**** **ISO/IEC JTC 1/SC 29/WG 3 N0660**

**ISO/IEC JTC 1/SC 29/WG 3**

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

**Document type:** Output Document

**Title: WD Encoder and Packager Synchronization**

**Status:** Approved

**Date of document:** 2022-07-22

**Source:** ISO/IEC JTC 1/SC 29/WG 3

1. **Expected action:** ACT

**Action due date:** 2022-07-22

**No. of pages:** 2 (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 3**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC 1/SC 29/WG 3 N** **00660**

**Online – July 2022**

**ISO 23009-9:2022**

ISO JTC 1/SC 29/WG 3

Date: 2022-07-22

**Title** Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 9: Encoder and Packager synchronization, multimedia ingest and storage

WD stage

**Warning for WDs and CDs**

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

© ISO 2022

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO’s member body in the country of the requester.

ISO copyright office

CP 401 • Ch. de Blandonnet 8

CH-1214 Vernier, Geneva

Phone: +41 22 749 01 11

Email: copyright@iso.org

Website: www.iso.org

Published in Switzerland

Contents

[Foreword vi](#_Toc103345929)

[Introduction vii](#_Toc103345930)

[1 Scope 1](#_Toc103345931)

[2 Normative references *(mandatory)* 1](#_Toc103345932)

[3 Terms and definitions *(mandatory)* 1](#_Toc103345933)

[3.1 Terms 1](#_Toc103345934)

[3.1.1 Media Stream 1](#_Toc103345935)

[3.1.2 Encoder 1](#_Toc103345936)

[3.1.3 Packager 1](#_Toc103345937)

[3.1.4 Origin 1](#_Toc103345938)

[3.1.5 Unix Epoch 1](#_Toc103345939)

[3.1.6 Ceil() 1](#_Toc103345940)

[3.1.7 Now 1](#_Toc103345941)

[3.2 Abbreviated terms 2](#_Toc103345942)

[4 Document organization 2](#_Toc103345943)

[5 Example workflow model for encoder and packager synchronization 3](#_Toc103345944)

[6 Example workflow model for asset storage and recording 4](#_Toc103345945)

[7 Encoder synchronization 4](#_Toc103345946)

[7.1 Encoder and origin/packager communication 4](#_Toc103345947)

[7.2 Constraints on the Ingest Media Presentation Description (I-MPD) and track format 5](#_Toc103345948)

[7.3 Constraints on the track and segment format 6](#_Toc103345949)

[7.4 Method for encoder synchronization 6](#_Toc103345950)

[8 Asset storage and recording Format 8](#_Toc103345951)

[8.1 Track format for storage of live archives 8](#_Toc103345952)

[8.2 Storage Media Presentation Description (S-MPD) 10](#_Toc103345953)

[8.2.1 Overview 10](#_Toc103345954)

[8.2.2 S-MPD for 24x7 live archiving and recording 11](#_Toc103345955)

[8.2.3 Generating an S-MPD from an I-MPD 12](#_Toc103345956)

[Annex A Example Media Presentation Description 14](#_Toc103345957)

[A.1 Example I-MPD (informative) 14](#_Toc103345958)

[A.2 Example S-MPD (informative) 16](#_Toc103345959)

[Annex B Recommended Configurations 17](#_Toc103345960)

[B.1 Recommended segment durations for encoder synchronization (informative) 17](#_Toc103345961)

[Annex C Example Applications 18](#_Toc103345962)

[C.1 Redundant and high availability distribution: 18](#_Toc103345963)

[C.2 Distributed encoding: 18](#_Toc103345964)

[C.3 A/B watermarking: 18](#_Toc103345965)

[Bibliography 19](#_Toc103345966)

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see [www.iso.org/directives](https://www.iso.org/directives-and-policies.html)).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see [www.iso.org/patents](https://www.iso.org/iso-standards-and-patents.html)).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see [www.iso.org/iso/foreword.html](https://www.iso.org/foreword-supplementary-information.html).

This document was prepared by Technical Committee ISO/TC *[or ISO/PC]* JTC1, *[name of committee]*, Subcommittee SC 29.

A list of all parts in the ISO 23009 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at [www.iso.org/members.html](https://www.iso.org/members.html).

Introduction

This standard specifies a method of synchronizing encoders and packagers in a stateless manner. The method is intended to be used to create media segments that are interchangeable based on a common timing mechanism.

This standard also specifies a method and related formats for large-scale multimedia asset storage, recording and archiving. This method includes support for 24x7 live recording and archiving.

The International Organization for Standardization (ISO) draws attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO takes no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO. Information may be obtained from the patent database available at [www.iso.org/patents](http://www.iso.org/patents).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. ISO shall not be held responsible for identifying any or all such patent rights.

# Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 9: Encoder and Packager synchronization, multimedia ingestion and storage

# 1 Scope

This document defines methods and media formats for encoder and packager synchronization. Related and derived formats for media ingest and large-scale distributed asset storage and recording are also defined in this document.

# Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 14496-12, *Information technology — Coding of audio-visual objects — Part 12: ISO base media file format*

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

ISO/IEC 23000-19, *Information technology — Multimedia application format (MPEG-A) — Part 19: Common media application format (CMAF) for segmented media*

# Terms and definitions

For the purposes of this document, the terms and definitions given in [1], [2], [3] and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

* ISO Online browsing platform: available at <https://www.iso.org/obp>
* IEC Electropedia: available at <https://www.electropedia.org/>

## Terms

### Media Stream

Media presentation, e.g. a CMAF presentation, a DASH Media Presentation or a media presentation based on the ISO base media file format.

### Encoder

Entity or computational unit used to encode or transcode an input to one or more outputs in one or more bitrates.

### Packager

Entity or computational unit used to package an input to an ISO BMFF track format and/or media streaming format.

### Origin

Entity or computational unit used to serve media content based on HTTP requests.

### Unix Epoch

Seconds elapsed since 00:00:00 UTC on January 1st 1970 (excluding leap seconds).

### Ceil()

Nearest integer larger than or equal to the input number.

### Now

Current time in seconds relative to 00:00:00 UTC on January 1st 1970 (excluding leap seconds), retrieved on a real-time system.

## Abbreviated terms

The following abbreviated terms are used in this document

DASH-IF DASH Industry Forum.

IETF Internet Engineering Task Force

CMAF Common Media Application Format as defined in ISO/IEC 23000-19.

DASH MPEG-DASH as defined in ISO/IEC 23009-1.

HLS HTTP Live Streaming as defined in IETF RFC 8216.

UUID Universally Unique Identifier

ID Identifier

MPD Media Presentation Description

I-MPD Ingest Media Presentation Description

S-MPD Storage Media Presentation Description

# Document organization

This specification is organized as follows: Clause 1 to 4 provide the introduction, scope, references, definitions, functions and abbreviated terms.

Clause 5 describes an example workflow assumed for encoder and packager synchronization and live media ingest.

Clause 6 describes an example workflow assumed for multimedia asset storage and recording.

Clause 7 specifies the mechanism used for encoder and packager synchronization, the required constraints on the MPD, the required constraints on the track and segment format as well as the method for encoder synchronization.

Clause 8 specifies the track format constraints for storage and the required constraints on the MPD for multimedia asset storage.

Annex A provides example media presentation descriptions for the ingest and storage applications.

Annex B provides informative recommendations for segment durations and configurations.

Annex C provides example and intended applications.

# Example workflow model for encoder and packager synchronization



1. An example workflow for encoder and packager synchronization. The encoders produce CMAF media segments whose boundaries are calculated based on the method described in this document. The epoch and D parameters are set to be the same for synchronized encoders

Figure 1 illustrates an example of using the specified method with more than one encoder to achieve a redundant workflow. The epoch reference and segment duration D are set as part of the encoder configuration. Encoders that are intended to have synchronized output shall have the same values set for these parameters. The example workflow shows the output of the encoders: media segments carrying the same label S1 S2.. are interchangeable. The segments are ingested by the origin or packagers and the output of these entities in Figure 1 is a streaming format such as DASH or HLS.

# Example workflow model for asset storage and recording



1. An example workflow for asset recording and archiving. In addition to the synchronized and redundant encoding, as in Figure 1, the workflow includes storing and archiving content in a cloud storage. The cloud storage may be accessed from the encoder, origin/packager or a client retrieving the media stream.

Figure 2 illustrates an example of using the specified method including cloud storage. Media streams can be written to the cloud storge from the encoder, from the origin/packager or from a client receiving the media content. The format for asset storage and recording enables efficient and interoperable writing of content to a cloud storage, and efficient retrieval later on.

# Encoder synchronization

## Encoder to origin/packager ingest

This specification assumes that segments of media data are transmitted from an encoder to a packager/origin, and optionally a constrained manifest to describe the groupings of tracks. Protocol details are out of scope of this specification. The process and formats used for this are referred to as media ingest in this document.

A general sequence of communication between the encoder and origin packager is as follows:

1. The receiver packager sets up a URL endpoint that a sender encoder can send requests towards.
2. The sender encoder authenticates, connects, and sends a constrained manifest, ingest media presentation description (I-MPD) and/or initialization segments for each Representation. The I-MPD is mainly used to define the groupings using AdaptationSets and the segment URL naming via SegmentTemplate elements.
3. The receiver origin/packager reads the I-MPD. When the CMAF profile for DASH is used, or when content is formatted according to the CMAF track format, all Representations map to CMAF tracks, and the adaptation sets will map to CMAF switching sets.
4. The sender sends each media segment using a separate HTTP request using the HTTP POST or PUT method. The URL is derived from URL endpoint combined with the segment name derived from the SegmentTemplate Element.
5. The receiver appends the media segments to the corresponding initialization segment and stores these in memory.
6. Each representation/track uses a separate TCP connection.
7. Video, audio, text/subtitle and metadata CMAF/ISOBMFF tracks are all supported.
8. Retransmission of media segments in a track/representation may happen in case of failures. In such cases it is also possible to retransmit the initialization segment.
9. The media segment's earliest presentation time is used as a “key” of segments in a track, the track name identified by Representation@id is used as a key for the track. Using these keys, interchangeable segments for different tracks from different senders can be identified by the receiver.
10. Optionally the sequence number in the MovieFragmentHeaderBox tracks the segment numbering

Some additional recommendations apply:

1. For encoder synchronization media segments should be of a constant duration **D**
2. In case of splicing, when a segment is created with duration A instead, the next media segment shall be of duration **2 x D - A** to keep the numbering and the number of segments since epoch. Alternatively, splicing can be achieved by inserting an IDR within the media segment and not creating a segment boundary.
3. In case of loss of encoder input filling segments with filler content should be inserted, to keep the track continuity. The SegmentTypeBox with the ‘slat’ brand may be used to indicate segments include artificial filler content.
4. Signalling the end of a track/representation may be achieved by inserting the ‘lmsg’

## Constraints on the Ingest Media Presentation Description (I-MPD) and track format

Before the segment exchange, an I-MPD is transmitted to the receiver. The I-MPD is used to establish the naming convention of the segment URLs and switching set groupings. SegmentTemplate@media and SegmentTemplate@initialization are used to define the segment URLs to be used by the sender. Tracks/segments can then be stored as compliant ISO-BMFF/CMAF track files by a receiver. Also, based on this information, a receiver can optionally produce the customized delivery manifest (D-MPD) for transmitting the content to a client.

The I-MPD conforms to ISO/IEC 23009-1. Additional constraints on the I-MPD are the following:

1. The I-MPD shall use a SegmentTemplate element in each AdaptationSet element
2. The SegmentTemplate@initialization in the I-MPD shall contain a single substring $RepresentationID$ and the SegmentTemplate@media shall contain a single substring $RepresentationID$ and a single substring $Number$ or $Time$ (not both). A separator character (- or \_) shall be between $Representation$ and $Number$ or $Time$ if one or more Representation@id end with a number.
3. SegmentTemplate@media shall be identical for each SegmentTemplate element
4. SegmentTemplate@initialization shall be identical for each SegmentTemplate element
5. The BaseURL element shall be absent.
6. The @availabilityStartTime should be set to “1970-01-01T00:00:00Z” and the Period @start should be set to PT0S or a semantically equivalent value.
7. Each representation in the I-MPD represents a CMAF track, each AdaptationSet in the I-MPD represents a CMAF switching set.
8. In case an ingest source issues an HTTP request with an updated I-MPD, identical naming conventions shall apply.
9. The I-MPD shall contain a single Period element.
10. The SegmentTimeline element may be empty.
11. The I-MPD should follow the CMAF profile for DASH indicated by the uri: “urn:mpeg:dash:profile:cmaf:2019” and the ISO live profile of DASH “urn:mpeg:dash:profile:isoff-live:2011”

## Constraints on the track and segment format

Entities conforming to this specification shall produce media segments in conformance with ISO/IEC 23000-19 (CMAF) clause 7 or as described in ISO/IEC 23009-1 (DASH).

In the case of CMAF, conforming entities shall apply the track synchronization model specified in clause 6 of ISO/IEC 23000-19. Conforming encoders shall use the Unix epoch the reference, therefore media presentation times shall be relative to 00:00:00 on January 1st 1970 excluding leap seconds.

Encoder entities conforming to this specification that produce CMAF or DASH media segments shall generate MovieFragmentBoxes (moof) and optionally SegmentTypeBox(es) with the following constraints:

1. The SegmentType box may contain a ‘slat’ brand in case the input of the encoder was missing frames and the encoder filled the gap with filler frames. In case the SegmentTypeBox contains a ‘slat’ brand, it indicates that one or more frames in the segment were replaced with filler content to avoid discontinuities.
2. In case the segment is the last segment, the SegmentTypeBox should contain the ‘lmsg’ brand. If the SegmentTypeBox contains the `lmsg` brand it is the last segment in the track.
3. The MovieFragmentHeaderBox shall contain the segment sequence number (K).
4. The TrackFragmentDecodeTime box shall contain a baseMediaDecodeTime that is equal to K\*D\*track\_timescale.

## Method for encoder synchronization

The method targets a workflow as depicted in Figure 1. The encoders are expected to receive an input signal with common per frame timing information. The assumption is that the timing information can be mapped back be relative to 1-1-1970 00:00:00 UTC excluding leap seconds as a common time anchor. If internal timing is not relative to such a common anchor, then a Synchronization Time Stamp (STS) is required to map input timestamps to output timestamps relative to this anchor.

The STS (Synchronization Time Stamp) is the difference between the zero time of the input signal to 1-1-1970 00:00:00 UTC excluding leap seconds.

The output frame time is calculated using:

*(1)*

Frame times lie on a continuous timeline on the track. In some cases, the input time will need to be converted from its original form to a continuous form that also matches the output timescale to derive .

In case the *track\_timescale* and the input timescale are not the same, the input time should also be adjusted for this by a timescale conversion. In this case the original input time is also multiplied by the *track\_timescale* and later divided by the original (input) timescale. In this case it is important to avoid rounding error and encoders should only choose integer multiples of the input timescale as the output .

The STS may not directly obtained from the encoder configuration, but may, for example, be retrieved from timing markers or other metadata markers (e.g. SCTE-35) in the input signal.

Another conversion that the encoder shall make is for timestamp wraparound to map to a continuous timeline. For example, an MPEG-2 timestamp would need to be converted to a continuous timeline even when the timestamp wraps due to overflow. This conversion is conducted by the encoder.

In equation (1), *track\_timescale* is the timescale used by the media track, where timescale is as defined in ISO/IEC 14496-12 in the mdhd (MediaHeader box). To avoid rounding errors, the timescale of the input should either be the same as the *track\_timescale* or an integer fraction of the *track\_timescale* .

To calculate segment boundaries equation (2) is used:

*(2)*

The segment boundary of the segment is the earliest presentation time of that segment. To calculate the number of the next segment (i.e., the next ), equation (3).

*(3)*

In equation (3) *now* is the time in seconds relative to epoch calculated using a real-time system routine. Next K is the value of K to compute the next segment boundary.

An encoder joining a synchronized encoder session, computes Next\_K and may also compute the previous K (Next\_K - 1) and start buffering and encoding frames based on received input. Once a complete media segment is encoded with the segment boundary computed as based on (2) it can be transmitted using an HTTP POST or PUT request or another method of transferring the media segment to a receiver. Joining or leaving encoders shall use the same segment naming, thus identical Representation@id and SegmentTemplate@media and SegmentTemplate@initialization shall be used in the I-MPD generated by different encoders.

The @availabilityStartTime attribute in the I-MPD shall be set to 1970-01-01T00:00:00Z and the Period@start should be set to PT0S. In case Period@start is not PT0s or a semantically equivalent value, the corresponding SegmentTemplate@presentationTimeOffset attributes shall be used to compensate the media presentation times relative to the period start time. The media presentation times are always relative to 00:00:00 1970-01-01 UTC excluding leap seconds.

Any receiver may identify identical segments based on the baseMediaDecodeTime and the naming scheme identified from the AdaptationSet@SegmentTemplate and Representation@id. Any segment from the same Representation of track is interchangeable given that it has the same baseMediaDecodeTime. This way receivers can detect redundant/duplicate segments and/or additional representations or tracks.

# Asset storage and recording Format

Asset storage and recording shall use ISO Base Media File format ISO/IEC 14496-12 and a storage Media Presentation Description (S-MPD) using Media presentation description defined in ISO/IEC 23009--1. Further restriction to the track storage format will be defined in clause 8.1. Further constraints to the Storage Media Presentation Description (S-MPD) constraints are defined in clause 8.2.

## Track format for storage of live archives

A storage track file is defined as follows.

**Storage Track file:** A stored ISOBMFF track that follows the constraints of aCMAF Track file, without the restriction that the track starts at presentation time zero. In addition, it may also contain interleaved SegmentIndexBoxes after the first MovieFragmentBox that may occur at the start of a segment to enable random access on portions of the track or segment.

The CMAF Track format structure is used to store media content as defined in clause 7 of ISO/IEC 23000-19 [2]. A CMAF track includes a CMAF Header for setting metadata such as language, image width/height, media timescale, defining the codec usage in the sample entry and several other metadata relating to the track. Next it may include a SegmentIndexBox (sidx box) for byte-range based access in the track.

In addition, this clause defines several urn’s and identifiers to code properties in each track to allow recreating the manifest if needed. Both identifiers corresponding to DASH and CMAF constructs are introduced and linked. Table 1 links the DASH identifiers and the CMAF identifier constructs . Table 2 defines the schemeURI for each identifier when used according to this proposal when signalling in a `kind` box. The value attribute in kind shall be set to the identifier and UUID/ID respectively.



1. Relationship between the manifest and its segments.



1. Orphaned content stream or asset with information to do reverse lookup.
2. Example CMAF identifiers and mapping to DASH identifiers.

|  |  |
| --- | --- |
| **DASH construct** | **CMAF construct** |
| MPD ID | CMAF Presentation ID |
| Period ID | CMAF Presentation ID |
| AdaptationSet  Group ID | Aligned Switching Set ID |
| AdaptationSet ID | SwitchingSet ID |
| Representation ID | CMAF track id (not to confuse with track\_id) |

1. Link between constructs in DASH and HLS

|  |  |
| --- | --- |
| Content ID (optional) | urn:mpeg:asset-storage-format:cmaf-content-id |
| CMAF Presentation ID (MPD) | urn:mpeg:asset-storage-format:cmaf-mpd-id |
| CMAF PresentationID (Period) | urn:mpeg:asset-storage-format:cmaf-period-id |
| Aligned Switching Set ID | urn:mpeg:asset-storage-format:cmaf-aligned-switching-id |
| SwitchingSet ID | urn:mpeg:asset-storage-format:cmaf-switching-id |
| CMAF track id (not to confuse with track\_id) | urn:mpeg:asset-storage-format:cmaf-track-id |

Track identifiers may be optionally present in tracks. If present, they shall be signalled using a schemeURI in the kind box in udta as and the value as a UTF-8 character encoded and null terminated UUID or id value as string. The defined schemeURI from Table 2 are supported.

## Storage Media Presentation Description (S-MPD)

### Overview

A manifest presentation description (MPD) can be used to reference stored track files. For video on demand content is it easy to use to on demand profile defined in ISO/IEC. For live content the isoff live profile can be used.

The case is for recording or archiving live 24x7 content is more challenging as receiver may be continuously updating and writing/appending segments. In this case the S-MPD references track file segments of intermediate durations (1 or more concatenated segments). For this case, we introduce constraints to the manifests and naming schemes to the S-MPD

Table 3 summarizes the profiles and constraints for the different storage, recording and archiving options.

1. Options and profiles for Storage MPD (S-MPD)

|  |  |  |
| --- | --- | --- |
|  | @profiles | constraints |
| 24x7 live | urn:mpeg:dash:profile:isoff-live:2011 | As in clause 8.2, may use CMAF profile for DASH |
| simple live | urn:mpeg:dash:profile:isoff-live:2011 | may include SegmentTimeline, may use CMAF profile for DASH |
| VoD | urn:mpeg:dash:profile:isoff-on-demand:2011 | may use CMAF profile for DASH |

### S-MPD for 24x7 live archiving and recording

S-MPD may be used to reference tracks used in an archived or stored/recorded media presentation.

Fixed duration (with small deviations allowed) intermediate duration (with one or more concatenated segments) track files are stored as long duration media segments in the S-MPD. These media segments contain all data from the archive track file except the initialization segment or CMAF Header. The may also include the SegmentIndexBox.

The first long duration archive track segment may also be of a shorter duration as compared to the target fixed duration.

NOTE: In case the initialization segment changes a new Period would need to be introduced to describe it, this is not described in this contribution but may be discussed more when the proposal is accepted.

The receiver/archiver is configured to generate and store track files using a fixed duration M. The S-MPD uses the iso live profile, and SegmentTemplate elements with @duration set to the fixed duration M accounted for the timescale of the tracks. Each track file is stored as a media segment in the Media Presentation Description.

Table 4 specifies the constraints on the manifest. The @availabilityStartTime is set to the time of the earliest presentation time of the first archive track file (if this S-MPD is written after that has been received) or the time the MPD was written given that this is before the earliest presentation time of the first archive track segment. The @timeShiftBufferDepth should be absent unless the solution also handles removing archive track segments. AdaptationSets are created corresponding to the I-MPD using similar configuration of the adaptation sets, but as SegmentTemplate@duration and $number$ based indexing instead of optional $Time$ based indexing.

The following example sequence illustrates initialising the S-MPD from an I-MPD:

1. Steps for initializating the S-MPD for 24x7 live content

|  |
| --- |
| 1. The MPD follows the iso live profile (urn:mpeg:dash:profile:isoff-live:2011) 2. MPD@type shall be ‘dynamic’ 3. @availabilityStartTime is set to the earliest presentation time of the first archive track segment (or if it is not known to the time the MPD was written) 4. Period@start is set to “PT0S” or a semantically equivalent value 5. MPD@timeShiftBufferDepth and Period@timeShiftBufferDepth shall not be present 6. Each AdaptationSet shall contain a SegmentTemplate element, the SegmentTemplate element constrained as follows:    1. Each SegmentTemplate@presentationTimeOffset is present to make the media presentation timing relative to the epoch anchor, i.e. 1970-01-01T00:00:00Z, the time is MPD@availabilityStartTime - 1970-01-01T00:00:00Z adjusted for the SegmentTemplate@timescale    2. The SegmentTemplate@duration is set to the fixed duration M adjusted for SegmentTemplate@timescale    3. SegmentTemplate@media is the same for each SegmentTemplate and contains the substring “$RepresentationID$” and “$Number$”   SegmentTemplate@initialization is the same for each SegmentTemplate and contains the substring “$RepresentationID$” |

### Generating an S-MPD from an I-MPD

Following the architecture in Figure 3, a receiver may create and write a S-MPD based on an I MPD following the steps in Table 4 and Table 5. A receiver may write the received segments as track files references in the S-MPD using the steps detailed in Table6.

When the live stream stops, and no more segments are expected, the MPD@type attribute may be changed from ‘dynamic’ to ‘static’ and the SegmentTemplate@endNumber, Period@duration attributes may be set in the manifest.

1. Steps for writing segments to the S-MPD for 24x7 live content

|  |
| --- |
| 1. Initialise the S-MPD based on constraints from Table 3, read the configuration values for duration M from a memory 2. Set the MPD@availabilityStartTime and SegmentTemplate@presentationTimeOffset fields to the earliest presentation time of a received archive track file (or if this is not known to the time the MPD was written) 3. Read the I-MPD and create the corresponding AdaptationSets in the S-MPD,    1. The SegmentTimeline is not used but instead SegmentTemplate with @duration is used, @duration being set to M accounted for the SegmentTemplate@timescale    2. Representation@id matches between the I-MPD and S-MPD    3. The @startNumber is set to floor(MPD@availabilityStartTime (in seconds relative to 1970-01-01T00:00:00Z)/M)    4. SegmentTemplate@media from I-MPD but replacing $Number$ instead of $Time$ 4. Only write the S-MPD to storage if it was not already present in the storage |

1. Steps for writing the archive track files for 24x7 live content or encoder synced content (assuming the S-MPD is already written)

|  |
| --- |
| 1. Compute the next archive track boundary next\_L, the L + 1th archive track file since the established anchor 1970-01-01T00:00:00Z by next\_L = ceil(now/M), where now is the seconds based timestamp relative to Epoch or the established anchor. 2. Compute the next archive track file boundary B as next\_L \* D \* output track timescale 3. Identify from the input signal the frame time information, in this case the input is a CMAF media segment, it is the CMAF media segments earliest presentation time, potentially adjust it with a synchronisation timestamp 4. Append the incoming media segment in memory to the continue creating the archive track file 5. In case the media segment earliest presentation time plus the media segment duration equals or is greater than B goto 6 otherwise to 3 6. Upload the in memory archive track file using the SegmentTemplate URL generation substituting $RepresentationID$ for the Representation@id derived from the input track and replacing $Number$ by next\_L – 1 7. GoTo 1 |

# Example Media Presentation Description

(Informative)

## Example I-MPD (informative)

An example I-MPD is shown below.

|  |
| --- |
| <MPD  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance”xmlns="urn:mpeg:dash:schema:mpd:2011"  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011"  type="dynamic“  availabilityStartTime="1970-01-01T00:00:00Z“  publishTime="2022-02-21T13:22:39.799801Z“  minimumUpdatePeriod="PT2S”  timeShiftBufferDepth="PT5M"  maxSegmentDuration="PT2S“  minBufferTime="PT10S“  profiles="urn:mpeg:dash:profile:isoff-live:2011">  <Period id="1“ start="PT0S">  <AdaptationSet  id="1“  group="1“  contentType="audio“  lang="en“  minBandwidth="64000“  maxBandwidth="128000“  segmentAlignment="true“  audioSamplingRate="48000"  mimeType="audio/mp4“  codecs="mp4a.40.2“startWithSAP="1">  <AudioChannelConfiguration  schemeIdUri="urn:mpeg:dash:23003:3:audio\_channel\_configuration:2011“  value="1" />  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" />  <SegmentTemplate  timescale="48000“  initialization="live-$RepresentationID$.dash“  media="live-$RepresentationID$-$Time$.dash">  <!-- 2022-02-21T13:17:36Z / 1645449456 - 2022-02-21T13:22:37.440000Z -->  <SegmentTimeline>  </SegmentTimeline> </SegmentTemplate>  <Representation id="audio\_eng=64000“ bandwidth="64000">  </Representation>  <Representation id="audio\_eng=128000“ bandwidth="128000"> </Representation>  </AdaptationSet>  <AdaptationSet  id="2“ group="2“  contentType="video“ par="16:9“  minBandwidth="500000“  maxBandwidth="1000000“  segmentAlignment="true“  width="1280“ height="720"  sar="1:1“  frameRate="25“  mimeType=”video/mp4“  codecs="avc1.42C01F“  startWithSAP="1">  <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" />  <SegmentTemplate  timescale="600“  initialization="live-$RepresentationID$.dash“  media="live-$RepresentationID$-$Time$.dash">  <!-- 2022-02-21T13:17:36Z / 1645449456 - 2022-02-21T13:22:37.440000Z -->  <SegmentTimeline>  </SegmentTimeline>  </SegmentTemplate>  <Representation id="video=500000“ bandwidth="500000“ scanType="progressive">  </Representation>  <Representation id="video=1000000“ bandwidth="1000000“ scanType="progressive">  </Representation>  </AdaptationSet>  </Period>  <UTCTiming  schemeIdUri="urn:mpeg:dash:utc:http-iso:2014"  value="https://time.akamai.com/?iso" />  </MPD> |

## Example S-MPD (informative)

TBD

# Recommended Configurations

(Informative)

## Recommended segment durations for encoder synchronization (informative)

In some cases, it is difficult to achieve segment boundary alignment. This is because audio samples in ISOBMFF contain multiple media samples (e.g. 1024 samples in an MP4 audio ISOBMFF sample) and therefore the granularity of audio and video segments may not be the same. In practice choosing values based on common frame-rates and sample rates can mitigate this problem. The table below illustrated some common options to avoid misalignment.

This proposal does not address the non-integer framerate case, but practice has shown that for M=L x D still fixed duration M can be achieved enabling a pattern in the media segment durations. This could be exploited to achieve the same mechanism to achieve consistency but based on super- segment duration M.

**Table 1 Example rates, timescale and segment duration at 25 fps and 30 fps**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Media type | Frame rate / sample rate | Timescale | Frames per segment | Segment duration D |
| Video | 25 fps | 25 | L\*24 | L\*0,96 |
| Audio | 48 Khz | 48000 | L\*45 | L\*0,96 |
| Timed text | - | 1000 | 1 or more | L\*0,96 |
| Metadata | - | 1000 | 1 or more | L\*0,96 |
| Video | 30 fps | 25 | L\*48 | L\*1,6 |
| Audio | 48 Khz | 48000 | L\*75 | L\*1,6 |
| Timed text | - | 1000 | 1 or more | L\*1,6 |
| Metadata | - | 1000 | 1 or more | L\*1,6 |

# Example Applications

(Informative)

## Redundant and high availability distribution:

in this application encoder synchronization method is used to achieve setups with multiple encoders and receiver mainly to be fault tolerant, if one encoder or receiver fails switchover is possible

## Distributed encoding:

in this application different encoders generate different bit-rates of different codecs of media types. In this case for example high bit-rate Ultra High Definition resolutions are encoded on a different distributed entity.

## A/B watermarking:

In this application, different sources are marked with an A or a B watermark resulting in watermarked segments. Segment sequences can then be combined to generate user specific watermarks to avoid content leaking. In such case the processed and outputs tracks of encoders generating A and B watermarks can be generated using the encoder synchronization method disclosed in this document.

Bibliography

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

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

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

[4] DASH-IF — DASH-IF Live Media Ingest Protocol Technical Specification, 14 March 2022: https://dashif-documents.azurewebsites.net/Ingest/master/DASH-IF-Ingest.html