Vewd Core Content Development Guide

CONFIDENTIAL

Updated on 10 February, 2025

Table of Contents

1 Introduction

1.1 Scope

This document documents features and APIs available on devices powered by Vewd Core.

Vewd Core powers some TiVo OS, the new TV OS from Xperi/TiVo, - but also Retail TVs, Set-Top boxes and Car entertainment systems from other brands and vendors.

The document defines a complete OTT platform suitable for any OTT content globally, with rich features for media streaming, content protection, and security. The document is based on feedback from content owners and app developers worldwide and should answer most technical questions in the area of media streaming.

1.2 Versions

Vewd Core is updated yearly with features required for the next year of apps and content owners requirements. The version number is 4.x where x denotes the upcoming year. (e.g. 4.25.0 will be released the summer of 2024)

For each update of the Vewd Core, the version number of this specification is updated to match the product.

1.2.1. Backward compatibility

New versions of Vewd Core are normally backwards compatible with older versions, the Web is a very stable platform these days. The exception is when some features are deprecated by the major browsers (Chrome, Safari, Edge) due to no usage or security/performance concerns.

When we are planning to remove a feature, we will normally go through a DEPRECATION period first, before the feature is removed, but this may not always be true.

Features that are planned to be removed will then be marked as [DEPRECATED] in this document.

- After a deprecation period, [DEPRECATED] items will be removed in later versions of the document.
- The Specification revision history table at the end of the document will be updated accordingly.

1.3 Definitions

Chromium and **Google Chrome** - Chromium is the open-source project that forms the basis of the Google Chrome browser (and of Vewd Core).

Vewd Core - An embeddable browser and streaming engine with an extensible API, based on the Chromium open-source project, which implements a set of international and industry standards to download and render webpages, execute web apps, and stream video and audio content.

Vewd Products - A set of products and solutions developed by Vewd Software AS (now a part of TiVo Software, again a part of Xperi Inc.) for TV and Set-Top Box (STB) manufacturers to enable HTML5 rendering and adaptive streaming in their devices.

Vewd Application - a Vewd-compliant web app that is certified to run on Vewd powered devices. Also referred to as App or Application in this document.

Vewd Device - A TV or STB device running software based on the Vewd Core, and certified to meet the requirements defined in this document.

1.4 Platform dependent features

1.4.1 DRM

All Vewd Core powered devices support content protection with Microsoft PlayReady, and many also support Google Widevine. Vewd Core itself does not implement DRM but integrates to content protection on the platform.

1.4.2 Codecs and media formats

Support for video codecs and media formats are dependent on the underlying platform, as these are dependent on the chipset powering the device, and on external licenses.

Some codecs normally supported are:

Audio:

- Opus
- Dolby AC3/E-AC3

- Dolby ATMOS
- Dolby AC4
- Ogg Vorbis

Video:

- 265
- VP9

- AV1

Container:

- WebM (Only when VP9 is supported)

1.4.3 Keys on the remote control

The design of the remote control is up to each manufacturer, and may vary in size and functionality. Some keys will always be present (such as. 4-way navigation keys), while others will only be supported when there is such a key on the remote control.

1.4.4 Resolution

The maximum resolution of a device is defined by the manufacturer. All devices support either HD (720p) or Full HD (1080p).

2 Technical Requirements

2.1 HTML5 <video> and <audio>

Vewd Devices support HTML5 <video> and <audio> elements according to the HTML 5.1 specification [9].

2.1.1 Media element

Support for video and audio media elements

All devices support the following features for the handling of <video> and <audio> media elements:

- Initial state of the object's attributes
- Handling of error states during playback
- Audio and video elements able to play both types of sources
- Proper handling of events during the playback
- Proper handling of playbackRate and defaultPlaybackRate
- The resource selection algorithm is capable of handling multiple source elements
- The preload attribute of the media element will be respected

Support for video media elements

All devices support the following requirements for the handling of <video> media elements:

- Handling the poster attribute
- Proper aspect ratio MUST be preserved during resizing of the containing element of the video
- Video elements are able to play a stream containing only audio and text tracks
- Proper rendering of video sources
- Transition between two video streams, one playing, and one preloaded, must be seamless

Codec support

All devices shall respond truthfully to codec support inquiries:

- canPlayType response for specified video and audio codecs

2.1.2 Track element

Devices support rendering of subtitles and closed captions as specified in HTML 5.1, "The track element" (4.7.13). Supported track formats are specified in the Subtitles and Closed Captioning section.

Common features for all track elements

Support for the following features for the handling of track elements:

- Initial state values of media tracks and text tracks
- readyState attribute handling for out-of-band text tracks
- Proper handling of enabling and disabling audio tracks
- Switching of video and text tracks

3.1.2.2. Specific for text track elements

- Devices support the synchronization of the text track and the audio track.
- Text tracks are rendered properly on video media elements.

2.2 Media streaming

2.2.1 Transport protocols

- Devices support the retrieval of any media content either by HTTP or HTTPS using HTTP protocol v1.1 and Range requests.
- Devices support Transport Layer Security (TLS) version 1.2 with forward security.
- Devices handle TLS key of at least 2048 bits for RSA and 256 bits for EC.

2.2.2 Progressive download

Supported combinations:

Container	Audio codec	Video codec	DRM	DRM trigger	In-band subtitles

ISO BMFF	AAC-LC HE-AAC v1 HE-AAC v2 MP3 Dolby AC3 Dolby AC4 Dolby E-AC-3	H.264 H.265	None	None	Not supported
MPEG2-TS	AAC-LC HE-AAC v1 HE-AAC v2 MP3 Dolby AC3 Dolby AC4 Dolby E-AC-3	H.264	None	None	Not supported
WebM	Opus	VP9	None	None	Not supported
ADTS / AAC MP3	AAC-LC HE-AAC v1 HE-AAC v2 MP3	None	None	None	Not supported

Note 1: All rules and restrictions for the support of media formats and codecs applies as outlined in the Video and audio formats section.

2.2.3 Adaptive Bitrate streaming protocols

The following Adaptive Bitrate (ABR) streaming protocols are supported:

Streaming Type	**MIME-Types**	**Notes**
Apple HTTP Live Streaming (HLS)	application/vnd.apple.mpegurl application/x-mpegURL	VoD (append-mode window) and Event (sliding window)
MPEG-DASH	application/dash+xml	Main and Live profiles of MPEG-DASH
Microsoft Smooth Streaming (MSS)	application/vnd.ms-sstr+xml application/ vnd.ms-playready.initiator+xml	

Apple HTTP Live Streaming (HLS)

Devices support HTTP Live Streaming Protocol version 7, as defined in Pantos versions up to 23 and in RFC 8216 [2] and HLS version 8, as defined in RFC 8216bis draft00 [2bis] with some exceptions. If you require a list of such exceptions please contact your Technical Account Manager.

Note that the implementation requires that all segments start with a keyframe. This is not an absolute requirement in the HLS specification, but necessary to achieve smooth change of quality when adapting bitrates.

At least the following M3U8 playlist tags are supported

- #EXTM3U
- #EXTINF
- #EXT-X-TARGETDURATION
- #EXT-X-MEDIA-SEQUENCE
- #EXT-X-KEY
- #EXT-X-ENDLIST
- #EXT-X-STREAM-INF (audio only)
- #EXT-X-DISCONTINUITY
- #EXT-X-DISCONTINUITY-SEQUENCE
- #EXT-X-START
- #EXT-X-VERSION

- #EXT-X-BYTERANGE
- #EXT-X-MEDIA

Container	Audio codecs	Video codecs	Encryption	Decryption trigger	In-band subtitles	MIME type
MPEG2-TS	AAC-LC HE-AAC v1 HE-AAC v2 MP3 Dolby AC3 Dolby AC4 Dolby E-AC-3	H.264 H.265	None		Not supported	application/vnd.apple.mpegurl application/x-mpegURL
MPEG2-TS	AAC-LC HE-AAC v1 HE-AAC v2 MP3 Dolby AC3 Dolby AC4 Dolby E-AC-3	H.264 H.265	AES-128	Manifest	Not supported	application/vnd.apple.mpegurl application/x-mpegURL
ADTS	AAC-LC HE-AAC v1 HE-AAC v2	None	None		Not supported	application/vnd.apple.mpegurl application/x-mpegURL

ADTS	AAC-LC HE-AAC v1 HE-AAC v2	None	AES-128	Manifest	Not supported	application/ vnd.apple.mpegurl application/x-mpegURL
MP3	MP3	None	None		Not supported	application/ vnd.apple.mpegurl application/x-mpegURL
MP3	MP3	None	AES-128	Manifest	Not supported	application/ vnd.apple.mpegurl application/x-mpegURL

Note 1: All rules and restrictions for the support of media formats and codecs applies as outlined in the Video and audio formats section.

Restrictions for HLS content

Devices are able to handle streams with the following limitations:

Parameter	**Requirements**
Frame rate	Up to 60fps
Audio sampling rate	Up to 48000 Hz
Number of audio channels	Up to 8 (7+LFE)
Media segment file size	Up to 15MB

Segment duration	In range 1s - 12s
Average bitrate over one segment	Up to 8 Mbit/s (for up to 1080p)
Manifest file size	Up to 2MB
Number of tracks in one M3U8 manifest file	Up to 36

Devices MAY fail gracefully on streams that do not abide by the following restrictions:

1. Audio/video encoding
    a. The same codec MUST be used across all variant streams (all quality levels).
    b. Audio parameters (number of channels and sample rates) MUST be the same across all variant streams.
2. Media segments
    a. All media segments MUST be independently decodable. Consequently, the first video frame in every segment that contains video MUST be an IDR frame.
    b. Discontinuities in timestamps, frame rate, encoding profiles, or audio/video parameters MUST NOT occur within segments.
3. Playlist files (M3U8)
    a. Audio and video playlists MUST use the same target duration, and MUST contain the same duration of content.
    b. A playlist MUST NOT contain invalid URLs.
    c. Media sequence numbers MUST be aligned across all variant streams (quality levels), so that media sequence numbers can be used to identify matching content.
    d. For live streams, media segments MUST remain available on the server for at least one target duration after the segment disappears from the playlist.
    e. Playlists MUST use sufficiently accurate segment durations to ensure that the sum of the #EXTINF durations of any contiguous group of segments is within one video frame duration of the actual duration.
    f. Playlists MUST provide at least 6 segments in live/linear streams.
    g. Discontinuities in timestamps, frame rate, encoding profiles, or audio/video parameters MAY occur between segments, but such discontinuities MUST be indicated using the #EXT-X-DISCONTINUITY tag.
    h. EXT-X-STREAM-INF tags MUST always provide CODECS and RESOLUTION attributes.
4. Subtitles
    a. The device MUST support subtitles that conform to the Subtitles and Closed Captions
5. DRM
    a. The decryption key MUST be directly downloadable via an HTTP or HTTPS URLs

b. Note that with AES-128 encrypted HLS, segments are completely encrypted.

MPEG-DASH

Profile	Identifier	Reference
ISO Base Media File Format Live	urn:mpeg:dash:profile:isoff-live:2011	[3], section 8.4
ISO Base Media File Format Main	urn:mpeg:dash:profile:isoff-main:2011	[3], section 8.5
DASH-AVC/264	http://dashif.org/guidelines/dash264 urn:com:dashif:dash264	[29], section 6.3
DASH-AVC/264 SD	http://dashif.org/guidelines/dash264#sd	[29], section 7.3
DASH-AVC/264 HD	http://dashif.org/guidelines/dash264#hd	[29], section 8.3
DASH-AVC/264 main	http://dashif.org/guidelines/dash264main	[5], section 8.2
DASH-AVC/264 high	http://dashif.org/guidelines/dash264high	[5], section 8.3

The DVB Profile of MPEG-DASH ([4], section 4.1), identified as "urn:dvb:dash:profile:dvb-dash:2014", is supported.

The following combinations are supported by the device:

Container	Audio codecs	Video codecs	DRM	DRM Trigger	In-band subtitle	MIME type

ISO BMFF	AAC-LC HE-AAC v1 HE-AAC v2 MP3 Dolby AC3 Dolby AC4 Dolby E-AC-3	H.264 H.265	None	None	Supported	application/ dash+xml
ISO BMFF	AAC-LC	AV1	None	None	Supported	application/ dash+xml
ISO BMFF	AAC-LC HE-AAC v1 HE-AAC v2 MP3 Dolby AC3 Dolby AC4 Dolby E-AC-3	H.264 H.265	ClearKey PlayReady Widevine	EME	Supported	application/ dash+xml

Note 1: All rules and restrictions to the support of media formats and codecs applies as outlined in the Video and audio formats section.
Note 2: All rules and restrictions for the support of DRM applies as outlined in the DRM and EME sections.

MPEG-DASH content restrictions

Limitations on MPEG-DASH streams:

Parameter	**Restrictions**
Frame rate	Up to 60fps

Audio sampling rate	Up to 48000 Hz
Number of audio channels	Up to 8 (7+LFE)
Media segment file size	Up to 15MB
Segment duration	In range 1s - 12s
Average bitrate over one segment	Up to 10Mbit/s (for 1080p content)
Manifest file size	Up to 2MB
Number of tracks in one MPD file	Up to 36

Devices may fail gracefully on streams that do not abide by the following restrictions and are not compliant with the [5] DASH-IF Interoperability Points documentation:

- The media segment container format MUST be the ISO Base Media File Format (aka MP4).
- All media segments MUST be independently decodable. Consequently, the first video frame in every segment that contains video MUST be an IDR frame.
- Manifest URLs MAY include the MPD anchor, but MUST NOT use any other than the 't' parameter ([3]section C.4)
- The device MUST support multiple audio tracks associated with multiple Adaptation Sets defined in an MPD of MPEG-DASH.
- Each audio track MUST be a separate media stream.
- Support for the <SegmentList> element is NOT REQUIRED.
- The device MUST support multiple Periods defined in an MPD of MPEG-DASH.

Microsoft Smooth Streaming (MSSS)

Microsoft Smooth Streaming Transport Protocol v2.2 as defined in [2], both Live and On-Demand streams is supported.

NOTE: The version number refers to the MajorVersion and MinorVersion attributes in the manifest, not the Smooth Streaming Protocol Specification version.
As of revision 6.0 of the specification, the only valid versions are 2.0 and 2.2 (see section 2.2.2.1 in [3]).

Container	Audio codecs	Video codecs	DRM	DRM Trigger	In-band subtitle	MIME type
PIFF v1.1 [13]	AAC-LC HE-AAC v1 HE-AAC v2	H.264	None	None	Supported	application/ vnd.ms-sstr+xml
PIFF v1.1 [13]	AAC-LC HE-AAC v1 HE-AAC v2	H.264	PlayReady	Manifest	Supported	application/ vnd.ms-sstr+xml
PIFF v1.1 [13]	AAC-LC HE-AAC v1 HE-AAC v2	H.264	PlayReady	WebInitiator	Supported	application/ vnd.ms-playready.initiator+xml

Note 1: all rules and restrictions to the support of media formats and codecs applies as outlined in the Video and audio formats section.
Note 2: All rules and restrictions to the support of DRM applies according to the DRM and WebInitiator sections.

Smooth Streaming content restrictions

Parameter	Requirements
Frame rate	Up to 60fps
Audio sampling rate	Up to 48000 Hz
Number of audio channels	Up to 8 (7+LFE)

Media segment file size	Up to 15MB
Segment duration	In range 1s - 12s
Average bitrate over one segment	Up to 10Mbit/s (for 1080p content)
Manifest file size	Up to 2MB

Streaming can fail gracefully on streams that do not abide by the following restrictions:

- All media segments MUST be independently decodable. Consequently, the first video frame in every segment that contains video MUST be an IDR frame.
- For live streams that use FragmentLookahead, segments MUST remain available on the server for one DVRWindowLength after they disappear from the Manifest file.

2.3 Media Source Extensions (MSE)

Media Source Extensions is supported according to the MSE specification [7]. The following combinations of containers and codecs are supported:

Container	**Audio codecs**	**Video codecs**
MP4	AAC / MP3	H.264 / H.265
WebM	Opus	VP9
MP4	AAC / MP3	<no video>
WebM	Opus	<no video>
MP4	<no audio>	H.264 / H.265
WebM	<no audio>	VP9

MP4	<no audio>	AV1
MP4	AAC / MP3	AV1

Note: All rules and restrictions to the support of media formats and codecs in MSE applies as outlined in the Video and audio formats section.

2.4 Subtitles and Closed Captioning

To display subtitles or Closed Captions, WebVTT is supported as specified in [15] to the extent that it is supported by the Chromium engine, and the EBU-TT-D text track profile of TTML, as specified in [17], is supported

In-band and out-of-band text tracks are supported as defined below:

Media delivery method	**In-band subtitles**	**Out-of-band subtitles**
Progressive playback	Not supported	Yes
HLS	Not supported	Yes
MPEG-DASH	Yes	Yes
Smooth Streaming	Yes	Yes
MSE	Not supported	Yes

2.5 DRM

As stated in 2.4.1, DRM Support is platform dependent and may vary between manufacturers and device models.

In general, all devices supporting HbbTV will support Microsoft PlayReady, and all Android-based devices will support Google Widevine, but many devices support both PlayReady and Widewine and some also support other DRM systems such as Marlin.

2.5.1 ClearKey

- ClearKey is supported with EME

2.5.2 PlayReady

If Microsoft PlayReady is supported:

- PlayReady security level "2000" will always be supported
- Is supported with EME
- Is supported with WebInitiator
- PlayReady Header Object v4.0.0.0 is supported [8]
- PlayReady Header Object v4.1.0.0 is supported [8]
- On devices supporting Ultra HD/4K (2160p) resolution, security level "3000" has to be supported

Play Ready - License Acquisition

- Device support automatic license acquisition, supporting the following:
  - License Acquisition URLs provided in the manifest MUST override those provided in the PSSH boxes
  - License Acquisition URL provided in the WebInitiator MUST override those provided in the manifest and the PSSH boxes
- Reactive License Acquisition (Post-delivery) is supported with CustomData and License Acquisition override mechanism using LA_URL and drm_custom_data variables passed as GET parameters of video source URL to override global settings

WebInitiator

WebInitiator can be used with Microsoft PlayReadyPlayReady :

- Licence Pre-Acquisition for PlayReady
- Available only together with Microsoft Smooth Streaming (MSSS) content
- Mime type: "application/ms-playready.initiator+xml"

2.5.3 Widevine

If Google Widevine is supported:

- Is supported with EME
- Is supported with security level "L1" on devices supporting Ultra HD/4K (2160p) resolution
- Devices with lower resolution support at least security level "L2"
- Is supported with automatic license acquisition
- Support "server certificate" and "privacy mode" features

2.5.4 Marlin

If Marlin DRM is supported:

- Support content license acquisition using MS3 tokens as well as Marlin Broadband (BB) tokens.
- Is supported for MPEG-DASH media format

2.5.5 ES-128 encrypted streams

- Is always supported for Apple HLS streams
    - At least one of the following encryption algorithms is supported:
        - AES 128 bit keys in CENC mode: Counter Mode (AES-CTR) and Full-Sample encryption
        - AES 128 bit keys in CBCS mode: Cipher Block Chaining mode (AES-CBC) and Sub-Sample encryption

2.5.6 DRM Invocation Methods

Encrypted Media Extensions (EME)

Encrypted Media Extensions (EME) is supported un-prefixed according to the W3C Working Draft 05 July 2016 [6].

Note: EME can only be used in secure contexts, it can not be used on any pages served over HTTP.

The following EME features are supported:

- ClearKey according to EME specification "9.1 Clear Key"
    - With key system ID: "org.w3.clearkey"
    - Supporting initialization data types: CENC [10], WEBM [11], KEYIDS [12]
- Widevine
    - Use key system ID: "com.widevine.alpha"
    - Supports initialization data type: CENC [10], KEYIDS [12]
    - Robustness "HW_SECURE_DECODE" MUST is normally supported (platform dependence)
    - Minimal robustness "SW_SECURE_CRYPTO" is supported for audio
    - Features "server certificate" and "privacy mode" will normally be supported
- PlayReady
    - Use key system ID "com.microsoft.playready"
    - Or key system ID "com.youtube.playready"
    - Supports initialization data type: CENC [10]

EME MUST can be used with MSE and with adaptive streaming with the native player.

- The HTMLMediaElement.onencrypted event is triggered if encrypted content has been detected.
- After a successful call to HTMLMediaElement.setMediaKeys() with valid MediaKeys, content will be fully playable.

Detecting encrypted content and triggering license acquisition is supported from:

- Manifest - with DRM initialization data stored in an adaptive streaming manifest
- Media container - with DRM initialization data stored in a video container (CENC [14]or PIFF [13])

Not all DRM invocation methods and DRM systems are available with all transfer protocols. This is specified in detail in the section for each protocol. Every CDM MUST be supported with all combinations of MSE-supported formats and codecs.

2.6 Video and audio formats

2.6.1 3.6.1. Media container formats

The following media container formats are supported:

- ISO Base Media File Format ISO/IEC 14496-12:2012
  - Streaming-optimized MP4 (moov box before the mdat box)
  - Unoptimized MP4 (mdat box before the moov box)
- WebM
  - Supported if VP9 video codecs are available on the device
- MPEG2-TS ISO/IEC 13818-1:2000
- ADTS / AAC (audio elementary stream)
- MPEG-1 Layer III (audio elementary stream)

2.6.2 Video codecs

Video codec support is platform dependent, but the following video codecs formats are normally supported:

- H264 as specified in [20]
  - All profile/level configurations up to and including High Profile Level 4.1.
- H265 as specified in [19]
  - These two levels are supported when the device supports Ultra HD/4K resolution (2160p)
  - These two levels are supported when the device supports HDR, either HDR10 or HLG10
  - H265 is ONLY supported for MPEG-DASH streaming

- H265/HEVC is supported for profile/level configurations up to and including High Profile Level 4.1
- HEVC Main Level 5 and 5.1
- HEVC Main 10 Level 4.1 and 5.1 is supported for MSE only.

- VP9 as specified in [22]
  - When device supports VP9 then VP9 profile 0 is supported
  - VP9 Profile 2 is is only supported with Media Source Extension(MSE)
  - The following VP9 levels are supported (described in [18]): 1, 1.1, 2.1, 3, 3.1, 4, 4.1

    - When the device supports video in Ultra HD/4K resolution (2160p) then it does support the VP9 levels (described in [18]): 5, 5.1
    - When the device supports HDR, either HDR10 or HLG10, then VP9 profile 2 is supported
- AV1 as specified in [21]
    - When device supports AV1 then AV1 profile 0 (MAIN) is supported.
    - The following AV1 levels are supported (described in [21]): 0, 2.1, 3.0, 3.1, 4.0, 4.1

2.6.3 Audio codecs

Audi codec support is platform dependent, and the following codecs are normally supported:

- HE-AAC v1
- HE-AAC v2
- LC-AAC
- MP3
- Opus
- Dolby AC3/ E-AC3 / AC4
- Dolby ATMOS
  (Dolby ATMOS is not a codec, but a technology embedded inside E-AC3 and AC4 streams)
- Ogg Vorbis

2.6.4 CMAF multimedia format

Common Media Application Format (CMAF) [36] combines and constrains several MPEG specifications to define a multimedia format optimized for delivery of a single adaptive multimedia presentation to a variety of devices, using various adaptive streaming formats, broadcast, download, and storage methods.

AAC audio CMAF tracks

List of supported AAC profiles [36], section 10.3 AAC audio CMAF tracks

AAC profile	**MIME type**	**Codecs parameter**
MPEG-4 AAC (AAC-LC)	audio/mp4	mp4a.40.2
MPEG-4 high efficiency AAC (HE-AAC)	audio/mp4	mp4a.40.5
MPEG-4 high efficiency AAC v2 (HE-AACv2)	audio/mp4	mp4a.40.29

Subtitles and captions

CMAF defines the following formats for carrying subtitles and captions [36], section 11.1 Subtitles and captions overview:

- WebVTT subtitle CMAF tracks (in-band, out-of-band)
- IMSC1 subtitle CMAF tracks - (EBU-TT-D conforms to IMSC text profile). IMSC1 is an application of the Timed Text Markup Language (TTML) for subtitle and caption delivery (in-band, out-of-band)
- CTA-608/CTA-708 captions embedded in video CMAF tracks (in-band)

AVC video CMAF media profiles and brands

AVC video CMAF media profile shall comply to one of the media profiles in table above [36], annex A.2:

Media profile	Profile	Level	Colour primaries	Max frame height	Max frame width	Max frame rate	CMAF File Brand
SD	High	3.1	BT.709 BT.601-7	576	864	60	'cfsd'
HD	High	4.0	BT.709	1080	1920	60	'cfhd'
HDHF	High	4.2	BT.709	1080	1920	60	'chdf'

HEVC video CMAF media profiles and brands

HEVC video CMAF media profile shall comply to one of the media profiles in table above [36], annex B.5:

Media profile	Profile	Level	Colour primaries	Max frame height	Max frame width	Max frame rate	CMAF File Brand
HHD8	Main MainTier	4.1	BT.709	1080	1920	60	'chhd'
HHD10	Main10 MainTier	4.1	BT.709	1080	1920	60	'chh1'
UHD8	Main MainTier	5.0	BT.709	2160	3840	60	'cud8'

UHD10	Main10 MainTier 10-bit	5.1	BT.709 BT.2020	2160	3840	60	‘cud1’
HDR10	Main10 MainTier 10-bit	5.1	BT-2020	2160	3840	60	‘chd1’
HLG10	Main10 MainTier 10-bit	5.1	BT-2020	2160	3840	60	‘clg1’

2.6.5 Media format combinations

Protocols and formats supported

Transport			Formats				Comment
Transmission	Transport	Protocol	Container	Video format	DRM	Subtitles	
Broadcast ***(Note 1)***	Out-of-band	N/A	N/A	N/A	N/A	N/A	N/A for Broadcast, ***Note 2***
	Inband	DVB	MPEG 2-TS	AVC_SD_25, AVC_HD_25	None	None, ***Note 3***	Subtitles handled by TV Middleware
			MP4	AVC_SD_25, AVC_HD_25, HEVC_HD_25_8, HEVC_HD_25_10, HEVC_UHD_25	None	None, ***Note 3***	Subtitles handled by TV Middleware

Broadband	Out-of-band	Any	N/A	N/A	N/A	WebVTT	No container, subtitle served as file
						EBU-TT-D	No container, subtitle served as file
	Inband	Progressive Download	MPEG 2-TS	H.264	None	DVB-Subtitles	DVB Bitmap subtitles, https://www.etsi.org/deliver/etsi_en/300700_300799/300743/01.03.01_60/en_300743v010301p.pdf
						Teletext	**Note 8**
			MP4	H.264, H.265	None	WebVTT	
			WebM	VP8, VP9	None	WebVTT	
			ADTS / AAC	None, ***Note 6***	None	None	

		MP3	None, ***Note 6***	None	None	
	MSE/EME	MP4	H.264, H.265	PlayReady Widevine	None	**Note 4**
		WebM	VP8, VP9	PlayReady Widevine	None	**Note 4**
	HbbTV DASH DVB DASH	MP4	H.264, H.265	PlayReady Marlin	EBU-TT-D WebVTT	**Note 5, Note 10**
	Apple HLS	MPEG 2-TS	H.264	Encryption	WebVTT frag.	Plain Text WebVTT fragments (outside container)
		MP4	H.264, H.265	Encryption	WebVTT	Plain Text WebVTT fragments (outside container)
		ADTS / AAC	None, ***Note 6***	None	None	

			MP3	None, ***Note 6***	Non e	None	
		Microsoft Smooth	PIFF	H.264	Play Rea dy	EBU-TT-D, WebVTT	**Note 5**

Note 1: Broadcast only supported for HbbTV contexts

Note 2: Out-Of-Band Subtitles with track element is not possible with Broadcast, but it is Possible to serve a Subtitle track in a separate HTML5 video element and synchronize this with broadcast (requires the Media Sync Module)

Note 3: *Subtitles NOT handled by the browser*

Note 4: Using MSE, subtitles can either be served and handled with an out-of-band <track> element, or extracted and rendered by JavaScript from the stream

Note 5: EBU-TT-D is a rich profile of Timed Text Markup Language (TTML) defined by DVB. Since TTML is XML-based, most (simpler) TTML based subtitle profiles will work where TTML is stated

Note 6: Audio-only Container, no video format, supported Audio codecs are listed in the table below

Note 7: Multiple parallel video streams and simultaneous video elements are supported, but device support for this is dependent on platform, integration and resources.

Note 8: Rendering of Teletext formatted subtitles served over Progressive Download in MPEG2-TS Containers over broadband is mandated in HbbTV but not supported by Vewd SW!. Vewd believe that there are no real life usecases for this feature and is working to have it deprecated from the HbbTV specification.

Note 9: All Codec support is dependent on and limited by capabilities of the underlying platform and APIs.

Note 10: MPEG DASH is supported according to MPEG DASH IOSFF-live defined by HbbTV (HbbTV 1.5) and MPEG DASH Main Profile, defined by DVB (HbbTV 2.x)

Supported Audio codecs are dependent on container format:

Container	Supported Audio Codecs

MPEG2-TS	AA-LLC, HE-AAC v1, HE-AAC v2, MP3, Dolby AC3, Dolby E-AC-3
MP4 (ISO-BMFF)	AA-LLC, HE-AAC v1, HE-AAC v2, MP3, Dolby AC3, Dolby E-AC-3
MP4 (when served in DASH)	Dolby AC4, MPEG-H
PIFF	AA-LLC, HE-AAC v1, HE-AAC v2
ADTS / AAC	AA-LLC, HE-AAC v1, HE-AAC v2
MP3	MP3
WebM	Opus

2.7 Input handling

2.7.1 Key mappings

A set of standardized key codes are defined, which are mapped from a remote control. The codes in the "JavaScript key code" column are available to web applications and pages in the global JavaScript context.

Hardware key	**Linux key code**	**Android key code**	**JavaScript key code**	**Requirement**
←	OMI_KEY_LEFT	KEYCODE_DPAD_LEFT	VK_LEFT	Mandatory
→	OMI_KEY_RIGHT	KEYCODE_DPAD_RIGHT	VK_RIGHT	Mandatory
↑	OMI_KEY_UP	KEYCODE_DPAD_UP	VK_UP	Mandatory

↓	OMI_KEY_DOWN	KEYCODE_DPAD_DOWN	VK_DOWN	Mandatory
Confirm / Select / OK	OMI_KEY_ENTER	KEYCODE_DPAD_CENTER / KEYCODE_ENTER	VK_ENTER	Mandatory
Back / Return	OMI_KEY_BACK	KEYCODE_BACK	VK_BACK	Mandatory
Exit/Close	N/A	N/A	N/A	CONDITIONALLY REQUIRED*
BLUE	OMI_KEY_BLUE	KEYCODE_PROG_BLUE	VK_BLUE	CONDITIONALLY REQUIRED*
RED	OMI_KEY_RED	KEYCODE_PROG_RED	VK_RED	CONDITIONALLY REQUIRED*
GREEN	OMI_KEY_GREEN	KEYCODE_PROG_GREEN	VK_GREEN	CONDITIONALLY REQUIRED*
YELLOW	OMI_KEY_YELLOW	KEYCODE_PROG_YELLOW	VK_YELLOW	CONDITIONALLY REQUIRED*
Menu	OMI_KEY_MENU	KEYCODE_MENU	VK_MENU	CONDITIONALLY REQUIRED*
0	OMI_KEY_0	KEYCODE_0	VK_0	CONDITIONALLY REQUIRED*
1	OMI_KEY_1	KEYCODE_1	VK_1	CONDITIONALLY REQUIRED*

2	OMI_KEY_2	KEYCODE_2	VK_2	CONDITIONALLY REQUIRED*
3	OMI_KEY_3	KEYCODE_3	VK_3	CONDITIONALLY REQUIRED*
4	OMI_KEY_4	KEYCODE_4	VK_4	CONDITIONALLY REQUIRED*
5	OMI_KEY_5	KEYCODE_5	VK_5	CONDITIONALLY REQUIRED*
6	OMI_KEY_6	KEYCODE_6	VK_6	CONDITIONALLY REQUIRED*
7	OMI_KEY_7	KEYCODE_7	VK_7	CONDITIONALLY REQUIRED*
8	OMI_KEY_8	KEYCODE_8	VK_8	CONDITIONALLY REQUIRED*
9	OMI_KEY_9	KEYCODE_9	VK_9	CONDITIONALLY REQUIRED*
PLAY	OMI_KEY_PLAY	KEYCODE_MEDIA_PLAY	VK_PLAY	CONDITIONALLY REQUIRED*
PAUSE	OMI_KEY_PAUSE	KEYCODE_MEDIA_PAUSE	VK_PAUSE	CONDITIONALLY REQUIRED*
PLAY/PAUSE	OMI_KEY_PLAY_PAUSE	KEYCODE_MEDIA_PLAY_PAUSE	VK_PLAY_PAUSE	CONDITIONALLY REQUIRED*
STOP	OMI_KEY_STOP	KEYCODE_MEDIA_STOP	VK_STOP	CONDITIONALLY REQUIRED*

NEXT	OMI_KEY_TRACK_NEXT	KEYCODE_MEDIA_NEXT	VK_TRACK_NEXT	CONDITIONALLY REQUIRED*
PREV	OMI_KEY_TRACK_PREV	KEYCODE_MEDIA_PREVIOUS	VK_TRACK_PREV	CONDITIONALLY REQUIRED*
FF (Fast-Forward)	OMI_KEY_FAST_FWD	KEYCODE_MEDIA_FAST_FORWARD	VK_FAST_FWD	CONDITIONALLY REQUIRED*
REWIND	OMI_KEY_REWIND	KEYCODE_MEDIA_REWIND	VK_REWIND	CONDITIONALLY REQUIRED*
SUBTITLE	OMI_KEY_SUBTITLE	KEYCODE_CAPTIONS	VK_SUBTITLE	CONDITIONALLY REQUIRED*
AUDIO	OMI_KEY_AUDIO_TRACK	KEYCODE_MEDIA_AUDIO_TRACK	VK_AUDIO_TRACK	CONDITIONALLY REQUIRED*
INFORMATION	OMI_KEY_INFO	KEYCODE_INFO	VK_INFO	CONDITIONALLY REQUIRED*

Navigation and Select keys

Navigation and Select keys are sendt corresponding key codes with key events into the web app according to the DOM Level 3 Events specification ([23]).

Back key

The Back/Return button is a mandatory button on the remote control to go back or close the app. The JavaScript window.VK_BACK is passed to the app, so it can be handled by JavaScript. The caption for this button should be "Back" or "Return" or similar, as long as it remains clear to the end user. The caption should also be consistent with the entire device UI.

3.7.1.3. Exit/Close key

The Exit/Close key is an optional but recommended key on the remote control. The device firmware handles the key, and it should close the app immediately with no event exposed to the app. Its caption should be clear so the end-user understands what they are pressing (example: Exit/Close).

window.close()

When a web app calls window.close() in JavaScript, the device will handle the call by closing the specific window (app) immediately.

Entering text

The will normally provide an on-screen keyboard to input characters into edit or password fields on web pages.

2.8 User Agent string

Vewd Core uses the following User Agent String:

Component	**Comment**
Mozilla/5.0 (<OS> <Architecture>)	• The browser states it's "Mozilla-compatible" • OS - operating system, e.g. Linux or Andr0id (Andr0id is used to avoid confusion with Mobile) • Architecture - CPU architecture, e.g. MIPS or ARM
AppleWebKit/537.36 (KHTML, like Gecko)	WebKit version
Chrome/xx.0.*	The Chrome version
Safari/xxx.xx	Safari version
OPR/xx.0.2207.0	Opera Desktop version
OMI/4.2x.0.*	The Vewd Core version
Model/<CustomerName>-<DeviceModel>	"CustomerName" MUST represent the name of the OEM company "DeviceModel" MUST represent generalized or particular model name of the OEM device (**Note**: not all devices use these fields)

The entire User Agent string MAY resemble the following example:

Mozilla/5.0 (Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.11 Safari/537.36 OPR/46.0.2207.0 OMI/4.22.0.161.master Model/MyManufacturerName-MyModelName

2.8.1 User agent client hints

Vewd Core support the User Agent Client Hints API.

The Chromium version on which Vewd Core is based on together with the Vewd Core version are reported in the brands and fullVersion properties.

2.9 Screen resolution

- Full HD: 1280 x 720
- HD: 1920 x 1080 (1080p)
- UHD or 4K: 2160p

Depending on the resolution of the graphics plane, logical resolution and CSS pixel resolutions ([28]) as outlined in the following table:

Graphics plane resolution	**Window resolution**	
	1280 x 720	**1920 x 1080**
1280 x 720	resolution 1dppx	-
1920 x 1080	resolution 1.5dppx	resolution 1dppx

Note that the resolution seen in a generic browser window, and specified in this document, may differ from the resolution seen by apps running in a HbbTV Windows

3 Security

3.1 Same-Origin policy

Applications must comply withSame-Origin policy with no exceptions. And signal and use CORS requests if required.

3.2 Mixed content

The following active mixed content is NOT allowed:

- <script> (src attribute)
- <link> (href attribute) (this includes CSS stylesheets)
- <iframe> (src attribute)
- XMLHttpRequest requests
- All cases in CSS where a URL value is used (@font-face, cursor, background-image, and so on)
- <object> (data attribute)

Passive/Display mixed content may be allowed, such as:

- <img> (src attribute)
- <audio> (src attribute)
- <video> (src attribute)
- <object> subresources (when an <object> performs HTTP requests)

3.3 Root certificates

Vewd Core powered devices support a common set of Root Certificates that is normally kept updated after the device ships. On Android based platforms, Vewd Core leverages the Root Store on the platform as installed by the manufacturer.

3.4 Cipher suites

The following Chiper suites are normally supported:

Cipher suite	Status
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)	Yes

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)	Yes
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)	Normally yes
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)	Normally yes
TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)	Yes: this is a fallback and should be after all other mentioned ciphers
Cipher suites with anonymous key exchange (TLS_DH_anon_* suites)	No
Cipher suites with NULL encryption	No
Cipher suites using RC4 encryption, for example: • TLS_RSA_WITH_RC4_128_SHA (0x0005) • TLS_RSA_WITH_RC4_128_MD5 (0x0004)	No
Cipher suites using encryption or signing algorithms offering less than 112 bits of security	No

3.5 Debugging tools

Vewd Core supports the development debugging tool (Chrome DevTools) as specified in [25]. This tool will normally ONLY be enabled in special debug versions of the device firmware and therefore in general not be available to content developers

3.6 Default colors

Empty and undefined browser elements will use the following default colors:

Property	**Default color**

Background color	White
Font color	Black

3.7 Storage

The following minimum storage is available for apps:

- 64MB for localStorage ([26])
- 16MB for Temporary Storage ([26])
- 32MB for HTTP cache

The memory for the localStorage is shared between all apps, and each app may use up to 10MB. The localStorage and sessionStorage is limited to 5MB per origin.

The memory for temporary storage is shared between all apps, and each app may use up to 20% of the shared pool (3.2MB).

3.8 Cookies

The platform is capable of storing at least 180 cookies per domain, with at least 4096 bytes (for example, a total of 8MB cookie storage). Persistent cookies will not be removed earlier than 30 days from last access.

3.9 Accessibility

3.9.1 Text-to-speech

Text-to-Speech (TTS) is an accessibility feature enabling audible spoken words of the text strings shown in an application. Users with impaired vision disability benefit from using Text-to-Speech.

Accessibility (Text to Speech) in applications should be implemented by using the Web Speech API, but not all platforms support this.

3.9.2 Peer-to-peer Data API

The Peer-to-peer Data API lets a web application send and receive generic application data peer-to-peer. The API for sending and receiving data uses Web Sockets.

The following interfaces and extensions are supported:

- RTCPeerConnection Interface Extensions
- RTCDataChannel
- RTCDataChannelEvent

The following parts of the WebRTC 1.0 specification are supported over a peer-to-peer connection:

- RTP media API
- Peer-to-peer Data API
- Media Stream API Extensions for Network Use

3.9.3 Media Capture and Streams

The Media Capture and Streams defines a set of JavaScript APIs that allow local media, including audio and video, to be requested from a platform []

Support for Media Capture is platform dependent, but if supported, the MediaDevices object is supported, this is the entry point to the API used to examine and get access to media devices available to the browser.

Access to the sources:

- Access to the Network resources
- Access to the Physical webcam and Microphone

3.10 Web Audio API

The Web Audio API specification describes a high-level Web API for processing and synthesizing audio in web applications.

This is supported as defined in the W3C Recommendation of the Web Audio API [39].

4 Abbreviations

AAC	Advanced Audio Coding
AC-3	(Dolby Digital) Audio Compression 3
ADTS	Audio Data Transport Stream
CENC	Common Encryption
CDM	Content Decryption Module
DASH	Dynamic Adaptive Streaming over HTTP
DASH-IF	DASH Industry Forum
DRM	Digital Rights Management
E-AC-3	(Dolby Digital) Enhanced Audio Compression 3
EBU-TT-D	EBU Timed Text format, part D - the format for the distribution of subtitles over IP
EME	Encrypted Media Extensions
HbbTV	Hybrid Broadcast Broadband TV
HDR	High Dynamic Range
HE-AAC	High-Efficiency Advanced Audio Coding
HLS	HTTP Live Streaming

HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDR	Instantaneous Decoder Refresh
ISO BMFF	ISO base media file format
KEYIDS	Stream-independent format for specifying a list of key ID(s) for DRM initialization
LC-AAC	Low-Complexity Advanced Audio Coding
LFE	Low Frequency Effects
MPD	DASH Media Presentation Description
MPEG	Moving Picture Experts Group
MPEG2-TS	MPEG-2 Transport Stream
MSS	Microsoft Smooth Streaming
PIFF	Protected Interoperable File Format
SHA	Secure Hash Algorithm
TLS	Transport Layer Security
TTF	TrueType Font
TTML	Timed Text Markup Language

WebM	WebM Stream Format
WebVTT	Web Video Text Tracks format
WOFF	Web Open File Format
WOFF2	WOFF File Format 2.0

5 References

[1] Smooth Streaming Protocol - https://msdn.microsoft.com/en-us/library/ff469518.aspx

[2] RFC 8216 - HTTP Live Streaming - https://datatracker.ietf.org/doc/html/rfc8216

[2bis] RFC 8216bis - HTTP Live Streaming - https://datatracker.ietf.org/doc/html/draft-pantos-hls-rfc8216bis

[3] DASH, Information technology — Dynamic adaptive streaming over HTTP (DASH): ISO/IEC 23009-1:2014

[4] DVB MPEG-DASH Profile for Transport of ISO BMFF - https://www.dvb.org/resources/public/standards/a168_dvb-dash.pdf

[5] DASH-IF Guidelines for Implementation - Interoperability Points v3.2 - http://dashif.org/wp-content/uploads/2015/12/DASH-IF-IOP-v3.2.pdf

[6] Encrypted Media Extensions - W3C Candidate Recommendation 05 July 2016 - https://www.w3.org/TR/2016/CR-encrypted-media-20160705

[7] Media Source Extensions - W3C Candidate Recommendation 05 July 2016 - https://www.w3.org/TR/2016/CR-media-source-20160705

[8] PlayReady documents - https://www.microsoft.com/playready/documents/

[9] HTML 5.1 W3C Working Draft, 2 June 2016 - https://www.w3.org/TR/html51/

[10] "cenc" DRM Initialization Data Format - https://www.w3.org/TR/eme-initdata-cenc/

[11] "webm" DRM Initialization Data Format - https://www.w3.org/TR/eme-initdata-webm/

[12] "keyids" DRM Initialization Data Format - https://www.w3.org/TR/eme-initdata-keyids/

[13] Protected Interoperable File Format - http://www.iis.net/learn/media/smooth-streaming/protected-interoperable-file-format

[14] Common Encryption Scheme - ISO/IEC 23001-7

[15] Web Video Text Tracks Format (WebVTT) - https://w3c.github.io/webvtt/

[16] TTML text track format - https://www.w3.org/TR/ttml-imsc1/

[17] EBU-TT-D text track format - https://tech.ebu.ch/docs/tech/tech3380.pdf

[18] Definition of video VP9 levels - http://www.webmproject.org/vp9/levels

[19] High efficiency video coding - International Telecommunication Union (ITU-T) Recommendation for h.265 - https://www.itu.int/rec/T-REC-H.265

[20] Advanced video coding for generic audiovisual services - International Telecommunication Union (ITU-T) Recommendation for h.264 - https://www.itu.int/rec/T-REC-H.264

[21] AV1 Bitstream & Decoding Process Specification - https://aomediacodec.github.io/av1-spec/av1-spec.pdf

[22] (Draft) VP9 Bitstream and Decoding Process Specification - http://www.webmproject.org/vp9/

[23] DOM Level 3 Events specifcation - https://www.w3.org/TR/DOM-Level-3-Events/

[24] Mozilla CA Certificate Policy - https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/policy

[25] Chromium Remote Debugging protocol (DevTools) - https://developer.chrome.com/devtools/docs/debugger-protocol

[26] Web Storage API - https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API

[27] HTML5 Persistant Storage - https://developer.chrome.com/apps/offline_storage#persistent

[28] CSS Values and Units Module Level 3, Chapter 6.4: Resolution units - https://www.w3.org/TR/2016/CR-css-values-3-20160929/#resolution

[29] Guidelines for Implementation: DASH-AVC/264 Interoperability Points - http://dashif.org/wp-content/uploads/2015/04/DASH-AVC-264-v2.00-hd-mca.pdf

[30] Key words for use in RFCs to Indicate Requirement Levels - https://www.ietf.org/rfc/rfc2119.txt

[31] ETSI TS 102 796 V1.4.1 Hybrid Broadcast Broadband TV Technical Specification - http://www.etsi.org/deliver/etsi_ts/102700_102799/102796/01.04.01_60/ts_102796v010401p.pdf

[32] IETF RFC 5246 - https://tools.ietf.org/html/rfc5246

[33] IETF RFC 5289 - https://tools.ietf.org/html/rfc5289

[34] HbbTV 2.0.1 Specification with Errata #2 Integrated and Changes Marked - https://www.hbbtv.org/wp-content/uploads/2018/02/HbbTV-SPEC20-00061-000-HbbTV_201_with_errata_2_integrated.pdf

[35] Accessible Rich Internet Applications (WAI-ARIA) 1.1 - https://www.w3.org/TR/wai-aria/

[36] ISO/IEC 23000-19:2018 Information technology — Multimedia application format (MPEG-A) — Part 19: Common media application format (CMAF) for segmented media https://www.iso.org/standard/71975.html

[37] WebRTC 1.0: Real-Time Communication Between Browsers https://www.w3.org/TR/webrtc/

[38] Media Capture and Streams: https://www.w3.org/TR/mediacapture-streams/

[39] Web Audio API: https://www.w3.org/TR/2021/REC-webaudio-20210617/

6 Annex A (informative). Media type strings for video and audio codecs

This section specifies codecs and media identifiers for audio and video codecs that MUST be supported with the HTML5 <video> tag. It also provides identifiers for some examples of combinations of those codecs and media containers. The Codec ID strings' list is not comprehensive, because not all the rules from section 3.6. Video and audio formats are shown below.

6.1 MP4 video and audio

6.1.1 H.264 profiles

Profile	**Level**	**Codec ID string [**rfc6381**]**
Baseline	3.1	avc1.42E01F avc3.42E01F
Main	3.1	avc1.4D401F avc3.4D401F
Main	4	avc1.4D4028 avc3.4D4028
High	4	avc1.640028 avc3.640028

6.1.2 H.265/HEVC profiles

Profile	**Level**	**Constraints**	**Codec ID string**
Main	3.1	None	hev1.1.6.L93.00 hvc1.1.6.L93.00

	4	None	hev1.1.6.L120.00 hvc1.1.6.L120.00
	4.1	None	hev1.1.6.L123.00 hvc1.1.6.L123.00
	5.1	None	hev1.1.6.L153.00 hvc1.1.6.L153.00
Main 10	4.1	None	hev1.2.4.L123.00 hvc1.2.4.L123.00
	5.1	None	hev1.2.4.L153.00 hvc1.2.4.L153.00

6.1.3 AV1 profiles

Profile	**Level**	**Constraints**	**Codec ID string**
Main	2.0	None	av01.0.00M.08
	2.1	None	av01.0.01M.08
	3.0	None	av01.0.04M.08
	3.1	None	av01.0.05M.08
	4.0	None	av01.0.08M.08
	4.1	None	av01.0.09M.08

6.1.4 AAC profiles

Profile name	Codec ID string
AAC-LC	mp4a.40.2
HE-AAC v1 (SBR)	mp4a.40.5
HE-AAC v2 (SBR+PS)	mp4a.40.29

6.1.5 MP3 (MPEG-1 Layer III) profiles

Codec ID string
mp4a.69
mp4a.6B

6.1.6 Dolby profiles

Profile name	Codec ID string
AC-3	ac-3
	mp4a.a5
E-AC-3	ec-3
	mp4a.a6
AC-4	ac-4

ATMOS (technology embedded in the codec)	ec-3, ec+3

6.1.7 Combination examples of media type strings

Video codec	Video profile	Audio codec	Audio profile	Media type string
H.264 level 3.1	baseline	AAC	aac_he	video/mp4; codecs="avc1.42E01F, mp4a.40.5"
			aac_lc	video/mp4; codecs="avc1.42E01F, mp4a.40.2"
		MP3		video/mp4; codecs="avc1.42E01F, mp4a.69"
				video/mp4; codecs="avc1.42E01F, mp4a.6B"
H.264 level 3.1	main	AAC	aac_he	video/mp4; codecs="avc1.4D401F, mp4a.40.5"
			aac_lc	video/mp4; codecs="avc1.4D401F, mp4a.40.2"
		MP3		video/mp4; codecs="avc1.4D401F, mp4a.69"

				video/mp4; codecs="avc1.4D401F, mp4a.6B"
H.264 level 4.0	main	AAC	aac_he	video/mp4; codecs="avc1.4D4028, mp4a.40.5"
			aac_lc	video/mp4; codecs="avc1.4D4028, mp4a.40.2"
		MP3		video/mp4; codecs="avc1.4D4028, mp4a.69"
				video/mp4; codecs="avc1.4D4028, mp4a.6B"
H.264 level 4.0	high	AAC	aac_he	video/mp4; codecs="avc1.640028, mp4a.40.5"
			aac_lc	video/mp4; codecs="avc1.640028, mp4a.40.2"
		MP3		video/mp4; codecs="avc1.640028, mp4a.69"
				video/mp4; codecs="avc1.640028, mp4a.6B"

6.2 WebM video and audio

Video format	Video codec	Audio codec	Media type string
WebM	VP9 profile 0	Opus	video/webm; codecs="vp9, opus"
			video/webm; codecs="vp9.0, opus"
	VP9 profile 2	Opus	video/webm; codecs="vp9.2, opus"

6.3 Audio only

Audio codec	Audio profile	Media type string
AAC	aac_he	audio/mp4; codecs="mp4a.40.5"
	aac_lc	audio/mp4; codecs="mp4a.40.2"
MP3		audio/mp4; codecs="mp4a.69"
		audio/mp4; codecs="mp4a.6B"
Ogg Vorbis		audio/ogg; codecs="vorbis"

7 Revision History

Version Number	Chromium	Date	Comment
4.9.0	53.0	2016-08-18	Publication release
4.9.0-r1	53.0	2016-11-04	Changes: • Updated Chromium version • Fixed MSE reference • Removed text about terminal-defined behaviour in the exit key. The application may now manage this key event freely. • The Application MUST handle both VK_BACK_SPACE and VK_BACK. • The Application MUST support HD, Full HD or both resolutions • Removed mention of viewport META tag • The Application MUST explicitly set colors • Restructured content • Improved clarity and grammar
4.10	56.0	2017-03-16	Publication release Changes: • Added “” • Added “” table
4.10-r1	56.0	2017-09-14	• Rebranded to a new company entity

4.11	59.0	2018-02-19	Changes: • Updated Chromium version • Fixed MSE reference • Added sections on HDR and 4K video support
4.12	63.0	2018-04-03	Changes: • Added cipher suites requirements: section • Added Advert Insertion performance requirements: section • Devtools must be disabled: section • Added requirement for MPEG-DASH period in section • Added requirement for PlayReady automatic license acquisition:
4.13	67.0	2018-07-20	Changes: • Added text-to-speech requirements in a new chapter.
4.13-r1	67.0	2018-02-05	Changes: • Added AV1 codec requirements: section • Added Ogg Vorbis audio codec requirements: section • Added CustomData and License Acquisition override feature optional requirement • Removed VP8 requirement.

4.20	77.0	2019-09-20	Changes: • Added Dolby AC4 codec and Dolby Atmos requirements: section • Added Marlin DRM requirements: section • Added image rendering performance requirements: section
4.20-r1	77.0	2019-12-19	Changes: • Added CBCS requirement for PlayReady and Widevine DRM: section • Added CMAF requirement: • Stricter Widevine and PlayReady DRM security level requirement
4.21	84.0	2020-07-24	No changes in requirements.
4.22	92.0	2021-09-02	Changes: • Added AV1 codec support in MPEG-DASH streaming protocol requirement:
4.22-r1	92.0	2022-02-06	Changes: • Added WebRTC requirements: 3.14. WebRTC • Added Widevine automatic license acquisition:
4.23.0	102.0	2022-09-16	Changes: • Updated section

4.24.0	114.0	2023-06-07	Changes: • Added WebAudio requirements: • Added requirement for PlayReady key system ID "com.youtube.playready"
4.25.0		2024-06	Changes: • Moved document to TiVo.com • Removed legacy information (except this change log) • Cleaned up language • Inserted new table 3.6.5 Media Format combinations
4.25		2024-08	Changes ▪ added info on keycodes for AUDIO and PLAY/PAUSE buttons
4.25		2025-01	Changes • fixed references to HLS