Streaming Media Test Suite – Devices Explained

Consumer Technology Association

Summary

The WAVE Project has launched its initial “WAVE Streaming Media Test Suite – Devices” to enable automated testing of web-based media playback on various devices like Smart TVs, media sticks, smart phones, laptops and set-top boxes.

This suite aims to streamline testing processes by allowing device implementers to prove compatibility with industry standards, reducing repetitive tests across the industry.

It features unit tests for media using CMAF formats and web streaming technologies like MSE and EME. The suite also includes preliminary tests for popular video and audio codecs, with plans to expand codec testing in 2024.

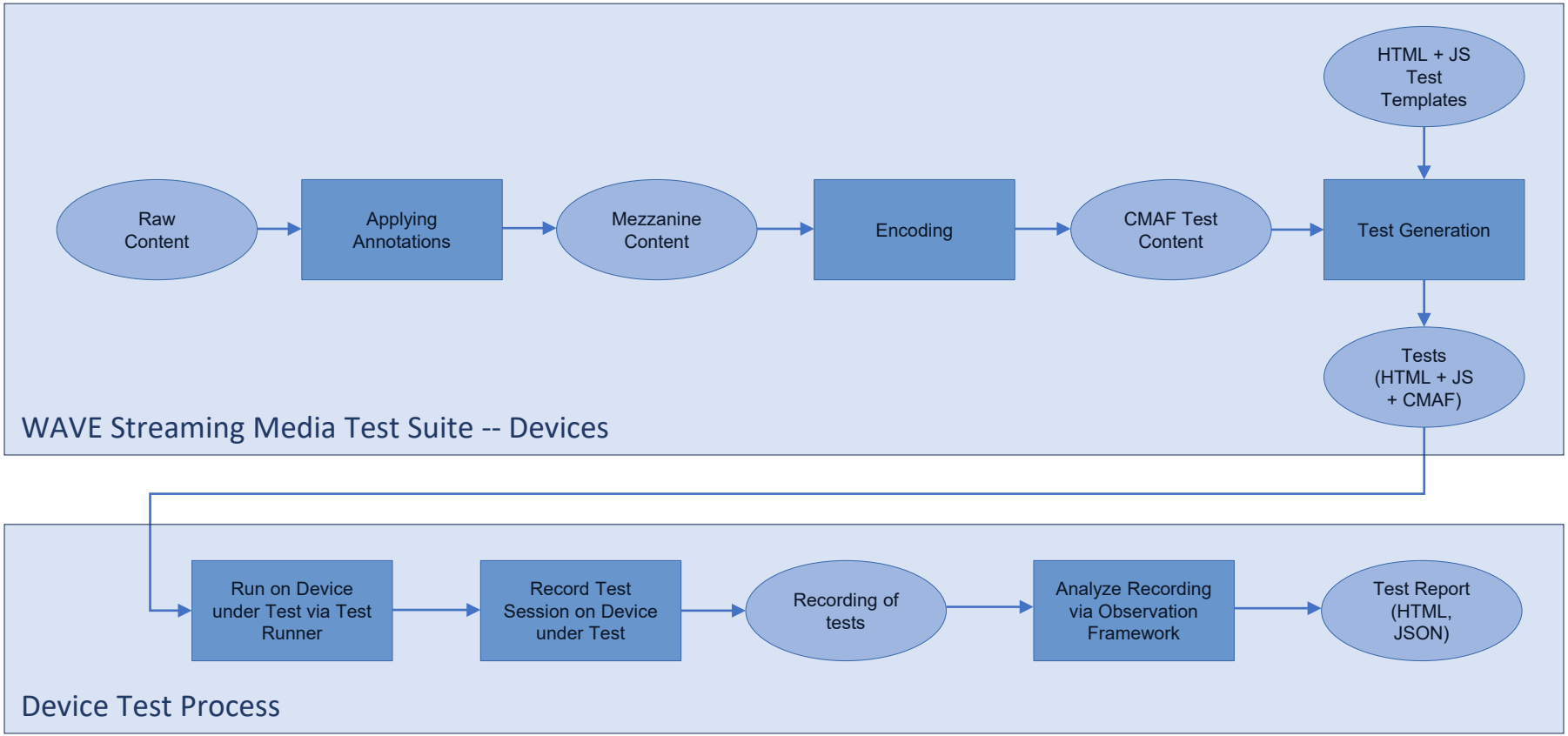
What?

- Tests for AVC video playback using MSE and EME (21 validated 'primary' tests)
  - Each test is the combination of an HTML+JavaScript template with CMAF test content
  - Duplicate tests for 25 / 50 Hz, 30 / 60 Hz and fractional frame rates
  - Primarily AVC profile/level 4.0 ('cfhd') but includes level 4.2 ('chdf') for 1080p50/60
  - Also some 'beta' tests that are still in validation
    - Total = 46 tests (25/50) and 39 tests (for each of 30/60 and fractional frame rates)
- Test Runner
  - Based on Web Platform Test (WPT) test runner
    - Also used by CTA WAVE 'Web Media API Snapshot' (WMAS) tests
    - Packaged as a docker image
    - Documentation extensively reviewed and improved in last 12 months
- Observation Framework
  - Analyzes recordings made from running tests on devices for compatibility with MSE spec and CTA-5003 Device Playback Capabilities Specification

Why?

- Improve inter-operability between apps and devices for media presentation using web APIs
  - Strongest contributor interest is Smart TVs
- Reduce combinatorial explosion of testing between apps and devices
  - Enable device implementers to test once and present results to content providers / platforms
- Improve viability of web as a format for media apps

System Components



How It Works: Annotated Mezzanine Content

- Annotated video content in many different resolutions
  - Based on big buck bunny for 60Hz and fractional frame rates & an EBU sequence from Croatia for 50Hz.
  - Green frame on the start and red frame on the end.
- Annotations burnt into the video
  - Rotating QR code for observation framework.
  - Human-readable text for debugging.
    - Same information as in the QR code.
  - Bit pattern for TV manufacturer in-house use.
  - Flashing square with beeps for simple A/V sync testing.
  - Red triangles to check all content is visible.
- See next slide for example.
- Audio content based on pseudo-random noise.
  - Observation framework can reconstruct a timeline from this.


I1
00:00:05.360;0000134;25

CMAF Test Content

- Encoded & validated mezzanine content.
  - CMAF media segments (or is it fragments?).
  - Metadata describing media segments provided in the form of a DASH MPD.
- 4 groups of streams encoded for each technology
  - Testing combinations of content options.
  - Testing content options individually for debugging.
  - Resolutions for testing CMAF switching sets.
  - Special content for testing splicing, encryption, ...
- For each codec, streams to encode are defined in a sparse matrix.
  - Examples for AVC video on next slides (based on what is just released).
- Validation in two phases
  - DASH-IF aka ‘JCCP’ validator checks the content is valid CMAF.
  - Script reads in the sparse matrix & checks each stream has the correct content options, resolution, cmaf profile and other properties.

HTML and JavaScript Templates

- Media codec / technology independent
  - Exercise MSE (and EME) playback.
  - Algorithms documented in clauses 8 & 9 of WAVE Device Playback Specification.
  - Common library that reads and plays CMAF content based on metadata in DASH MPD.
- Mixture of templates for either video or audio or for both video+audio.
- Examples
  - Sequential Track Playback
    - Play a CMAF video or audio track from beginning to end.
  - Random Access to Fragment, Random Access to Time
    - Play a CMAF video or audio track from somewhere in the middle.
  - Switching Set Playback
    - Play a CMAF switching set, switching between tracks at various points.
  - Playback over WAVE Baseline Splice Constraints
    - Playback switching from one CMAF track to a second and back again.
  - Buffer Underrun and Recovery
    - Playback terminating.

Test Generation

- Sparse matrix that defines test content to encode also defines combinations of HTML+JS templates to use with test content.
- Script copies templates and inserts URLs for MPDs.

Tests (1); Validated Primary Tests

buffer-underrun-and-recovery__t2	random-access-to-fragment__t1
fullscreen-playback-of-switching-sets__ss1-1	random-access-to-time__t1
fullscreen-playback-of-switching-sets__ss1-2	regular-playback-of-chunked-content__chunked
low-latency-initialization__t2	regular-playback-of-chunked-content-non-aligned-append__chunked
low-latency-short-buffer-playback__t2	restricted-splicing-of-encrypted-content-https__splice_main-cenc_splice_ad-cenc
mse-appendwindow__t1	sequential-playback-of-encrypted-and-non-encrypted-baseline-content-https__splice_main-cenc_splice_ad
out-of-order-loading__t1	sequential-track-playback__t1
overlapping-fragments__ss1	sequential-track-playback__t3
playback-of-encrypted-content-https__t1-cenc	switching-set-playback__ss1-1
playback-over-wave-baseline-splice-constraints__splice_main_splice_ad	switching-set-playback__ss1-2
random-access-from-one-place-in-a-stream-to-a-different-place-in-the-same-stream__t2	

WAVE
WEB APPLICATION VIDEO ECOSYSTEM

These are the tests that would be referenced by a content / service provider when requiring devices to pass the CTA WAVE test suite.

Consumer Technology Association

Tests (2); Secondary Tests for Debugging Content Options

Test	Content option tested
sequential-track-playback__t10	Same as t1 but with only picture timing SEI message different.
sequential-track-playback__t11	Same as t1 but with only VUI timing different.
sequential-track-playback__t12	Same as t1 but with different sample entry type.
sequential-track-playback__t13	Same as t1 but with 3rd possible sample entry type.
sequential-track-playback__t14	Same as t1 but with only CMAF fragment duration different.
sequential-track-playback__t15	Same as t1 but with only initialization constraints different

Streams for primary tests
t1 = baseline set of content options
t2 = all content options different from t1
t3 = for content options with 3 possible values, content options different from t1 and t2
If there are problems playing t2 or t3, these secondary tests can be used to isolate which content option is causing the problem.

Tests (3); Secondary Tests for Debugging CMAF Switching Sets

Test	Resolution & Frame Rate
sequential-track-playback__t20 (chdf)	Same as t1 but 1080p50/60 instead of 1080p25/30.
sequential-track-playback__t21	Same as t1 but 1600x900
sequential-track-playback__t22	Same as t1 but 1280x720
sequential-track-playback__t23	Same as t22 but 720p50/60 instead of 720p/25/30.
sequential-track-playback__t24	Same as t1 but 1024x576, higher bitrate
Not generated	Same as t1 but 1024x576 lower bitrate
sequential-track-playback__t26	Same as t1 but 960x540
sequential-track-playback__t27	Same as t1 but 852x480
sequential-track-playback__t28	Same as t1 but 768x432.
sequential-track-playback__t29	Same as t1 but 720x404
sequential-track-playback__t30	Same as t1 but 704x396
sequential-track-playback__t31	Same as t1 but 640x360
sequential-track-playback__t32	Same as t1 but 512x288
sequential-track-playback__t33	Same as t1 but 480x270
sequential-track-playback__t34	Same as t33 but half frame rate

Tests (4); Beta tests for video and video+audio

- AVC without audio
  - truncated-playback-and-restart__splice_main_splice_ad
  - long-duration-playback
- AVC with audio
  - random-access-of-a-wave-presentation__t1_at1
  - regular-playback-of-a-cmaf-presentation__t1_at1

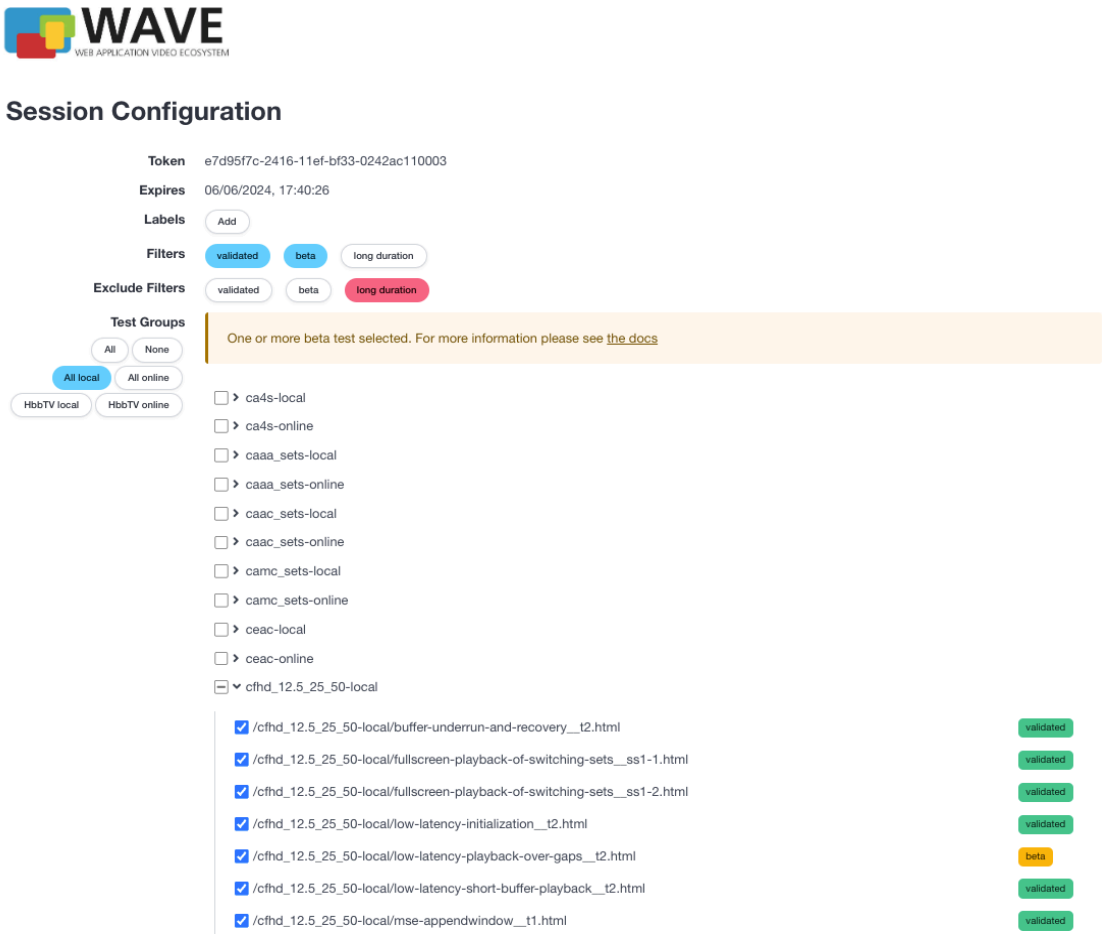
Tests (5); Secondary Tests for Debugging CMAF Switching Sets

CMAF media profile 4CC	Description	#
ca4s	AC-4, single stream	11
caaa	Adaptive AAC core	3
caac	AAC core	15
camc	AAC multichannel	2
ceac	Enhanced AC-3, including AC-3	11

Test Runner

- Extended version of test runner used for ‘Web Platform Tests’
  - Device Under Test loads HTML page from the test runner.
  - Test operator connects to test runner with desktop or mobile browser.
- Test operator configures a session by selecting one or more groups of tests.
  - Based on CMAF media profile and/or use by HbbTV.
  - Filtered by validated or beta (see later) or long duration.
- Device runs the session controlled by the test runner.
  - Results and debug output saved to the test runner.
- See next slide for screen shot.

Test Runner UI for Session Configuration



Consumer Technology Association

Installer

- Creates two docker images, one with test runner and one with observation framework.
- Can be run on Windows with Docker Desktop
  - Commercial product, not free to use in larger organizations.
- Can be run on Unix-like systems
  - Linux, Mac, Windows with WSL2 (without Docker Desktop)
- Installation needs someone who knows what they're doing with docker, IP routing, TLS server certificates, ...
  - Either someone from IT or a local power user.
  - Ensuring the test runner can talk to the Device Under Test and to the Observation Framework is not straightforward in some configurations.

Camera

- Recording & analysing 50/60Hz video playback needs camera recording at 120Hz
  - Validation done with top of the range mobile phone, e.g. Samsung S23+ or equivalent
    - Care needed as phones may have issues with adequately obtaining audio from TV.
- Avoiding reflections and glare when setting up camera is very important
  - Otherwise, Observation Framework will fail to extract QR codes
- Recording made to local memory card.
  - Downloaded to PC by moving the memory card across
- Experiments can limit video <= 30Hz and use cheaper cameras.
- Investigations in progress on streaming from camera to test runner.
  - Cameras likely to be more expensive professional equipment.
  - Mobile phone video streaming typically has limitations compared to recording on memory cards.

Observation Framework

- Processes recordings from the camera and reports results to the test runner.
  - Results are merged into the results from when the test was actually run.
- Two steps
  - Extraction of QR codes from the recording.
  - Processing of data from QR codes against test procedures in the WAVE Device Playback Specification (CTA-5003).
- OF can be run on the same device as runs the test runner or a different (faster) device.
  - OF can be installed either as a docker image or directly.
- OF generates log files and debug output for developers to analyze when a device is reported as failing.

Example of Failing Observations

cfhd_12.5_25_50-local: All Results

Test files: 1; **Total subtests**: 6

Test Files

1. /cfhd_12.5_25_50-local/low-latency-playback-over-gaps__t2.html

Test Show/Hide Messages	Xx01
/cfhd_12.5_25_50-local/low-latency-playback-over-gaps__t2.html	
Test workflow	PASS
[OF] Every video frame S[k,s] shall be rendered and the video frames shall be rendered in increasing presentation time order.	FAIL
Xx01: First frame found is 8, expected to start from 1. First frame number tolerance is 0. Last frame found is 251, expected to end at 750. Last frame number tolerance is 0. Mid frame number tolerance is 10. Total of missing frame count is 506. Last frame detected before gap 115 exceeded 'stall_tolerance_margin'=7.5 frames of expected frame 125.	
[OF] Video: The playback duration shall match the duration of the CMAF Track	FAIL
Xx01: Playback duration 10089.88ms does not match expected duration 9760.0ms +/- tolerance of 50ms. Detected duration is different by 329.88ms. Allowed tolerance is 50ms and duration frame tolerance is 0. Starting missing frame number is 7. Ending missing frame number is 499.	
[OF] Video: The presented sample shall match the one reported by the currentTime value within the tolerance.	PASS
Xx01: Total failure count is 0. Tolerances: +/-(1 frame(s) + 150ms.)	
video ended event fired	PASS
video remains in waiting state until skipping over the gap	PASS

From https://github.com/cta-wave/dpctf-tests/issues/152

Debugging a Failing Test

- Start your favorite video editor, go to the frame number from the log files and step through.
- Record what happens at what frame number (e.g.)
- Has the OF correctly failed the recording according to the test procedure?
- Is the test procedure wrong?
- Is the device wrong?

	A	B	C	D	E	F
1	Frame	Event				
2	73557	"Next test is about to start"				
3	74210	s: waiting; a: initialize				
4	74253	s: waiting; a: initialize; with app drawn QR code				
5	74271	s: ready; a: initiialize' ct: 0;				
6	74337	s: buffering; a: initialize; ct: 0;				
7	74648	s: playing; a: play; ct: 0;				
8	74692	s: playing; a: play; ct: 0; - QR code burnt into video detectable				
9	74696	1st frame of croatia detectable				
10	75315	last frame before seek 1st detectable – 00:00:05.240; 0000131;25				
11	76667	1st frame after seek may be detectable from QR code				
12	76669	1st frame after seek fully visible – 00:00:15.040;0000376;25				
13	78462	last frame of croatia first detectable – 00:00:29.960;0000749;25				
14	78463	1st frame of red first detectable – 00:00:30;0000750;25				

System Validation

- Tests run on Smart TVs at HbbTV interop event / plugfests.
  - June, October 2023 and January 2024.
- January 2024 used 8 different TVs from different manufacturers.
  - Prioritized three best TVs from those eight.
- Final regression tests based on TVs from two of those three manufacturers.

Example Errors Found by this Test Suite

- Dropped frames in the middle of content, either
  - Media timeline is preserved, one frame persists during the period of the dropped frames.
  - Media timeline is not preserved, period of the dropped frames is cut out of the media timeline, frames persist for their expected time.
- Dropped frames at the start and end of the content.
  - Either no video is shown, or broadcast video is shown in the case of a Smart TV.
- Media timeline stalls.
  - Media timeline just stops advancing for some period.
- Errors with JavaScript currentTime property relative to displayed video.
  - Either drift or currentTime continuing to increment after video playback has stopped.
  - Likely results in loss of sync between subtitles / captions and video.
- Seeking starts playback from the wrong location.
- Failure to restart after buffer underflow.

Give It a Try

- Instructions at https://github.com/cta-wave/dpctf-deploy/tree/v2.0.0
- Successfully used with four host configurations
  - Linux (primary supported host environment)
  - Windows using WSL2 (almost the same as Linux)
  - Mac (almost the same as Linux)
  - Windows using Docker Desktop commercial product
    - Note "Commercial use of Docker Desktop at a company of more than 250 employees OR more than $10 million in annual revenue requires a paid subscription (Pro, Team, or Business)."
- Three phases
  - Deployment (one-time action, to be performed with support from IT personnel or power users)
    - Note it is possible to get yourself tied in knots between virtualization systems unless done by someone who knows about Docker and IP routing between LAN, host and docker instances
    - Note doing this 'for real' requires a TLS server certificate which may be a challenge in a company network. It needs to be done by someone who knows what they're doing.
  - Test execution and recording
  - Observation

Acknowledgements

- Romain Bouqueau (Motion Spell) with support from Qualcomm and CTA
- Nicholas Frame, Jon Piesing (TP Vision)
- Louay Bassbouss, Fritz Heiden, Martin Lasak, Ilja Gavrylov (Fraunhofer FOKUS) with support from CTA
- Yan Jiang, Richard Cottingham, Peter Shorrock (Resillion) with support from Dolby, Fraunhofer IIS, CTA
- Thomas Stockhammer (Qualcomm)
- Zachary Cava (Walt Disney Co.)
- Mike Bergman, Bill Rose, Alexandra Blasgen (CTA)

Thank you