 05 stands for film grain characteristics SEI and 00 for the frequency filtering model (as opposed to the alternative autoregressive model).

# Discussion at MPEG #136 (<http://mpegx.int-evry.fr/software/MPEG/Systems/DASH/spec/-/issues/223>)

# Comments 1

I understand the problem, I was pointing to this many times. The idea that a new codecs parameter is defined for a codec with some SEI messages was plain wrong. but I am not sure why we need to solve this in DASH.

I want to point what the DASH profile for CMAF content says

* The @mimeType shall be set to "<contentType>/mp4". NOTE 1: Setting @mimeType to "<contentType>/mp4, profiles='cmfc'" is correct, but may result that the media stream is not recognized. Hence, conformance to the CMAF profile is preferably documented by the @containerProfiles signalling.
* The @containerProfiles parameter should be present. If present, it shall include at least one profile string, namely a structural brand being either 'cmfc' or 'cmf2'. In addition, it should include a CMAF media profile brand. NOTE 2: A CMAF Media profile brand is not required as conformance to a structural CMAF brand may be sufficient.
* The @codecs parameter shall be set to according to the sample entry codingname field of the CMAF Principal Header. NOTE 3: By doing so, it is expected that this parameter is sufficient for capability exchange and media pipeline initialization. Representations may potentially signal different values for the @codecs parameter.

So we have lots of flexibility already. However, on the sample entry you need to be using something from the file format. Only if the file format defines this, then we can check how we can move this forward on the codecs parameter. The list is not good at all, and saying that this is implementation dependent does not work.

on the media profile, the containerProfiles allows this signaling.

On the SEI messages, we have defined the SEI manifest for this purpose to declare what you find the bitstream. DASH cannot define all of this, it needs to be done on lower levels first.

# Comments 2

We need to discuss how to solve this at the file format level, i.e. how it is described in CMAF tracks.

# [DuI] Clarification on use of SegmentBase.FailoverContent (m59704)

# Introduction

Example G.22 in ISO/IEC 2300901 5th edition shows a SegmentBase element at the Representation level to include the FailoverContent element. By itself, this is expected as FailoverContent is defined as part of the segment base information. However, in the example, a SegmentTemplate is present higher up the hierarchy at the AdaptationSet level. This seems to conflict with other parts of the spec regarding how a representation builds its segment information, the restrictions around it, and when a SegmentBase element should be used.

This reduced snippet illustrates the hierarchy in example G.22:

```

<AdaptationSet *id*="1">

    <SegmentTemplate>

        <SegmentTimeline>

        </SegmentTimeline>

    </SegmentTemplate>

    <Representation *id*="video1">

        <SegmentBase>

        </SegmentBase>

```

A representation builds its segment information by inheriting the information found at higher levels (e.g., Period, AdaptationSet) and combining that with the information at its level. The Representation inherits the SegmentTemplate from the AdaptationSet level in the example above. This information then needs to be combined with the SegmentBase found at its level. While these elements share, or inherit, from the same segment base information, they are different elements. The following section of the spec could be interpreted as the information in these elements would not be merged.

5.3.9.1 General

```

SegmentBase, SegmentTemplate and SegmentList shall inherit attributes and elements from the same element on a higher level. If the same attribute or element is present on both levels, the one on the lower level shall take precedence over the one on the higher level.

Several mechanisms are available to specify the Segment Information. Specifically, each Representation shall have assigned exactly one of the following choices to determine the Segment Information, either by direct presence in the Representation element or by inheritance from the higher levels:

— one SegmentList element — for syntax and semantics, refer to subclause 5.3.9.3,

— one SegmentTemplate element — for syntax and semantics, refer to subclause 5.3.9.4,

— one or more BaseURL elements, at most one SegmentBase element, and no SegmentTemplate or SegmentList element. The SegmentBase element is defined in subclause 5.3.9.2.

```

Additionally, only one type may be used to determine the segment information at the Representation level. This appears to restrict having both elements being a source of segment information.

Due to the inherited segment information, the representation is a multiple segment representation. Whereas SegmentBase is mentioned as only being used when the other types do not apply and there is only one media segment in a Representation:

5.3.9.2.1

```

The SegmentBase element is sufficient to describe the Segment Information if and only if a single Media Segment is provided per Representation and the Media Segment URL is included in the BaseURL element.

In case multiple Media Segments are present, either a SegmentList or a SegmentTemplate shall be used to describe the Segment Information. SegmentList or a SegmentTemplate share the multiple Segment base information as provided in subclause 5.3.9.2.2, Table 16.

```

5.3.9.5.3 Media Segment information

```

None of the above: in this case, only a single Media Segment shall be present with the URL provided by a BaseURL element and the SegmentBase element may be present.

```

FailoverContent is defined as part of the SegmentBaseType. SegmentTemplate and SegmentList are both MultipleSegmentBaseInformation types which in turn inherit from the SegmentBaseType. From this inheritance, another way to include the FailoverContent in the example would be to use the same multiple segment type found at the AdaptationSet level that only contains the failover information:

```

<AdaptationSet *id*="1">

    <SegmentTemplate>

        <SegmentTimeline>

        </SegmentTimeline>

    </SegmentTemplate>

    <Representation *id*="video1">

        <SegmentTemplate>

            <FailoverContent>

            </FailoverContent>

        </SegmentTemplate>

```

The above example also aligns with G.13 from the spec where a Representation is providing its own initialization attribute in addition to the inherited SegmentTemplate from the AdaptationSet

## 8.2 Proposal

The intent of the FailoverContent element was to provide a Representation-level “negative” override, and its combination with Representation.SegmentTemplate.SegmentTimeline defeats the purpose.

We propose to modify the text to make it clear that Representation-level “override” of parts of AdaptationSet.SegmentTemplate is feasible.

This can be achieved by adding text to sec. 5.3.9.1, clarifying that Representation.SegmentBase can be present together with AdaptationSet.SegmentTemplates and will be used to describe representation-specific properties such as segment availability when different from at least one other representation in the current adaptation set.

# 9 Defect from DASH-IF liaison (m63532)

In the implementation of ISO/IEC 23009-1, and the template mechanism for $time$, we identified some confusing aspects. The specification says as follows

* $time$: This identifier is substituted with the value of the MPD start time of the Segment being accessed. For the Segment Timeline, this means that this identifier is substituted with the value of the **SegmentTimeline**@t attribute for the Segment being accessed.
* **S**@t: This value of this attribute minus the value of the @presentationTimeOffset specifies the MPD start time, in @timescale units, of the first Segment in the series. The MPD start time is relative to the beginning of the Period.
* The Segment URL for a Media Segment is obtained by replacing the $Time$ identifier by the earliest presentation time obtained from the **SegmentTimeline**.

From the dash.js implementers understanding it should be described more clearly that for **SegmentTimeline** we are replacing $time$ with a media time and not with an MPD start time. Otherwise, we would need to apply the @presentationTimeOffset before replacing $time$. Moreover, we might want to add to the $time$ description that the value is specified in timescale units unless this is already implicitly defined with “MPD start time”.

As an example:

* dash.akamaized.net/dash264/TestCasesHD/1c/qualcomm/1/MultiRate.mpd
* This MPD contains a @presentationTimeOffset and uses $time$. To fetch the first segment, we replace $time$ with 0 and not with 0 – 1024. This is stated in the definition of $time$ but as mentioned above it could be confusing as it is not the MPD start time of the segment.

We also identified another issue as follows:

* From DASH spec (section 5.3.9.6): "If a Segment Index ('sidx') box is present, then the values of the SegmentTimeline shall describe accurate timing of each Media Segment, Specifically, these values shall reflect the information provided in the Segment index ('sidx') box,..."
* However, if $Number$ is used, then the accuracy may not be needed. Hence the above condition should be further restricted to $Time$. Additional information on maximum deviation may be provided.

We also identified the inefficiency of accurate signaling of SegmentTimeline if audio requires occasionally shorter segments. This is applied with a pattern, but such pattern has not been signaled. DASH-IF has discussed this in details and provided a draft solution in the issue tracker [here](https://github.com/Dash-Industry-Forum/Live/issues/75). DASH-IF would be interested if MPEG can address this issue in an amendment.