

**ISO/IEC JTC 1/SC 29/WG 11 N 18854**

**ISO/IEC JTC 1/SC 29/WG 11**

**Coding of moving pictures and audio**

**Convenorship: UNI (Italy)**

**Document type: Approved WG 11 document**

**Title: WD of EventMessage track format**

**Status: Approved**

**Date of document: 2019-10-17**

**Source: Systems**

**Expected action:**

**No. of pages: 9**

**Email of convenor: leonardo@chiariglione.org**

**Committee URL: mpeg.chiariglione.org**

**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC1/SC29/WG11**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11 MPEG128/** n**18854**

**Oct 2019, Geneva, CH**

|  |  |
| --- | --- |
| **Source** | **Systems Group** |
| **Status** | **Approved** |
| **Title** | **WD of Event Message track format** |
|  |  |

# Introduction

The DASHEventMessageBox, defined in ISO/IEC 23009-1, is a box structure that may arise in streaming applications. The fact that it is a top-level box and has its own timeline makes it tricky to work with in ISOBMFF formatted media files, such as those based on the Common Media Track Format (CMAF) [2]. For instance, identifying active events at any point in the media track may require scanning of a large sub part of that track. In addition, it is not clear what happens to DashEventMessageBoxes when tracks are de-fragmented. Further, DashEventMessageBoxes cannot be de-multiplexed from track files based on ISO/IEC 23009-1 or ISO/IEC 23000-19.

This text specifies alternative methods for carriage of DashEventMessageBox structures using ISOBMFF Timed metadata track format. This event message track format associates the timeline of the DashEventMessageBox to the ISOBMFF track timeline. The specified track format enables all common ISOBMFF processing such as multiplexing and de-fragmentation. In addition, multiplexing and de-multiplexing operations using top-level DashEventMessageBox based on this event message track format are defined. The carriage of DASHEventMessageBoxes in the event message track format will also make this information more easily accessible to devices that can seek through ISOBMFF formatted media files.

The event message track format is defined in the following clauses.

## DashEvent MessageBox

In CMAF Track files and DASH segments DASHEventMessageBoxes may occur as top-level boxes as defined in ISO/IEC 23009-1[1] and ISO/IEC 23000-19[2]. The DASHEventMessageBox arises in two forms as defined in ISO/IEC 23009-1:

aligned(8) class DASHEventMessageBox extends FullBox('emsg', version, flags = 0){  
 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[];  
}

The semantics as per ISO/23009-1 clause 5.10.3.3.4 are as follows:

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 time delta and duration fields within version 0 of this box;

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.

event\_duration provides the duration of event in media presentation time. In version 0-, the timescale is indicated in the timescale field; in version 1, the timescale of the MovieHeaderBox is used. The value 0xFFFFFFFF indicates an unknown duration.

id: a field identifying this instance of the message. Messages with equivalent semantics shall have the same value, 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.

# Event Message Track format

DashEventMessageBoxes shall be carried in ISOBMFF timed metadata tracks as defined in [3] clause 12.3, and as a consequence, use the 'meta' media handler type, and the associated media header (‘nmhd’). DashEventMessageBox structures are carried in samples as defined by the ISOBaseMediaFileFormat, which means carriage in the MovieData box (mdat).

## Sample Entry Format

Tracks shall use the URIMetaSampleEntry (‘urim’). In the sample entry, a URIBox must occur, carrying the exact urn scheme:

urn:mpeg:dash:event:2019

NOTE: an alternative scheme urn may be defined

The different scheme\_id\_uri used in the different DashEventMessageBoxes may be signalled in one or more kind boxes using a schemeIdUri urn:mpeg:dash:event:2019 and the value field of the kind box may be used to signal the schemes signalled in the DashEventMessageBoxes.

## Sample Format

Each ISOBMFF sample in the metadata track may contain one or more DashEventMessageBox structures.

Samples may contain a single EventMessageBoxEmptyCue, in this case, the sample contains no DashEventMessageBox structures, and no events are active during the sample presentation time.

This clause defines an EventMessageBoxEmptyCue:

aligned(8) class EventMessageBoxEmptyCue extends Box('embe')

{

}

The EventMessageBoxEmptyCue is defined to signal duration when no event is active, with the following semantics. The box is empty, the duration and presentation time are defined by the sample that carries this box structure.

NOTE: empty samples to signal when no event metadata or event is active are avoided because this may cause problems in some devices.

## Timing

Following the general time processing in ISOBMFF tracks, each DashEventMessageBox in a sample shall be passed at the time from the time-to-sample table, as mapped by the edit list (if any), by the timing of the MovieFragment enclosing the samples defined in the TrackFragmentRunBox, or by the time the top level DashEventMessageBox is parsed, in case of a top level DashEventMessageBox as per ISO/IEC 23009-1.

Any sample with a presentation time and duration, must contain all DashEventMessageBoxes active in the timespan from the presentation time up to but not including presentation time plus duration.

Samples shall only contain DashEventMessageBoxes that start at or are active on the respective ISOBMFF sample presentation\_time.

DashEventMessageBoxes with a duration extending multiple samples, must be carried in each of these samples.

The duplicate carriage of DashEventMessageBox in different samples avoids dependencies between samples and enables seeking in tracks based on a single sample. Duplicate DashEventMessageBoxes have the same scheme\_id\_uri, id and value fields. Such duplicates can thus be detected easily and ignored if they have already been processed before.

Duplicate carriage cases should only occur in the case DashEventMessageBoxes are overlapping in the track. If this is not the case, duplicate carriage of DashEventMessageBoxes shall be avoided.

There are internal timing values in a DashEventMessageBox, namely the presentation\_time, timescale, presentation\_time\_delta and the duration. These timing structures relate to the presentation time and duration of the ISOBMFF sample in the following way.

1. The DashEventMessageBox.timescale SHALL match the timescale defined in the MovieHeaderBox, in case of differing timescales, conversion of the DashEventMessageBox must be applied.

Note: this matching is always the case for CMAF [2] tracks

1. In case a DashEventMessageBox is version 0, it must be converted to a version 1 box. This can be done by computing the presentation\_time by adding the earliest presentation time of the subsequent fragment and the presentation\_time\_delta.

Boxes of version 0 relate to tfdt box which may not be present in some tracks (in case fragmentation is not used for example).

1. If the sample is the first sample containing one or more specific instances of a DashEventMessageBox, the sample presentation time must equal the DashEventMessageBox.presentation\_time for each of these specific instances
2. In case there is no subsequent DashEventMessageBox to be carried after all DashEventMessageBoxes enclosed in the current sample cease to be active, the ISOBMFF sample duration MUST equal the largest duration at which all DashEventMessageBoxes enclosed in the sample cease to be active, given that this duration is not zero or unknown (in that case 6 applies). The next sample may then, in addition, contain an EventMessageBoxEmptyCue to signal a timespan at which no event is active.
3. In case there are one or more subsequent DashEventMessageBoxes to be carried before all of the enclosed DashEventMessageBoxes cease to be active, the sample duration shall equal the difference between the presentation\_time of the current sample and first subsequent DashEventMessageBox carried in a subsequent sample, that was not active in the prior sample.
4. If the DashEventMessageBox.duration is zero or unknown, the sample duration shall equal the difference between the presentation\_time of the current sample and a subsequent new instance DashEventMessageBox carried in a next sample. In case there is no next sample, the duration may be set to an arbitrary value or zero and updated once it becomes available.
5. If the sample is not the first sample containing one or more specific instances of a DashEventMessageBox, the sample presentation time and the DashEventMessageBox.presentation\_time of these specific instances are different. This occurs when a DashEventMessageBox carried in a prior sample is still active in a subsequent sample that also introduced new other DashEventMessageBox instances. NB: This follows from prior clauses and happens in case of overlapping events.

Note: In this case the presentation\_time in the DashEventMessageBox is smaller than the sample presentation time.

Note: in case relative timing would be considered relating to the sample presentation this would lead to negative offset signalling.

Note: such negative offsets are not supported in current DashEventMessageBox.

1. One or more samples carrying EventMessageBoxEmptyCue shall be used to cover timespans where no event is active.

# Processing

The following processing operations are defined for the Event Message Track format.

## 3.1 Client processing

The semantics of the DashEventMessageBox shall be used when processing the samples, as per ISO/23009-1 clause 5.10.3.3.4. Briefly summarized they are as follows:

1. i.e. scheme\_id\_uri signals the scheme of the message,
2. id and value can be used to detect duplicate messages and signal sub-schemes
3. presentation\_time signals presentation time of event (should be equal to presentation time of sample, or smaller in case the event was already active in a previous sample)
4. timescale signals timescale of the event, shall be equal to the timescale of the ISOBMFF track
5. event\_duration signals the actual duration of the event (may be longer than sample duration), and may also signal indefinite duration
6. message\_data contains the binary payload of the message

Samples can be acquired and then passed to application as defined by DASH processing model for events and timed metadata defined in ISO/IEC 23009-1 [1].

## 3.2 General processing

The following general processing is defined:

De-multiplexing a CMAF track file with top level DashEventMessageBoxes to an event message track in Clause 3.2.1

Multiplexing of an event message track with a CMAF track file resulting in a CMAF Track file with top level DashEventMessageBoxes Clause 3.2.2.

Fragmentation and de-fragmentation of the event message track. 3.2.3.

### 3.2.1 Demultiplex a CMAF track file with top level DASHEventMessageBoxes

The following steps convert a CMAF Track file with top level DashEventMessageBoxes (e.g. a CMAF file based on ISO/IEC 23000-19:2018) to a separate event message track.

NOTE: Timescale of DashEventMessageBox in a CMAF track file is equal to the timescale defined in the MediaHeaderBox.

NOTE: For the case where events do not overlap or have zero/indefinite duration the processing model will be simplified.

NOTE: There are no strict rules for order of DashEventMessageBox in a CMAF track file, this is why the entire track file is scanned first in this processing model.

Table 1 Demultiplex a CMAF Track file with top level boxes

|  |
| --- |
| Input(CMAF Track File in\_cmaf) Output(track out\_meta)   1. Initialize out\_meta as an ISOBMFF metadata track with constraints defined in 2.1, the MovieHeaderBox is set to the timescale defined in the MediaHeaderBox of the source CMAF Track file, the sampleEntry is URIMetaSampleEntry, the URIBox contains urn:mpeg:dash:event:2019 2. Initialize a Set Event\_Boxes to hold DashEventMessageBoxes structures (‘emsg’) 3. Read in\_cmaf, fragment by fragment, if no more fragments continue to 5 4. In each case a DashEventMessageBox (‘emsg’) is detected do:    1. If version 0, convert to version 1 by setting presentation\_time to presentation\_time\_delta + earliest\_presentation\_time of the CMAF fragment   Add the ‘emsg’ to Event\_Boxes, continue to 3)   1. Sort Event\_Boxes by increasing presentation\_time 2. For each ‘emsg’ in Event\_Boxes do:    1. If no sample with emsg.presentation time exist in out\_meta, add a sample with presentation\_time equal to emsg.presentation\_time and duration emsg.duration to out\_meta. In case emsg.duration is zero or indefinite set the sample duration to the difference between the next emsg with presentation\_time greater than the current emsg and the current emsg.presentation\_time. If there is no such next emsg, the duration may be set to 0 or an arbitrary value and updated later once this next emsg becomes available.   In case the presentation\_time of the new sample would overlap a prior sample in out\_meta, update the prior sample duration to the difference of the presentation time of the emsg and the presentation time of that prior sample and include any one or more ‘emsg’ from the prior sample that are still active in the added sample.  In case adding a sample to out\_meta would result in a timeline gap, as the presentation\_time of the added sample would be greater than the presentation time of the prior sample plus its duration, an intermediate sample containing a DashEventMessageBoxEmptyCue MUST be added with the duration to fill that gap.   * 1. If a sample already exist at that presentation time, update the sample to also include this `emsg`. The duration of the sample will be updated to the duration at which all enclosed ‘emsg’ events would cease to be active. In case all emsg have duration zero or are indefinite the duration MUST be set to zero or an arbitrary duration, and updated once the next emsg with presentation time greater than the current presentation\_time becomes available. |

This processing model results in event message tracks carrying DashEventMessageBoxes using ISOBMFF samples. In addition the kind box may be used to signal each of the schemes used in the track (note: this step was not included in the processing model).

### 3.2.1 Multiplex a CMAF track file with top level DASHEventMessageBoxes

A CMAF track [2] may contain one or more DashEventMessageBoxes. In a live stream these may contain signalling relevant for the broadcast. This clause describes the processing of inserting DashEventMessageBox in a CMAF track using an event message track. The announce\_time is the amount of time a DASHEventMessageBox should be included before the basemedia decode time of the fragment it applies to.

NOTE: Announce\_time is a variable introduced for this processing operation. It is an approximation of the time in advance a DashEventMessageBox should be inserted.

In this case it is assumed that both the CMAF Track file and the event message track have a common timeline (time line origin and timescale). The content of the metadata tack is multiplexed in the CMAF track file using top level boxes.

Table 2 Demultiplex a CMAF Track file with top level boxes

|  |
| --- |
| Input (CMAF Track File in\_cmaf, EventTrack in\_meta, announce\_time )  Output(CMAF Track file out\_cmaf)   1. Initialize out\_cmaf   For each CMAF Fragment frag in in\_cmaf   1. Load the Earliest presentation time (frag.ept) and duration (frag.dur) of frag. 2. Load samples from the in\_meta corresponding to time interval starting at frag.ept + announce\_time with duration frag.dur. Extract all DashEventMessageBoxes enclosed in these samples and copy each unique DashEventMessageBox as a top level box in the out\_cmaf, copy the frag to out\_cmaf. |

NOTE: This processing model may lead to more identical DashEventMessageBoxes being inserted in front of different fragments. Duplicate DashEventMessageBoxes can be detected as being duplicate based on the value, id and schemeIdUri. Alternative processing to avoid duplicate event message boxes may be defined. However, duplicate event message boxes do not introduce harm, and are often needed for redundant delivery of events.

### 3 Fragmentation and de-fragmentation of tracks carrying DashEventMessageBoxes

All samples in the event message track are sync samples. Event message tracks can therefore be fragmented in case desirable (this may be useful for some delivery schemes or protocols or to deliver the track piece by piece). In case samples with long durations exist where no metadata is defined, signalled by EventMessageBoxEmptyCue, such cues may be signalled using multiple samples each carrying this cue. In addition, samples carrying DashEventMessageBoxes with long durations may be broken into two or more samples carrying the same payload as to facilitate fragmentation. Fragmented tracks can also be de-fragmented, in this case contiguous samples should be merged if they are identical.

# Conclusion

The event message track format and processing model was defined in this text. The track format makes it easier to handle DashEventMessageBoxes by providing:

1. Carriage in ISOBMFF compliant timed metadata tracks
2. Processing model for De-multiplexing and multiplexing event message tracks from and to CMAF tracks files
3. Fragmentation and de-fragmentation of the event message track
4. Timing structure similar to MPEG-4 part 30 [4]

# Technologies under consideration

Several aspects under consideration for event message track:

1. Relative timing: DashEventMessageBox fields support absolute (v1) and relative (v0) timing. The current text only considers v1 absolute timing. In some cases relative timing may be considered where timing is relative to sample presentation time. *However, in case of overlapping events this may lead to negative offsets. This is currently not supported by the DashEventMessageBox structure which can only signal positive numbers. Converting may not be desirable.*
2. It was considered that by using fragments with only 1 sample v0 dash events could be kept. *Does it make sense to have such restrictions at ISO BMFF level ?*
3. Currently new events trigger sample boundary. In case of overlapping events some events may end during a sample duration, should this also trigger a sample boundary ?

# References

[1] ISO/IEC 23009-1:2019 Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats

[2] ISO/IEC 23000-19:2018 Information technology — Multimedia application format (MPEG-A) — Part 19: Common media application format (CMAF) for segmented media

[3] ISO/IEC 14496-12:2015 Information technology -- Coding of audio-visual objects -- Part 12: ISO base media file format

[4] ISO/IEC 14496-30:2014 Information technology — Coding of audio-visual objects — Part 30: Timed text and other visual overlays in ISO base media file format