Creating WAVE Test Content for a Media Profile

# Introduction

This document contains

# Content options to be tested

The first step in developing test content is to list the content options that need testing for the specific media profile. Ideally these would be documented as part of the media profile specification but they may be hidden in normative references. There may be options within the content bitstream and / or ISOBMFF options all of which should be considered for testing.

If there is a pervasive testing or certification program for the media profile which covers some or all of the content options then there is no need to duplicate such a program however it should be documented which of the content options are covered by that program.

Content options will include both options in the bitstream format and options for how the bitstream can be packaged in CMAF.

For video, a decision needs to be made about the extent to which testing of different frame rates and frame rate families is required. Three options exist.

* Frame rates are considered as a content option and are tested in combination with the other content options.
* The set of content option tests are duplicated for each frame rate family. (Testing each combination of content options and frame rate is not considered further).
* The set of content option tests are only run for one frame rate family. The other frame rate families are only tested with a default or baseline set of content options in a CMAF Switching Set.

For audio, a decision needs to be made about the extent to which testing of different sample rates is required.

* Audio sample rates are considered as a content option and are tested in combination with the other content options.
* The set of content options are only run for one audio sample rate. The other sample rates are only tested with a default or baseline set of content options in a CMAF Switching Set.

**Results**: A list of content options and their possible values.

**Example**: The WAVE content options to be tested for AVC are as follows;

* with and without picture timing SEI message.
* with and without VUI timing information.
* Sample entry
  + avc1 sample entry type (parameter sets within the CMAF Header)
  + avc3 sample entry type (in-band parameter sets) without parameter sets within the CMAF header
  + avc3 sample entry type (in-band parameter sets) with parameter sets within the CMAF header
* CMAF Fragment durations – 2s or 5s
* Initialization Constraints either;
  + Regular Switching Set, do not apply CMAF clause 7.3.4.2 and 9.2.11.4
  + Single initialization constraints, see CMAF clause 7.3.4.2 and 9.2.11.4
* Fragments containing one or multiple moof/mdat pairs

# Create a list of the test streams & their properties

## Introduction

The WAVE tests need test streams that fall into a number of groups.

* Test streams for content options (see above)
* Test streams for switching within a CMAF Switching Set
* Encrypted streams
* Splice conditioned streams

These are described in more detail in each of the sections below.

## Test streams for content options

People running tests prefer to minimize the time taken to run them. Hence it is preferred to have as few streams as possible and combine content options into a small number of streams. If an implementation fails one of these combined streams then individual debug streams should be defined for debugging purposes. One approach is to define a baseline stream and then a number of streams that differ from this.

Here is an abstract example, others are possible. Assume the following 3 content options;

* A – Boolean - true / false
* B – Tri-state - 0/1/2
* C – Optional - Present / absent

This would result in 3 streams for testing (a baseline stream, a stream where all options are the opposite of the baseline stream and a third stream). There are also 4 debug streams that differ from the baseline by only one content option.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Content  Option | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|  | Baseline stream | Opposite of baseline | Anything else | Debug option A | Debug option B -1 | Debug option B-2 | Debug option C |
| A | True | False | *Either* | False | True | True | True |
| B | 0 | 1 | 2 | 0 | 1 | 2 | 0 |
| C | Absent | Present | *Either* | Absent | Absent | Absent | Present |

This particular example could be optimised by combining streams 2 and 5 but they have been left separate for clarity.

Each of these streams (except for the baseline) would be used in a test that plays the stream once from the beginning to the end (the “Sequential Track Playback” test).

At this point it is also recommended to check that the encoding and packaging tools support all the options and all the combinations of options. Some options may not be supported at all. Some combinations may not be supported or indeed may be prohibited by the CMAF profile or the bitstream format.

**Results** : A list of streams with the values of the content options defined for each.

**Example**: The following table shows 11 streams for AVC, 1 baseline stream, 2 streams for testing combinations of content options and 8 debug streams for individual content options. In this case, the frame rates / frame rate families are not treated as content options.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Description** | Baseline | All AVC content options different from baseline | Same as baseline except for AVC content options that have 3 values different from 1 and 2. | Same as baseline but with only picture timing SEI message different. | Same as baseline but with only VUI timing different. | Same as baseline but with different sample entry type. | Same as baseline but with 3rd possible sample entry type. | Same as baseline but with only CMAF fragment duration different. | Same as baseline but with only initialization constraints different | Same as baseline but with only moof/mdat pair different | Same as baseline but with third option for moof/mdat pairs. |
| **with and without picture timing SEI message.** | With | Without | With | Without | With | With | With | With | With | With | With |
| **with and without VUI timing information.** | Without | With | Without | Without | With | Without | Without | Without | Without | Without | Without |
| **Sample entry, see CMAF clause 9.4.1.2** | avc1 sample entry type (parameter sets within the CMAF Header) | avc3 sample entry type (in-band parameter sets) without parameter sets within the CMAF header | avc3 sample entry type (in-band parameter sets) with parameter sets within the CMAF header | avc1 sample entry type (parameter sets within the CMAF Header) | avc1 sample entry type (parameter sets within the CMAF Header) | avc3 sample entry type (in-band parameter sets) without parameter sets within the CMAF header | avc3 sample entry type (in-band parameter sets) with parameter sets within the CMAF header | avc1 sample entry type (parameter sets within the CMAF Header) | avc1 sample entry type (parameter sets within the CMAF Header) | avc1 sample entry type (parameter sets within the CMAF Header) | avc1 sample entry type (parameter sets within the CMAF Header) |
| **CMAF Fragment durations (s)** | 2 | 5 | 2 | 2 | 2 | 2 | 2 | 5 | 2 | 2 | 2 |
| **Initialization Constraints** | Regular Switching Set, do not apply CMAF clause 7.3.4.2 and 9.2.11.4 | Single initialization constraints, see CMAF clause 7.3.4.2 and 9.2.11.4 | Regular Switching Set, do not apply CMAF clause 7.3.4.2 and 9.2.11.4 | Regular Switching Set, do not apply CMAF clause 7.3.4.2 and 9.2.11.4 | Regular Switching Set, do not apply CMAF clause 7.3.4.2 and 9.2.11.4 | Regular Switching Set, do not apply CMAF clause 7.3.4.2 and 9.2.11.4 | Regular Switching Set, do not apply CMAF clause 7.3.4.2 and 9.2.11.4 | Regular Switching Set, do not apply CMAF clause 7.3.4.2 and 9.2.11.4 | Single initialization constraints, see CMAF clause 7.3.4.2 and 9.2.11.4 | Regular Switching Set, do not apply CMAF clause 7.3.4.2 and 9.2.11.4 | Regular Switching Set, do not apply CMAF clause 7.3.4.2 and 9.2.11.4 |
| **Fragments containing one or multiple moof/mdat pairs** | Fragment is 1 chunk | Fragment contains multiple chunks (p-frame to p-frame with b-frames) | Each sample constitutes a chunk (p-frame only) | Fragment is 1 chunk | Fragment is 1 chunk | Fragment is 1 chunk | Fragment is 1 chunk | Fragment is 1 chunk | Fragment is 1 chunk | Fragment contains multiple chunks (p-frame to p-frame with b-frames) | Each sample constitutes a chunk (p-frame only) |

## Test streams for switching within a CMAF Switching Set

A number of the WAVE tests require a CMAF switching set. For video, there may be 3 of these, one for each frame rate family. For audio, there may be only one. Such switching sets will likely contain all frame rates / sample rates that might be used – far more of each than would normally be found in a frame rate ladder for real content.

## Encrypted streams

WAVE includes tests for decryption which therefore need encrypted streams. Ideally these would use common encryption and be decryptable with any of the common DRM systems. In practice, we are likely starting with Clear Key as defined in EME.

## Splice conditioned streams

WAVE tests include splice conditioned streams.

## Results

The result of this step needs to be a list of all the CMAF Tracks that need to be created with a complete definition of the content options and frame rate / sample rate for each.

# Cross-checking

## Checking mezzanine streams exist for all test streams

Once there is a list of the test streams (CMAF tracks), it needs to be checked that the corresponding mezzanine stream exists for each test stream. One thing to note Is that, for CMAF video Switching Sets, each Track needs to be generated from the corresponding mezzanine stream of the same resolution and not by down-sampling the highest quality mezzanine stream.

**Results**: A table listing, for each test stream (CMAF track) which mezzanine stream it would be generated from

## Checking tool and parameters to be used for generating each test stream

Once there is a list of the test streams (CMAF tracks), it needs to be checked that tools exist to generate each test stream from the corresponding mezzanine stream. It needs to be checked that the parameters / arguments needed for each test stream are supported in the combinations needed and don’t have any unexpected side-effects. This may require iteration potentially even needing revising the proposed combinations of content options and test streams.

A goal of WAVE is enabling organizations to repeat the process of generating test content. As a result, the following preferences apply for tools for generating test streams in descending priority order;

1. Open source tools that are freely available to anyone
2. Commercial tools that are available to anyone who can afford them
3. Future versions of commercial tools that will, in time, be available to anyone who can afford them
4. Tools inside an organization that are unlikely to ever be made available externally

It is understood that mixtures of tools may need to be used if no single tool supports generating all the identified test streams. If the only tools that support generating all the identified test streams are internal to an organization then it is preferred to use widely available tools for as many streams as possible and limit the use of the internal tools to the absolute minimum.

**Results**: A table listing, for each test stream (CMAF track) which mezzanine stream it would be generated from, which tool would be used to generate the test stream and which options for that tool need to be used to generate the required result.

# Generating test streams

The generation of test streams from mezzanine content should be scripted so that it is reproducible with minimum manual intervention when issues are found.

Where tools used are open source or commercially available, the scripts need to be contributed to the CTA WAVE github repository under conditions that allow them to be re-used by others in the future.

**Result**: A set of test streams (CMAF tracks) – media segments, initialization segment(s) and a DASH MPD providing metadata about the stream (see xxx). The script to generate the test streams.

# Validating test streams

The test streams need to be validated for two criteria.

* Compliance with CMAF, the CMAF profile and the WAVE content profile being tested
* That each stream indeed has the correct content options, frame rate / sample rate (etc)

The same logic applies as with generation of test streams.

* The validation of test streams from mezzanine content should be scripted so that it is reproducible with minimum manual intervention when issues are found.
* The following preferences apply for tools for validating test streams in descending priority order;

1. Open source tools that are freely available to anyone
2. Commercial tools that are available to anyone who can afford them
3. Future versions of commercial tools that will, in time, be available to anyone who can afford them
4. Tools inside an organization that are unlikely to ever be made available externally

* It is understood that mixtures of tools may need to be used if no single tool supports validating all the identified test streams.
* If the only tools that support validating all the identified test streams are internal to an organization then it is preferred to use widely available tools as far as possible and limit the use of the internal tools to the absolute minimum.

**Result**: A set of test streams (CMAF tracks) – media segments, initialization segment(s) and a DASH MPD providing metadata about the stream (see xxx). The results of validating the test streams should be archived in the CMAF github for later reference in case problems are found.

# Defining mapping between test streams and HTML/JavaScript test code

The mapping between test streams and HTML/JavaScript test code is essentially a sparse matrix. This needs to be documented in a .CSV file. Each line in this file corresponds to a test that will be generated.

A summary of this is below.

|  |  |  |
| --- | --- | --- |
| # | Title | Special stream requirements |
| 8.2 | Sequential track playback | None – used with all streams for content options where a fragment is 1 chunk |
| 8.3 | Random Access to Fragment | None – used with baseline stream from content options |
| 8.4 | Random Access to Time | None – used with baseline stream from content options |
| 8.5 | Switching Set Playback | None – used with Switching Set |
| 8.6 | Regular Playback of Chunked Content | None – used with all streams for content options where a Fragment contains multiple chunks |
| 8.7 | Regular Playback of Chunked Content, non-aligned append | None – used with one stream where fragment contains multiple chunks |
| 8.8 | Playback over WAVE Baseline Splice Constraints | Uses specially constructed content with video and audio segment durations |
| 8.9 | Out-Of-Order Loading | None – used with baseline stream from content options |
| 8.10 | Overlapping Fragments | None – used with Switching Set |
| 8.11 | Full Screen Playback of Switching Sets | None – used with baseline stream from content options |
| 8.12 | Playback of Encrypted Content |  |
| 8.13 | Restricted Splicing of Encrypted Content | ???? |
| 8.14 | Sequential Playback of Encrypted and Non-Encrypted Baseline Content | None – use |
| 9.2 | Regular Playback of a CMAF Presentation | None – use baseline streams from the appropriate video and audio media profiles. |
| 9.3 | Random Access of a WAVE Presentation | None – use baseline streams from the appropriate video and audio media profiles. |
| 9.4 | Splicing of WAVE Program with Baseline Constraints | ???? |

To summarise;

* Some HTML+JavaScript will be run with only one test stream for each WAVE content profile
* Some HTML+JavaScript will be run with more than one test stream.
* Some test streams will be run with only one set of HTML+JavaScript
* Some tests streams will be run with more than one set of HTML+JavaScript

# Generate and run the tests

It is encouraged that media profile proponents arrange for the tests to be generated and run. This could be done internally or by a volunteer or sub-contractor.