

DASH-IF IOP-10 V5.0.0 (2023-01)



**DASH-IF Interoperability Points;  
Part 10: Events and Timed Metadata**



***DASH Industry Forum***

3855 SW 153rd Dr.  
Beaverton, OR 97003 - USA

---

Email : admin@dashif.org

***Important notice***

---

The present document can be downloaded from:  
<http://www.dashif.org/guidelines>

# Contents

Intellectual Property Rights .....	5
Foreword.....	5
Modal verbs terminology .....	5
Executive summary .....	5
Introduction .....	5
1 Scope.....	7
2 References .....	7
2.1 Normative references .....	7
2.2 Informative references .....	7
3 Definition of terms, symbols and abbreviations.....	7
3.1 Terms .....	7
3.2 Symbols .....	8
3.3 Abbreviations.....	8
4 Introduction .....	8
5 Client processing for DASH Event streams and timed metadata tracks .....	8
6 Inband event authoring.....	8
6.1 General characteristics and constraints .....	8
6.2 Authoring guidelines for Inband Events .....	10
6.2.1 Dispatch mode selection.....	10
6.2.2 Duplication of Inband Events in Periods.....	10
6.2.2.1 Duplicating the event instances in representations in one Adaptation Set.....	10
6.2.2.2 Duplicating the event instances in different Adaptation Sets .....	11
6.2.2.3 Repeating the event instances in one Period.....	11
6.2.2.4 Repeating the event in different Periods.....	11
6.2.3 Event timing constraints.....	11
6.2.3.1 Early insertion of Inband Events .....	11
6.2.3.2 Early insertion of InbandEventStream Element .....	11
6.2.3.3 The ‘event_duration’ field .....	12
6.2.4 Inband Event message_data and payload size constraints.....	12
6.2.5 MPD update constraints .....	12
6.2.6 Efficiency and Deployment Constraints.....	12
6.3 Common use-cases .....	12
7 MPD event authoring .....	12
7.1 General Properties and constraints.....	12
7.2 Authoring guidelines for MPD events .....	13
7.2.1 Dispatch mode selection.....	13
7.2.2 Event instances order .....	14
7.2.3 Event instances without @id.....	14
7.2.4 Repeating events .....	14
7.2.4.1 Repeating event instances in one Period.....	14
7.2.4.2 Repeating the event cross Period.....	14
7.2.5 Event timing considerations .....	14
7.2.6 Announcement and removal of Events in the MPD .....	14
7.2.7 Event payload size constraints .....	14
7.2.8 Event payload character coding constraints .....	15
7.2.9 Event and EventStream MPD update constraints.....	15
7.2.10 MPD Event Efficiency and Deployment recommendations.....	15
7.3 Common use-cases .....	15
8 Combination of MPD events and Inband events.....	15
9 Timed Metadata Representations .....	16

9.1	Characteristics of timed metadata Representations.....	16
9.2	Authoring guidelines.....	16
9.2.1	Dispatch mode selection.....	16
9.2.2	Metadata sample timing considerations .....	16
9.2.3	Payload considerations.....	16
9.2.4	Scheme signaling and Sample Entry Requirements .....	16
9.2.5	MPD signaling .....	16
9.2.6	Metadata samples carrying the same event instances.....	17
9.2.7	Application processing considerations .....	17
9.3	Common use-cases .....	17
10	Support of DASH events.....	18
11	Externally defined event schemes .....	18
	Annex A (Informative): Change History .....	19

---

## Tables

Table 6-1	Recommended usage of optional InbandEventStream attributes .....	8
Table 6-2	Recommended usage of fields in the DASHEventMessageBoxes .....	9

---

## Examples

Exempl 9- 1	Example for describing of a metadata track in MPD .....	17
Exempl 9- 2	Example for describing of Event Message Track in MPD .....	17

---

# Intellectual Property Rights

## Disclaimer

This is a document made available by DASH-IF. The technology embodied in this document may involve the use of intellectual property rights, including patents and patent applications owned or controlled by any of the authors or developers of this document. No patent license, either implied or express, is granted to you by this document. DASH-IF has made no search or investigation for such rights and DASH-IF disclaims any duty to do so. The rights and obligations which apply to DASH-IF documents, as such rights and obligations are set forth and defined in the DASH-IF Bylaws and IPR Policy including, but not limited to, patent and other intellectual property license rights and obligations. A copy of the DASH-IF Bylaws and IPR Policy can be obtained at <http://dashif.org/>.

The material contained herein is provided on an "AS IS" basis and to the maximum extent permitted by applicable law, this material is provided AS IS, and the authors and developers of this material and DASH-IF hereby disclaim all other warranties and conditions, either express, implied or statutory, including, but not limited to, any (if any) implied warranties, duties or conditions of merchantability, of fitness for a particular purpose, of accuracy or completeness of responses, of workmanlike effort, and of lack of negligence.

In addition, this document may include references to documents and/or technologies controlled by third parties. Those third-party documents and technologies may be subject to third party rules and licensing terms. No intellectual property license, either implied or express, to any third-party material is granted to you by this document or DASH-IF. DASH-IF makes no warranty whatsoever for such third-party material.

Note that technologies included in this document and for which no test and conformance material is provided, are only published as candidate technologies, and may be removed if no test material is provided before releasing a new version of this guidelines document. For the availability of test material, please check <https://www.dashif.org>.

---

## Foreword

This Technical Specification (TS) has been produced by the DASH-IF Technical Working Group.

---

## Modal verbs terminology

In the present document "shall", "shall not", "should", "should not", "may", "need not", "will", "will not", "can" and "cannot" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in deliverables except when used in direct citation.

---

## Executive summary

Part 10 of DASH-IF IOP v5 provides a common timing and client processing model for application MPD events, inband events, and timed metadata tracks. This processing model describes a timing model for the DASH client to process and use a common processing pipeline to parse the event and metadata instances, and dispatch them to the Application. This part also includes a sample API for applications to subscribe to one or more event and/or timed metadata schemes. Additionally, this part provides authoring guidelines and best practices for using the DASH events and metadata tracks to carry application-specific sparse events through the DASH client. Finally, a few event schemes and their timing characteristics that are currently defined in some consortia specifications are described as examples.

---

## Introduction

This document is Part 10 of a multipart set of documents, collectively called "IOP V5.0.0". All the parts are:

1. Overview, architecture and interfaces
2. Core principles and CMAF mapping
3. DASH on-demand services

4. DASH live and low-latency live services
5. Ad insertion and content replacement
6. Content protection
7. Video
8. Audio (this document)
9. Text
10. Events and timed metadata (this document)
11. Additional functionalities
12. Conformance and reference tools

---

# 1 Scope

The scope of this part is the timing and processing of the DASH event and timed metadata track timing as well as the corresponding DASH client-Application API for these events and metadata.

The DASH standard includes MPD events to include in DASH manifest. It also enables including inband events to be carried with the media segments. Finally, it allows the streaming of metadata tracks. All three functionalities may be used to deliver application-related events to the Application. Since these data are timed sensitive, the DASH client must receive this data and pass it to Application such that the application can use them on time.

This part draws the timing model for DASH events and timed metadata tracks based on the MPEG DASH specification. Based on this timing model, this document also includes a client processing model for these data. This part also includes an API for dispatching events and timed metadata information to the Application. Finally, authoring guidelines and best practices are provided for content/service providers on how to use DASH events and timed metadata to deliver application-specific sparse data through the DASH client.

---

## 2 References

### 2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

The following referenced documents are necessary for the application of the present document.

- [1] ISO/IEC 23009-1 "Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats, 5th Edition"
- [2] ISO/IEC 14496-12 "Information technology -- Coding of audio-visual objects -- Part 12: ISO base media file format"
- [3] ISO/IEC 23000-19 "Information technology — Multimedia application format (MPEG-A) — Part 19 Common media application format (CMAF) for segmented media"
- [4] ISO/IEC 23001-18 "Information technology — Multimedia application format (MPEG-B) — Part 18 Event message track format for the ISO base media file format"

### 2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are not necessary for the application of the present document, but they assist the user with regard to a particular subject area.

- [i.1] ANSI/SCTE 35 2020, Digital Program Insertion Cueing Message for Cable
  - [i.2] SCTE 214-1 2022, MPEG DASH for IP-Based Cable Services Part 1: MPD Constraints and Extensions, [https://www.scte.org/documents/5085/SCTE\\_214-1\\_2022.pdf](https://www.scte.org/documents/5085/SCTE_214-1_2022.pdf)
- 

## 3 Definition of terms, symbols and abbreviations

### 3.1 Terms

For the purposes of the present document, the following terms apply:

**ABR encoder** – live encoder that converts a broadcast stream or mezzanine into a ladder of different bit-rate tracks

## 3.2 Symbols

For the purposes of the present document, the following symbols apply:

## 3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

CMAF	Common Media Application Format
DASH	Dynamic Adaptive Streaming over HTTP
DASH-IF	DASH Industry Forum
HTTP	HyperText Transport Protocol
ISO	International Standards Organization
LAT	Latest Arrival Time
MPD	Media Presentation Description
MPEG	Moving Pictures Experts Group
SCTE	Society of Cable Telecommunications Engineers

---

## 4 Introduction

This part describes the DASH events and timed metadata track timing and processing model. It describes the timing models of MPD and inband events as well as the timing model of the timed metadata tracks. This part also outlines the DASH player's reference architecture for processing the DASH events as well as timed metadata tracks, the possible dispatch modes and the information carried to the Application. Finally, it defines the reference API for the Application to subscribe to the events and/or metadata tracks as well as the API for dispatching event instances and metadata samples.

A server/application provider should consider the information provided in this section for building interactive application as the timing and processing model of the events and metadata impact the usability and capabilities of application build using these features.

---

## 5 Client processing for DASH Event streams and timed metadata tracks

The DASH-IF client shall conform to MPEG-DASH [1] clause A.13 except the following clauses which are optional:

- A.13.11.

---

## 6 Inband event authoring

### 6.1 General characteristics and constraints

Inband events are signalled in media segments using the `DASHEventMessageBox` as defined in MPEG-DASH [1] clause 5.10.3.3.

Presence of Inband events should be signalled using an `InbandEventStream` element in the Media Presentation Description (MPD).

Inband Events signalled by `DASHEventMessagesBoxes` with a `scheme_id_uri` and value announced in the MPD by a corresponding `InbandEventStream` element may be ignored by a DASH client.

Table 6-1 details constraints of attributes in the `InbandEventStream` Element.

**Table 6-1 Recommended usage of optional `InbandEventStream` attributes**

<code>@xlink:href</code>	May be present in case the <code>InbandEventStream</code> is a remote element
--------------------------	---



@xlink:actuate	Shall be set to 'onLoad' or absent
@schemeIdUri	Shall be set to the <code>scheme_id_uri</code> value used in <code>DASHEventMessageBoxes</code> of the event stream that are expected to be processed by the DASH client.
@value	Should be set to the <code>value</code> field used in the <code>DASHEventMessageBoxes</code> of the event stream (if present) that are expected to be processed by the DASH client. Otherwise, it shall be absent.
@timescale	This attribute should be absent, if present it shall match the <code>timescale</code> field used in the corresponding <code>DASHEventMessageBoxes</code> .

NOTE: Since MPEG-DASH 5<sup>th</sup> edition, **InbandEventStream** shall not carry the **InbandEventStream@presentationTimeOffset** attribute, instead the **SegmentTemplate/SegmentBase@presentationTimeOffset** associated to the media representation carrying the event message box is applied for calculating the correct event start time. In other words, it is assumed the media segments and the `DASHEventMessageBox` carried in them use the same presentation time offset (after taking into account a possibly different in timescale.)

Each Inband event is defined by the following parameters in the MPEG-DASH [1] `DASHEventMessageBox` 5.10.3.3, shown in Table 6-2.

**Table 6-2 Recommended usage of fields in the `DASHEventMessageBoxes`**

Field	Field type	Comment
<code>emsg.version</code>	flag	Defined by the version flag, version 0 is used for segment relative timing, version 1 for representation relative timing
<code>emsg.scheme_id_uri</code>	string	Signals a uri to define the scheme as defined by the developer of the scheme (which may be another external organization)
<code>emsg.value</code>	string	Field may be set according to the guidelines provided by the used scheme, otherwise it shall be set to the empty string
<code>emsg.timescale</code>	unsigned int 32	This field should be the same as in <code>mdhd</code> box of the initialization segment of the Representation.  NOTE: In a CMAF track, the <code>mdhd</code> timescale and this field are required to be equal according to CMAF [3].
<code>emsg.presentation_time</code>	unsigned int 64	Shall be set to the relative time offset of the event start time. <code>presentation_time</code> is relative to <b>Period@start</b> adjusted with the <code>@presentationTimeOffset</code> associated to the media representation carrying the <code>DASHEventMessageBox</code> .  NOTE: the <b>InbandEventStream</b> element, since the 5 <sup>th</sup> edition of MPEG-DASH no longer carries the <code>@presentationTimeOffset</code>
<code>emsg.presentation_time_delta</code>	unsigned int 32	Shall be set to the relative time offset of event start time in the scale of timescale. <code>presentation_time_delta</code> is relative to the segment's earliest presentation time.
<code>emsg.event_duration</code>	unsigned int 32	Shall be set to the event duration in units of the timescale.
<code>emsg.id</code>	unsigned int 32	Unique identifier to distinguish events with the same <code>scheme_id_uri</code> and <code>value</code> and to detect repetitions.

		Two Inband event instances are identified to be equivalent if they have identical values in the following fields: <ul style="list-style-type: none"> <li>• <code>scheme_id_uri</code></li> <li>• <code>value</code></li> <li>• <code>id</code></li> </ul>
<code>emsg.message_data</code>	<code>unsigned int8[]</code>	Shall be set to the message data payload, i.e. the data to be passed to the application containing the scheme specific message. It may be empty if no message data is needed to be passed to the application.

## 6.2 Authoring guidelines for Inband Events

### 6.2.1 Dispatch mode selection

The scheme URI/value owner establishes the dispatch mode for the use of information carried by Inband Events:

- In on-receive dispatch, the `event_duration` is the active duration of the event from the start of the event instance.
- In on-start dispatch, the `event_duration` indicates the valid duration in which the DASH client would dispatch the event to the application.

Therefore, the recommendations for on-receive dispatch and on-start dispatch are as follows:

1. The on-receive dispatch mode is recommended in the following cases:
  - a. On-receive dispatch mode is used for events that typically have a “state” or convey a “state.” In this case, an event instance is usually expected to carry information about its active interval. The active interval of an event instance is the duration in which the properties of the event instance are valid for that duration. For example, it is an interval in which a specific text is overlaid over the video. The active interval is signalled by the `event_duration` field.
  - b. The on-receive dispatch mode is used whenever instantly exposing the event to the Application has benefits for the application logic.
2. The on-start dispatch mode is recommended in the following cases:
  - a. The on-start dispatch mode is used for events that typically do not have any “state” and convey a “change of state” instead. In this case, an event instance doesn’t have an active interval. It is a toggle switch that may cause a change of state. In this case, the event instance carries the duration in which the event can be conveyed to the application.
  - b. The on-start dispatch mode is used whenever exposing the event instance at the start time of the event instance is beneficial to the application logic. An example of this could be a message or popup to be present during the specific start time, or the common case of monitoring viewer events when viewing an ad break. In the last example, when reaching the media time corresponding to the event, the application may trigger a callback to a server or tracker and the triggering of the event. Therefore, the on-start mode is required to match event time the to actual presentation time and dispatch time accurately.

### 6.2.2 Duplication of Inband Events in Periods

#### 6.2.2.1 Duplicating the event instances in representations in one Adaptation Set

Due to dynamic switching between two or more representations in one Adaptation Set by the DASH client, the event instances should be duplicated across all representations of one Adaptation Set such that in the case of switching, the DASH client does not miss an event instance. The switching between representations may occur at the segment boundaries, or the chunk boundaries in the case of low latency streaming, and in both cases, adequate repeating of one event instance should be considered to assure that the DASH client does not miss an event.

### 6.2.2.2 Duplicating the event instances in different Adaptation Sets

Inband events may also be present in more than one Adaptation Set in case that only one of those Adaptation Sets is selected for download and playback by a client. For example, in case an MPD includes multiple AdaptationSet elements of the same @contentType, each should carry the InbandEventStream. Since each Adaptation Set or Representation may have a different @presentationTimeOffset as well as a different time scale, the presentation\_time/presentation\_time\_delta, event\_duration and timescale values of each DASHEventMessageBox must be adjusted to align its timing with the same event in other Representations.

### 6.2.2.3 Repeating the event instances in one Period

The recommendations are as follows:

1. Since DASHEventMessageBox(es) with version 1 carry the presentation time relative to the media Period@start time adjusted with SegmentTemplate@presentationTimeOffset or SegmentBase@presentationTimeOffset, any event instance can be repeated with the same values for the presentation\_time and duration fields and identical values for scheme\_id\_uri, value, and id. Therefore, that DASHEventMessageBox can be exactly repeated in two or more segments, without any need for updating its fields.
2. Since DASHEventMessageBox(es) based on version 0 carry the presentation time relative to the segment's earliest presentation time (EPT), any event instance for repeating the same event must have a different presentation\_time\_delta field. If the event has started in the past, the event instance cannot signal that start time, since the offset is a non-negative number. In this case, the duration field needs to be updated to reflect the remaining active time of the event.
3. The DASHEventMessageBox(es) should be repeated in all segments that fall in their active duration.

### 6.2.2.4 Repeating the event in different Periods

Inband events from a previous Period that continue into the next Period should be repeated, including an InbandEventStream element in the MPD and DASHEventMessageBoxes in the media segments.

## 6.2.3 Event timing constraints

### 6.2.3.1 Early insertion of Inband Events

As part of the event scheme definition, the event scheme owner should define and publish the minimum in-advance offset to insert a DASHEventMessageBox relative to the intended event start time, such that the DASH client has enough time to dispatch and the Application has enough time to process the event and take the required operations before the event's start time. In this case, all inserted DASHEventMessageBox(es) shall be conforming to this offset.

If such in-advance offset is not defined, it is recommended that DASHEventMessageBox(es) are inserted at least 4 seconds before the corresponding media presentation time of the event's start time. For example, in the case of 2-second segments, the event is repeated in 2 segments prior to and the segment during which the event is starting. This practice avoids "missing" any events by the DASH client while enables the processing that may be required to activate the event by the application.

### 6.2.3.2 Early insertion of InbandEventStream Element

The InbandEventStream Element corresponding to the DASHEventMessageBox(es) should be present in the MPD before the actual DASHEventMessageBox (es) appear in the available segments.

DASHEventMessageBox(es) without a corresponding InbandEventStream element may be ignored by DASH clients.

### 6.2.3.3 The 'event\_duration' field

The event duration may have a significant meaning for the application, such as when the application uses the event duration in its logic. The repeated events when using version 0 may have different `event_duration` for any event that is already in its active interval.

DASHEventMessageBox version 1, the equivalent event instances that carry information for the same event can have the same `event_duration`.

### 6.2.4 Inband Event message\_data and payload size constraints

The content author should consider the impact of the size of DASHEventMessageBoxes on the representation actual bandwidth. The size and the frequency may have a significant impact on the required bandwidth for downloading segments.

### 6.2.5 MPD update constraints

The recommendations are as follows:

1. Only DASHEventMessageBox(es) that are listed in **InbandEventStream** would be parsed by the client.

NOTE: DASH events, such as MPD validity expiration and callback events also must be indicated in MPD to be processed by the client.

2. In an MPD update, a new **InbandEventStream** element may be introduced to request the DASH client start processing the corresponding inband emsgs.

### 6.2.6 Efficiency and Deployment Constraints

Following recommendations and constraints apply with regard to efficiency and deployment.

1. Only inband events relevant to the application should be included.
2. Inband events should be identical for different users.
3. An MPD update may add or remove an event scheme, thus only event schemes relevant for a given timespan should to be signalled.

## 6.3 Common use-cases

Some common use-cases for inband Events include the following:

1. Tracking events: These are events that are triggered on presentation time and can be used by an application to report viewing of certain presentation times in the media.
2. The id3 metadata, such metadata may convey authoring information valid at a specific point in time to the media.

## 7 MPD event authoring

### 7.1 General Properties and constraints

The MPD events, defined in the EventStream Element of MPEG-DASH [1] clause 5.10.2.2, have the following characteristics:

1. The EventStream Element is signalled inside a Period Element
  - a. The optional `@timescale` should be set, otherwise it is inferred to be 1.
  - b. `@schemeIdUri` shall be set to indicate the externally defined scheme, this scheme may be defined by an external organization.
  - c. The `@value` attribute is optional and can be used to signal a sub-scheme or URI relating to the `@schemeIdUri`

- d. The dispatch mode is defined by the scheme owner as part of the event scheme characteristics, the default is on-receive. The dispatch mode used by a DASH client is determined by the subscription of the client/player to an @schemeIdUri.
- e. The @presentationTimeOffset can be used to indicate the presentation time offset associated with the **Period**@start time of the Period enclosing the EventStream element, if not present it defaults to 0

Table 7-1 recommends the values for the EventStream attributes.

**Table 7- 1 Recommended usage of optional EventStream attributes**

@xlink:href	may be used for remote element
@xlink:actuate	Shall be set to 'onLoad' or absent
@schemeIdUri	Shall be set to scheme type of the event stream.
@value	May be absent, if present it shall be set according to guidelines of the scheme.
@timescale	Shall be set to the timescale of the <b>Event</b> @presentationTime and <b>Event</b> @duration
@presentationTimeOffset	Shall be set to a value that would align the @presentationTime of an <b>Event</b> element to the <b>Period</b> @start

2. The Event Elements are then enclosed in the EventStream Element and may be added/removed on MPD updates
  - a. The @duration attribute signals the duration of the Event
  - b. The @presentationTime signals the start time of the Event on the media presentation timeline based on the **EventStream**@timescale and **EventStream**@presentationTimeOffset attributes
  - c. The @id attribute is a unique identifier, it can be used to detect duplicates
  - d. The @contentEncoding may be present, and if present and set to base64 can be used to indicate that the data in the value space of the element is base64 encoded binary data. This is useful for encoding binary data that may include characters that would break the xml utf-8 schema
  - e. Data in the value space of the Event element needs to be character escaped in case characters occur that may break the schema, an alternative way to do that is to use the <!CDATA escaping mechanism available in XML
  - f. The timing of events is relative to the Period's start time adjusted for **EventStream**@presentationTimeOffset.
  - g. Event Elements of **EventStream** Elements may be updated (added/removed) by an update to the MPD.
  - h. The value space of the Event element is recommended to be used for carriage of event's payload rather than the use of @messageData.
  - i. The Latest Arrival Time (LAT) of an MPD Event may vary depending on whether the period is played from the beginning or random accessed, or if the event is added during an MPD update.

## 7.2 Authoring guidelines for MPD events

### 7.2.1 Dispatch mode selection

Same as 6.2.1.

## 7.2.2 Event instances order

Event elements in an **EventStream** element should be listed from top to bottom in MPD by increasing presentation time.

## 7.2.3 Event instances without @id

The DASH client dispatches any event instances without @id, since it can not use the equivalency rules to find the repeated event. Therefore, repeating such events without use of @id should be avoided.

## 7.2.4 Repeating events

### 7.2.4.1 Repeating event instances in one Period

Events in the same Period should not be repeated. Since the player sees all events of a Period, there is no benefit for repeating events in that Period.

### 7.2.4.2 Repeating the event cross Period

Events of the previous Period that continue into the next Period (i.e. are still active) should be repeated at the next Period. The next Period should include a corresponding EventStream and Event elements. The EventStream element of an event continuation should include an @presentationTimeOffset attribute.

## 7.2.5 Event timing considerations

The considerations are as follows:

1. An EventStream element may have an @presentationTimeOffset, an Event with a presentation time matching the value of the @presentationTimeOffset will align to the **Period@start**. By setting an event's presentation time to a value smaller than the @presentationTimeOffset, an event that starts before Period start time can be signalled.
2. In some common cases, the @presentationTimeOffset of an **EventStream** Element matches the @presentationTimeOffset of the Adaptation Sets the events apply to (after timescale conversion). It is recommended to use such common timeline for events and media.

## 7.2.6 Announcement and removal of Events in the MPD

The event scheme owner should define and publish the minimum in-advance offset insert an Event from the event start time, such that the DASH client has enough time to dispatch and the application has enough time to process the event and take the required operations before event's start time.

In case this offset is not defined, it is recommended to publish Events in the MPD at least 4 seconds before their target corresponding presentation time, or as early as possible.

In case the **MPD@type="dynamic"** the offset is relative to the **MPD@publishTime**. An MPD Event targeting a media time corresponding to a wall clocktime 'W' should be available in a manifest with **MPD@publishTime** 'W- the advance offset' or earlier. For the case when **MPD@type="static"** this restriction does not hold, as all MPD events will be available upon first download of the MPD.

MPD Events may be removed from an **EventStream** element in the MPD, when segments overlapping the event active duration are no longer available. Events with indefinite duration may be removed at any MPD update.

## 7.2.7 Event payload size constraints

MPD Events should only be used to carry metadata with small payloads and limited overhead on the MPD Size. Players usually have limits in the size of the MPD that they can handle e.g. 256 kb – 1MB have been reported, thus using MPD Events should not lead to violating such player limitations.

NOTE: If use of larger size metadata is required including them by reference or by means of a timed metadata track is preferred over including them as MPD Events.

## 7.2.8 Event payload character coding constraints

Restrictions apply when inserting characters in the value space of an Event. Events containing a payload that can break the XML encoding syntax of an MPD, e.g., unescaped characters "<>, / tab, newline ...", should either use `!CDATA` or use the `@contentEncoding` set to `base64` and apply a base64 encoding to the message data or value of the MPD event before setting it in the `value` or `message_data` fields of the MPD.

Event schemes that embed XML directly in the Event element must signal the namespace of this introduced XML schema.

## 7.2.9 Event and EventStream MPD update constraints

In an MPD update, the following attributes should not be changed for a previously existing **EventStream** in a period:

- a. **EventStream**@timescale
- b. **EventStream**@presentationTimeOffset
- c. **EventStream**@schemeIdUri
- d. **EventStream**@value

In an MPD update, the following attributes should not be changed for a previously existing **Event**:

- a. **Event**@presentationTime
- b. **Event**@contentEncoding
- c. **Event**@messageData
- d. **Event**@duration

## 7.2.10 MPD Event Efficiency and Deployment recommendations.

The following recommendations apply to enable efficient MPD Event usage:

1. MPD Events should only have limited impact of the size of the MPD.
2. To achieve the above and limit the size of the MPD one or more of the following techniques are recommended:
  - a. Use a smaller `@timeShiftBufferDepth`, thereby reducing the number of Periods in the MPD update
  - b. Use of remote period or remote EventStream for the periods of the time shift buffer that are older in MPD update
  - c. Use of MPD patching in MPD update
3. Since MPD events are included in the MPD, the frequency of updating the MPD depends on the frequency of generating new MPD events. Therefore, MPD events are not suited for high-frequency events when the minimum update period is high; inband events or time metadata tracks should be used instead in that case or a smaller minimum update period must be used.

## 7.3 Common use-cases

Common use cases of MPD Events include:

- Programme metadata such as programme information
- Ad slots and splice points signalling using SCTE-35 [i.1]

---

## 8 Combination of MPD events and Inband events

The same event stream, i.e. (`schemeIdUri`, `value`) pair may be present in both MPD (**EventStream**) and inband events (**InbandEventStream**) at the same time. In the case of the existence of the same `@schemeIdUri` and `@value` in **EventStream** and **InbandEventStream**, a DASH client may apply the equivalency rules between both.

## 9 Timed Metadata Representations

### 9.1 Characteristics of timed metadata Representations

Timed metadata representation carry tracks that are ISOBMFF metadata tracks (ISOBMFF [2] clause 12.3) where metadata is carried in samples similarly (i.e. as in media).

Metadata Representations are delivered in a similar fashion as media segments.

### 9.2 Authoring guidelines

#### 9.2.1 Dispatch mode selection

The default dispatch mode for timed metadata representations is on-start, i.e. on at the presentation time of the metadata samples. However, in-advance fetching and buffering of metadata samples in tracks, and dispatching them using on-receive dispatch mode is possible for the client.

#### 9.2.2 Metadata sample timing considerations

All samples are sync-samples, thus no knowledge of prior samples is required.

Timed metadata track used in a Representation may use an edit list and/or composition time offsets.

Timed metadata track representation timing is relative the **Period@start** adjusted with the **SegmentBase@presentationTimeOffset** or **SegmentTemplate@presentationTimeOffset**.

#### 9.2.3 Payload considerations

Metadata samples shall contain all metadata active or relevant for the time interval their duration spans, for the given scheme of the metadata track.

#### 9.2.4 Scheme signaling and Sample Entry Requirements

Custom metadata schemes can be signalled in a metadata track using **URIMetaSampleEntry**. Other sampleEntries derived from **MetaDataSampleEntry** may also be used.

#### 9.2.5 MPD signaling

Timed metadata tracks shall be signalled as follows:

1. A single representation in a single adaptation set shall be used.
2. Language may be defined by **@lang** if language is relevant for the timed metadata
3. An additional descriptor at the adaptation set or representation, in a supplemental or essential property, with the following attributes:
  - a. **@schemeIdUri** set to “urn:dashif:events:metadataconfiguration:2022”
  - b. **@value** set to the list of white space separated metadata scheme id URI and value pairs
4. The **@mimeType** attribute set to **application/mp4**
5. The **@contentType** attribute set to “meta”
6. The **@codecs** attribute set to the four-character code of the Sample entry of the timed metadata tracks, e.g. in the case of **URIMetaSampleEntry** this would be “urim”.
7. The **@timescale** attribute that matches the timescale in the **mdhd** box in the initialization segment
8. The timed metadata track should be continuous, i.e. not have gaps on the timeline.
9. A Role element is optional, if present it shall not be set to **main**
10. The **@bandwidth** attribute set to a value higher than the maximum bandwidth of any timed metadata segment
11. The **@associationId** attribute may be set to the list of **Representation@ids** to which the metadata track applies.

Example 9-1 shows an example of a metadata track **AdaptationSet** element for a track using **URIMetaSampleEntry**.



### Example 9- 1 Example for describing of a metadata track in MPD

```

<AdaptationSet codecs="urim" contentType="meta" mimeType="application/mp4" id="1"
lang="en" startWithSAP="1">
  <SupplementalProperty
schemeIdUri="urn:dashif:events:metadataconfiguration:2022"
value="urn:scte:scte35:2013:bin urn:mpeg:dash:event:2012 " />
  <Representation bandwidth="8000" id="meta-track" associationId="1 2 4" >
    <SegmentTemplate initialization="segments-$RepresentationID$.dash"
media="segments-$RepresentationID$-$Time$.dash" timescale="1000">
      <SegmentTimeline>
        <S d="2000" t="0" r="29"/>
      </SegmentTimeline>
    </SegmentTemplate>
  </Representation>
</AdaptationSet>

```

## 9.2.6 Metadata samples carrying the same event instances

In an ISO-BMFF metadata track each sample carries the active metadata for that timespan. In case metadata is repeated in samples, ISO-BMFF metadata tracks typically do not carry ids for the events that they carry to detect duplicate samples. To implement such, it may be covered by the specific scheme that is used in the timed metadata track.

MPEG-B part 18 [4] defines a metadata track format that can be used to carry additional ids to detect duplicated ISO/IEC 23001-18 spanning multiple samples. In case duplicate event detection is required, metadata track based on MPEG-B part 18 may be considered.

MPEG-B part 18 [4] based timed metadata track uses the event message instance box (EventMessageInstanceBox) (emib) to carry event messages in samples, and the EventMessageEmptyBox (emeb) for durations of samples when no event metadata is active. The emib box has fields that correspond to DASHEventMessageBox, namely scheme\_id\_uri, id, value, event\_duration and presentation\_time\_delta, and can therefore be used directly for carriage of DASH events in timed metadata tracks.

In an EventMessageInstanceBox the timing is relative to the presentation time of the sample carrying the EventMessageInstanceBox. As the presentation\_time\_delta is a 64 bit signed integer both advance and past events can be signalled.

Example 9-2 demonstrates an example of a metadata track AdaptationSet Element for a metadata track using Event Message Track based on MPEG-B part 18 [4].

### Example 9- 2 Example for describing of Event Message Track in MPD

```

<AdaptationSet codecs="evte" contentType="meta" mimeType="application/mp4" id="1"
startWithSAP="1">
  <SupplementalProperty
schemeIdUri="urn:dashif:events:metadataconfiguration:2022"
value="urn:scte:scte35:2013:bin " />
  <Representation bandwidth="8000" id="meta-track">
    <SegmentTemplate initialization="segments-$RepresentationID$.dash"
media="segments-$RepresentationID$-$Time$.dash" timescale="1000">
      <SegmentTimeline>
        <S d="2000" t="0" r="29"/>
      </SegmentTimeline>
    </SegmentTemplate>
  </Representation>
</AdaptationSet>

```

## 9.2.7 Application processing considerations

Timed metadata track segments should be available before the audio and video content with similar presentation time.

## 9.3 Common use-cases

As mentioned, timed metadata tracks are beneficial for frequent metadata (e.g. positional coordinates for each frame based on capture location) or media related metrics.

---

## 10 Support of DASH events

Table 10-1 show the DASH events [1] which shall be supported by the client.

**Table 10 - 1 DASH events supported by the DASH client**

Event scheme	Referenrece in MPEG-DASH [1]	Dispatch mode
urn:mpeg:dash:event:2012	5.10.4	on-receive
urn:mpeg:dash:event:callback:2015	5.10.4.5	on-start
urn:mpeg:dash:event:tfn:2016	5.10.4.6	on-start

---

## 11 Externally defined event schemes

These event schemes are cataloged at [https://dashif.org/identifiers/event\\_schemes/](https://dashif.org/identifiers/event_schemes/).

---

## Annex A (Informative): Change History

Date	Version	Information about changes
2022-05	0.9	Initial version for community review
2022-12	0.91	Submitted for final approval and publication
2023-01	5.0.0	Published version on January 26, 2023