AAMP UVE – API Documentation

V2.10

Table of Contents

Unified Video Engine (UVE) APIs

Overview

AAMP is an open source native video engine that is built on top of GStreamer and optimized for performance, memory use, and code size. AAMP Reference Player demonstrates how to use the Unified Video Engine (UVE) JavaScript binding APIs to interact with an AAMP player.

The bindings are made available in JavaScript with the help of the injectedbundle component once the DOM elements are loaded by WebKit.

Target Audience

This document is targeted to OTT app vendors and HTML5 developers who are interested in evaluating/adopting AAMP for their media player applications on settops running RDKV based firmware.

Features

- Formats: HLS, DASH, Fragmented MP4 HLS
- DRM Systems: Clear Key, Adobe Access, Vanilla AES-128, PlayReady, Widevine
- Captions: CEA-608/708 Captions , WebVTT

Roadmap

- Video Guard (VGC) DRM
- DVB, EBU-TT captions

Release Version

S.No.	Release Version	Release Notes
1	0.7	Initial draft of UVE APIs implemented
2	0.8	CDAI support, configuration options for tune optimization API: • setAlternateContent • notifyReservationCompletion • addCustomHTTPHeader Configuration: • stereoOnly • asyncTune • bulkTimedMetadata • useWesterosSink • parallelPlaylistDownload Events: • bufferingChanged • timedMetadata • adResolved • reservationStart • reservationEnd • placementStart • placementEnd • placementProgress • placementError
3	0.9	"Player Switching" Feature • load (autoplay=false support) • detach() method
4	1.0	Added support to get available audio track and closed captioning info API: • getAvailableAudioTracks • getAvailableTextTracks Configuration: • playlistTimeout • parallelPlaylistRefresh

		• useAverageBandwidth • preCachePlaylistTime • progressReportingInterval • useRetuneForUnpairedDiscontinuity • drmDecryptFailThreshold
5	2.4	April 2020 Release Update Configuration • initialBuffer • useMatchingBaseUrl • initFragmentRetryCount Event Notification
6	2.6	June 2020 Release Update Seek while paused, get/set audio and text track supported API: • getAudioTrack • setAudioTrack • getTextTrack • setTextTrack • setClosedCaptionStatus • setTextStyleOptions • getTextStyleOptions Configuration: • nativeCCRendering • langCodePreference • descriptiveTrackName
7	2.7	Aug 2020 Release Update Configuration • Deprecated useWesterosSink
8	2.9	Sept 2020 Release Update Configuration • authToken • useRetuneForGstInternalError

9	2.10	Oct 2020 Release update. • Updated getAvailableAudioTracks / getAvailableTextTracks API: • setAudioLanguage Configuration: • propagateUriParameters • reportVideoPTS ATSC – UVE Features Added .

Minimal Sample Player

```
<html><head><title>IP Video Playback in WPE browser using UVE API</title></head>
    <script>
window.onload = function() {
    var player = new AAMPMediaPlayer();
    var url = "https://cpetestutility.stb.r53.xcal.tv/multilang/main.m3u8";
    player.load(url);
}
</script>
    <body>
        <div id="videoContainer">
              <video style="height:100%; width:100%; position:absolute; bottom:0; left:0">
                    <source src="dummy.mp4" type="video/ave"> <!-- hole punching -->
              </video>
        </div>
    </body>
</html>
```

General Setup

To setup the AAMP Reference Player in RDK devices(Comcast):

a. Host the ReferencePlayer folder in a web server.

b. Use Comcast's IBIS tool (https://ibis.comcast.com/app-dev-tool/send-html-app) to launch the reference player in the device:

   a. Under Launch HTML App, select **Select a device to get started**.
   b. From the list, find your device (it should be registered previously).
   c. Enter the ReferencePlayer URL in the **URL** field.
   d. Enter any name in the **App name** field.
   e. Click **Launch**.

Folder Structure: Full Reference Player

```
-icons                  // UI elements of reference players and homepage
-UVE
 -index.html            // Homepage of UVE reference player
 -UVEMediaPlayer.js     // Includes "AAMPPlayer" JS class which wraps UVE binding object AAMPMediaPlayer
 -UVEPlayerUI.js        // JS code for the UI elements and their functionality
 -UVERefPlayer.js       // Main JS file
```

```
  -UVERefPlayerStyle.js      // JS code for reference player and its UI
-index.html                  // Homepage of reference player
-ReferencePlayer.js          // JS code for Homepage and redirection to respective reference players
-URLs.js                     // list of selectable streams
-ReferencePlayerStyle.css    // CSS for Homepage and its UI
```

Universal Video Engine APIs

PROPERTIES:

Name	Type	Description
version	number	May be used to confirm if RDKV build in use supports a newer feature

METHODS:

load(uri, autoplay)

- Begin streaming.

Name	Type	Description
Uri	String	URI of the Media to be played by the Video Engine
autoplay	Boolean	optional 2nd parameter (defaults to true) If false, causes stream to be prerolled/prebuffered only, but not immediately automatically presented. Available starting with version 0.8.

play()

- Supported UVE version 0.7 and above.
- Start playback (if stream is in prebuffered state), or resume playback at normal speed. Equivalent to setPlaybackRate(1).

pause()

- Supported UVE version 0.7 and above.
- Pauses playback. Equivalent to setPlaybackRate(0).

stop()

- Supported UVE version 0.7 and above.
- Stop playback and free resources associated with playback.

seek(offset)

- Supported UVE version 0.7 and above.
- Specify initial or new stream playback position. May be called prior to first load() call (or implicitly using initConfig's "offset" parameter), or while streaming.

Name	Type	Description
offset	Number (s)	Offset from beginning of VOD asset. For live playback, offset is relative to eldest portion of initial window. Offset value should be in seconds Note that ability to seek is currently limited to fragment granularity.
keepPause	Boolean	Flag indicates if player was in paused state before seek then maintain the same state post seek Available starting with version 2.6

getCurrentPosition()

- **Supported UVE version 0.7 and above.**
- Returns current playback position in seconds.

getCurrentState()

- Supported UVE version 0.7 and above.
- Returns one of below logical player states as number:

State Name	Value	Semantics	Remarks
idle	0	eSTATE_IDLE	Player is idle
initializing	1	eSTATE_INITIALIZING	Player is initializaing resources to start playback
	2	eSTATE_INITIALIZED	Player completed playlist download and metadata processing.
	3	eSTATE_PREPARING	Create internal resources required

State Name	Value	Semantics	Remarks
			for DRM decryption and playback
	4	eSTATE_PREPARED	Required resources are initialized successfully
	5	eSTATE_BUFFERING	When player does internal buffering mid-playback. Note -send out in initial buffering
paused	6	eSTATE_PAUSED	Indicates player is paused
seeking	7	eSTATE_SEEKING	Indicates player is seeking
playing	8	eSTATE_PLAYING	Indicates player has started playback
	9	eSTATE_STOPPING	Not supported, for future
	10	eSTATE_STOPPED	Not supported, for future
	11	eSTATE_COMPLETE	When the media reches end.
	12	eSTATE_ERROR	In case any error occurred
	13	eSTATE_RELEASED	Not supported, for future

getDurationSec()

- Supported UVE version 0.7 and above.
- Returns current duration of content in seconds. Duration is fixed for VOD content, but may grow with DVR content.

getVolume()

- Supported UVE version 0.7 and above.
- Get current volume (value between 0 and 100). Default audio volume is 100. Volume is normally mapped from remote directly to TV, with video engine used to manage an independent mute/unmute state for parental control.

setVolume (volume)

- Supported UVE version 0.7 and above.
- Sets the current volume (value between 0 and 100). Updated value reflected in subsequent calls to getVolume()

Name	Type	Description
volume	Number	Pass zero to mute audio. Pass 100 for normal (max) audio volume.

setVideoMute(enabled)

- Supported UVE version 0.7 and above.
- Enable or black out video for parental control purposes, default is false

Name	Type	Description
volume	Number	Pass false to black out video. Pass true to resume presenting video.

getPlaybackRate()

- Supported UVE version 0.7 and above.
- Returns the current playback rate.

setPlaybackRate(rate)

- Supported UVE version 0.7 and above.
- Change playback rate, supported speeds are given below -

Value	Description
0	Pause
1	Normal Play
4	2x Fast Forward (using iframe track)
16	4x Fast Forward (using iframe track)
32	8x Fast Forward (using iframe track)
64	16x Fast Forward (using iframe track)
-4	2x Rewind (using iframe track)
-16	4x Rewind (using iframe track)
-32	8x Rewind (using iframe track)
-64	16x Rewind (using iframe track)

getVideoBitrates()

- Supported UVE version 0.7 and above.
- Return array of available video bitrates across profiles.

getCurrentVideoBitrate()

- Supported UVE version 0.7 and above.
- Return current video bitrate, as bits per second.

setVideoBitrate(bitrate)

- Supported UVE version 0.7 and above.

Name	Type	Description
bitrate	Number	Pass bitrate from getVideoBitrates to disable ABR and lock playback to single profile. Pass zero to (re)enable ABR, allowing Video Engine to select from available bitrates based on network bandwidth.

getCurrentAudioBitrate()

- Supported UVE version 0.7 and above.
- Return current audio bitrate, as bits per second.

setVideoRect(x, y, w, h)

- Supported UVE version 0.7 and above.
- Set display video rectangle coordinates. Note that by default video will be fullscreen.
- Rectangle specified in "graphics resolution" coordinates (coordinate space used by graphics overlay).
- Window size is typically 1280x720, but can be queried at runtime as follows:
  var w = window.innerWidth || document.documentElement.clientWidth ||document.body.clientWidth;
  var h = window.innerHeight|| document.documentElement.clientHeight|| document.body.clientHeight;

Name	Type	Description
X	Number	Left position for video
Y	Number	Top position for video.
W	Number	Video width.
H	Number	Video height.

setVideoZoom(videoZoom)

- Supported UVE version 0.7 and above.
- Set video zoom, by default its set to "full"

Name	Type	Description
videoZoom	String	"none" to disable video zoom mode. "full" to enable video zoom mode.

addCustomHTTPHeader(headerName, headerValue, isLicenseRequest)

- Supported UVE version 0.8 and above.
- Add custom headers to HTTP requests

Name	Type	Description
headerName	String	HTTP header name
headerValue	String Array	HTTP header value
isLicenseRequest	Boolean	(defaults to false) indicates if the HTTP header is for exclusive use with PlayReady/Widevine license requests

removeCustomHTTPHeader(headerName)

- Supported UVE version 0.8 and above.
- Remove a custom header set previously. If called with no arguments, will remove all custom headers.

Name	Type	Description
headerName	String	HTTP header name

getAvailableAudioTracks()

- Supported UVE version 1.0 and above.
- Returns the available audio tracks information in the content.

DASH

Name	Type	Description
name	String	Human readable language name e.g: Spanish,English.
language	String	Specifies dominant language of the audio e.g: spa,eng
rendition	String	Role for DASH If not present, the role is assumed to be main e.g: caption,subtitle,main.
characteristics	String	Not mapped
Channels	String	Indicates the maximum number of audio channels 1 = mono, 2=stereo, up to 8 for DD+
bandwidth	String	Represents variants of the bitrates available for the media type e.g: 288000
codec	String	codec associated with Adaptation Set. e.g: mp4a.40.2

Example:

```
{
       "name": "5",
       "language":     "ger",
       "codec":        "mp4a.40.2",
       "rendition":    "german",
       "bandwidth":    288000
}
```

Reference:

```
<AdaptationSet id="3" contentType="audio" segmentAlignment="true" bitstreamSwitching="true" lang="ger">
<Role schemeIdUri="urn:mpeg:dash:role:2011" value="german"/>
<Representation id="5" mimeType="audio/mp4" codecs="mp4a.40.2" bandwidth="288000" audioSamplingRate="48000"
>
<AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="1"/>
</AdaptationSet>
```

HLS

Name	Type	Description
name	String	The value is a quoted-string containing a human-readable description of the Rendition e.g:english, commentary, german
language	String	Identifies the primary language used in the Rendition. In practice, this should be present in vast majority of production manifests, but per HLS specification, this attribute is OPTIONAL e.g: eng,ger,spa.
codec	String	Comma-delimited list of formats, where each format specifies a media sample type that is present in one or more Renditions specified by the Variant Stream. e.g: mp4a.40.2,avc1.4d401e

rendition	String	Specifies the group to which the Rendition belongs. GROUP-ID for HLS.
characteristics	String	One or more comma-delimited Uniform Type Identifiers [UTI]. This attribute is OPTIONAL.
bandwidth	String	Decimal-Integer encoding - bits per second. Represents peak segment bit rate of the Variant Stream.
channels	String	Indicates maximum number of audio channels present in any Media Segment in the Rendition e.g: An AC-3 5.1 rendition would have a CHANNELS=6

```
Example:

  {
            "name": "6",
          "language":     "eng",
          "codec":        "mp4a.40.2",
          "rendition":    "english",
          "bandwidth":    288000
  }

Reference

#EXT-X-MEDIA:TYPE=AUDIO,GROUP-
ID="mono",NAME="english",LANGUAGE="eng",URI="hls/en.m3u8",DEFAULT=YES,AUTOSELECT=YES
#EXT-X-STREAM-INF:PROGRAM-
ID=1,AUDIO="mono",BANDWIDTH=800000,RESOLUTION=640x360,CODECS="avc1.4d400d,mp4a.40.2"
hls/360p.m3u8
```

getAvailableTextTracks()

- Supported UVE version 1.0 and above.
- Returns the available text tracks(CC) in the content.

DASH

Name	Type	Description
name	String	Human readable language name e.g: sub_eng
language	String	iso language code. The language should be present. If not present, the language is unknown or no language applies. e.g: eng
codec	String	Codecs used for the Adaptation Set e.g: stpp
type	String	The value specifies the media type. Valid strings are AUDIO, VIDEO, SUBTITLES and CLOSED-CAPTIONS. This attribute is REQUIRED. e.g: CLOSED-CAPTIONS
rendition	String	Role for DASH If not present, the role is assumed to be main e.g: caption,subtitle,main.
characteristics	String	Not mapped
instreamId	String	Not mapped.

Example:

```
{
    "name": "caption_en",
    "type": "SUBTITLES",
    "language":     "en",
    "codec":        "text/vt
}
```

Reference:

```
</AdaptationSet>
    <AdaptationSet group="14" mimeType="text/vtt" lang="en">
      <Representation id="caption_en" bandwidth="256">
        <BaseURL>subtitles/subtitles_en.vtt</BaseURL>
      </Representation>
    </AdaptationSet>
```

HLS

Name	Type	Description
name	String	Human-readable description of the Rendition. e.g:english,spanish
type	String	Specifies the media type. Valid strings are AUDIO, VIDEO, SUBTITLES and CLOSED-CAPTIONS. This attribute is REQUIRED. e.g: CLOSED-CAPTIONS
language	String	Identifies the primary language used in the Rendition. This attribute is OPTIONAL. e.g: es
rendition	String	Specifies the group to which the Rendition belongs. GROUP-ID for HLS.
instreamId	String	Specifies a Rendition within the segments in the Media Playlist. This attribute is REQUIRED if the TYPE attribute is CLOSED-CAPTIONS e.g: "CC1", "CC2", "CC3", "CC4", or "SERVICEn" where n MUST be an integer between 1 and 63

codec	String	Comma-delimited list of formats, where each format specifies a media sample type that is present in one or more Renditions specified by the Variant Stream.
characteristics	String	Pne or more comma-delimited Uniform Type Identifiers [UTI]. This attribute is OPTIONAL.

Example:

```
{
        "name": "Deutsch",
        "type": "SUBTITLES",
        "language":     "de",
        "rendition":    "subs"
}
```

Reference

```
#EXT-X-MEDIA:TYPE=SUBTITLES,GROUP-
ID="subs",NAME="Deutsch",DEFAULT=NO,AUTOSELECT=YES,FORCED=NO,LANGUAGE="de",URI="subtitles_de.m3u8"
#EXT-X-STREAM-INF:PROGRAM-
ID=1,BANDWIDTH=258157,CODECS="avc1.4d400d,mp4a.40.2",AUDIO="stereo",RESOLUTION=422x180,SUBTITLES="subs"
```

getVideoRectangle()

- Supported UVE version 1.0 and above.
- Returns the current video rectangle co-ordinates.

getAudioTrack( )

- Supported UVE version 2.6 and above.
- Returns the index of current audio track in available audio track list.

setAudioTrack(index)

- Supported UVE version 2.6 and above.
- Set the audio track language from available audio track list.

Name	Type	Description
index	Number	Track Index of desired audio track in available audio track list

setAudioLanguage(language)

- Supported UVE version 2.10 and above.
- Set the audio track language from available audio track list.

Name	Type	Description
language	String	Language of desired audio track in available audio track list

getTextTrack( )

- Supported UVE version 2.6 and above.
- Returns the index of current text track in available text track list.

setTextTrack(trackIndex)

- Supported UVE version 2.6 and above.
- Set the text track at trackIndex in available text track list.

Name	Type	Description
trackIndex	Number	Index of desired text track in available text track list

setClosedCaptionStatus (status)

- Supported UVE version 2.6 and above.
- Set the ClosedCaption rendering to on/off.

Name	Type	Description
Status	Boolean	To turn on/off ClosedCaption rendering

getTextStyleOptions ( )

- Supported UVE version 2.6 and above.
- Returns the JSON formatted string of current ClosedCaption style options and values.

setTextStyleOptions (options)

- Supported UVE version 2.6 and above.

- Set the ClosedCaption style options to be used for rendering.

Name	Type	Description
options	String	JSON formatted string of different rendering style options and its values

EVENTS

Event Name	Event Payload	Description
playbackStarted		- Supported UVE version 0.7 and above. - fired when playback starts
playbackStateChanged	state: number	- Supported UVE version 0.7 and above. - fired as state changes across play/pause seek/not-seek quadruplet
playbackProgressUpdate	durationMiliseconds: number, positionMiliseconds: number, playbackSpeed: number, startMiliseconds: number, endMiliseconds: number, currentPTS: number, videoBufferedMiliseconds : number	- Supported UVE version 0.7 and above. - fired based on the interval set - Added video PTS reporting if enabled with reportVideoPTS config - Added video buffer value (2.4 version)
bufferingChanged	buffering: bool	- Supported UVE version 0.8 and above.

		- fired when AAMP encounters buffering mid-playback, buffering flag indicates buffering status (on/off)
playbackCompleted		- Supported UVE version 0.7 and above. - fired when there is nothing left to play
playbackSpeedChanged	speed: number, reason: string	- Supported UVE version 0.7 and above.
playbackFailed	shouldRetry: boolean, code: number, description: string	- Supported UVE version 0.7 and above. - fired when an error occurs
decoderAvailable	decoderHandle: number	- Supported UVE version 0.7 and above. - fired when video decoder handle becomes available, required for closedcaption parsing + rendering by RDK ClosedCaptions module
mediaMetadata	durationMiliseconds: number, languages: string[], bitrates: number[], playbackSpeeds: number[],	- Supported UVE version 0.7 and above. - fired with metadata of the asset currently played, includes duration(in ms), audio language list, available bitrate list, hasDrm, supported playback speeds

	width: number, height: number, hasDrm: boolean	
speedsChanged	playbackSpeeds: number[]	- Supported UVE version 0.7 and above. - fired when supported playback speeds changes (based on iframe availability)
vttCueDataListener	start: number, duration: number, text: string	- Supported UVE version 0.7 and above. - fired for VTT cue parsed from the WebVTT playlist in the asset
drmMetadata	code: number, description: string	- Supported UVE version 0.7 and above. - fired when there is a change in DRM metadata (especially expiration of DRM auth data)
enteringLive		- Supported UVE version 0.7 and above. - fired when entering live point of a live playlist during/after a seek/trickplay operation
timedMetadata	time: number, duration: number, name: string, content: string, type: number, metadata: object, id: string	- Supported UVE version 0.8 and above. - fired when a subscribed tag is found in the playlist

bitrateChanged	time: number, bitRate: number, description: string, width: number, height: number, framerate: number position: number	- Supported UVE version 0.7 and above. - fired when video profile is switched by ABR with the metadata associated with newly selected profile.
adResolved	resolvedStatus: bool, placementId: string, placementStartTime: number, placementDuration: number	- Supported UVE version 0.8 and above. - Confirmation that an upcoming ad's main manifest has been successfully downloaded and parsed.
reservationStart	adbreakId: string, time: number	- Supported UVE version 0.8 and above. - Sent upon playback into an ad break (one or more ads).
reservationEnd	adbreakId: string, time: number	- Supported UVE version 0.8 and above. - Sent upon completion of an ad break (back to main content) - it is NOT sent (per previously agreed contract) if user does trickplay or seek to abort ad playback
placementStart	adId: string, time: number	- Supported UVE version 0.8 and above. - This is sent in real time when injecting first frame of a new ad on content->ad or ad->ad transition. Should be accurate compared to onscreen frames.

placementEnd	adId: string, time: number	- Supported UVE version 0.8 and above. - This is sent in real time after passively playing to end of an ad - it is NOT sent (per previously agreed contract) if user does trickplay or seek to abort ad playback.
placementProgress	adId: string, time: number	- Supported UVE version 0.8 and above. - Sent periodically while ad is being played out, giving an estimate percentage-watched metric. It's interpolated based on elapsed time, and should repeat same value if paused.
placementError	adId: string, time: number, error: number	- Supported UVE version 0.8 and above. - Generated only for exception while attempting to play out ad content.

addEventListener(name, handler)

Name	Type	Description
name	String	Event Name
handler	Function	Callback for processing event.

removeEventListener(name, handler)

Name	Type	Description
Name	String	Event Name
handler	Function	Callback for processing event.

CDAI Mechanism#1 – Engine Managed CDAI

Supported for DASH Linear, working with period structure and SCTE35 markers, with optional replacement for like-amount of content.

setSubscribedTags(tagNames)

- Supported UVE version 0.8 and above.
- Subscribe to specific tags / metadata in manifest

Name	Type	Description
tagNames	String Array	List of tag names of interest. Examples: #EXT-X-IDENTITY-ADS #EXT-X-MESSAGE-REF #EXT-X-CUE #EXT-X-ASSET-ID #EXT-X-TRICKMODE-RESTRICTION #EXT-X-CONTENT-IDENTIFIER

setAlternateContent(reservationObject, promiseCallback)

- Supported UVE version 0.8 and above.

Name	Type	Description

reservationObject	Object	reservationObject provides context for alternate content to be played out at ad opportunities.
		
		`{`
		`"reservationId": "1234", // period id from DASH manifest`
		`"reservationBehavior": number,`
		`"placementRequest":`
		`{`
		`// uuid generated to identify this placement`
		`"id": string,`

		// position at which placement will begin playback on the main timeline
		"pts": number,
		
		"url": "",
		},
		}

promiseCallback	Function	Signals success/failure while retrieving ad manifest and preparing for playback.

notifyReservationCompletion(reservationId, time)

- Supported UVE version 0.8 and above.
- Notify video engine when all ad placements for a particular reservation have been set via setAlternateContent.

Name	Type	Description
reservationId	String	
Time	Number	

CDAI Mechanism#2 – "Player Prebuffering" Feature

Can be leveraged for quick stream transitions. Suitable for preroll, and midroll insertions. No limitations with respect to content type – can transition between DASH and HLS.

detach()

- Supported UVE version 0.9 and above.
- Optional API that can be used to quickly stop playback of active stream before transitioning to 2nd prebuffered stream.

Example use of detach and buffering:

```
var player = new AAMPMediaPlayer();
player.load( "http://test.com/content.m3u8" ); // begin streaming main content
…
var adPlayer = new AAMPMediaPlayer(); // create background player
adPlayer.load( "http://test.com/ad.m3u8", false ); // preroll
…
player.detach(); // stop playback of active player
adPlayer.play(); // activate background player (fast transition)
player.stop(); // release remaining resources for initial player instance
```

Example of midroll Ad insertions and resume main content playback:

Main content (0 – 180 Sec)	AD1 (0 -40 Sec)	AD2 (0 – 30 Sec)	Main Content (180 – 600 Sec)

Main Content (0 – 180 sec)	**create foreground player and start streaming of main content**
	var player = new AAMPMediaPlayer();
	player.load("http://test.com/content.mpd");
	create background player and preload AD1
	var adPlayer1 = new AAMPMediaPlayer();
	adPlayer1.load("http://test.com/ad1.mpd", false);

AD1 (0 – 40 sec)	**time of AD1 start, stop active player and activate background player for AD1**
	var position = Player. getCurrentPosition() // get current playback position
	player.detach();
	adPlayer1.play();
	player.stop();
	preload AD2 in background player
	var adPlayer2 = new AAMPMediaPlayer();
	adPlayer2.load("http://test.com/ad2.mpd", false);

AD2 (0 – 30 sec)	**EOS of AD1, stop active player and activate background player for AD2**
	adPlayer1.detach();
	adPlayer2.play();
	adPlayer1.stop();
	preload Main content in background and set last playback position
	var player = new AAMPMediaPlayer();
	player. Seek (position)
	player.load("http://test.com/content.mpd", false);

Main Content (180 – 600 sec)	**EOS of AD2, stop active player and activate background player for main content**
	adPlayer2.detach();
	player.play();
	adPlayer2.stop();

CONFIGURATION

initConfig(config)

Configuration options are passed to AAMP using the UVE initConfig method. This allows the application override default configuration used by AAMP player to give more control over player behavior. Parameter is a JSON Object with one or more attribute/value pairs as follows:

Property	Type	Default Value	Description
initialBitrate	Number	2500000	max initial bitrate (bps)
initialBitrate4K	Number	13000000	max initial bitrate for 4k video playback (bps)
Offset	Number (s)	0	start position offset in seconds(same as seek() method)
networkTimeout	Number (s)	10	network request timeout for fragment/playlist/manifest downloads (in seconds)
manifestTimeout	Number (s)	10	Manifest download timeout; overrides networkTimeout if both present; available starting with version 0.8 . Applied to Main manifest in HLS and DASH manifest download. (in seconds)
playlistTimeout	Number (s)	10	HLS playlist download timeout; overrides networkTimeout if both present; available starting with version 1.0 (in seconds)
downloadBuffer	Number	3	max amount of time to download ahead of playhead (fragments) example: - with a downloadBuffer of 3 (default) there will be 3 fragments (typically 2s each) of video or audio harvested and buffered in advance, in additional to internal playback buffering
minBitrate	Number	-	Optional profile clamping (in bps)
maxBitrate	Number	-	Optional profile clamping (in bps)
preferredAudioLanguage	String	en	ISO-639 audio language preference; for more than one language, provide comma delimited list from highest to lowest priority: '<HIGHEST>,<...>,<LOWEST>'

timeShiftBufferLength	Number	-	(not supported, for future)
stereoOnly	Boolean	False	Optional forcing of playback to only select stereo audio track available starting with version 0.8
liveOffset	Number (s)	15	Allows override default/stream-defined distance from live point for live stream playback (in seconds)
asyncTune	Boolean	False	Return control to JS immediately when tuning with load() method (not supported)
bulkTimedMetadata	Boolean	False	Send timed metadata using single stringified JSON array instead of individual events available starting with version 0.8
networkProxy	String	-	Network proxy to use (Format <SCHEME>://<PROXY IP:PROXY PORT>)
licenseProxy	String	-	Network proxy to use for license requests (Format same as network proxy)
downloadStallTimeout	Number (s)	-	Optional optimization - Allow fast-failure for class of curl-detectable mid-download stalls (in seconds)
downloadStartTimeout	Number (s)	-	Optional optimization - Allow fast-failure for class of curl-detectable stall at start of download (in seconds)
preferredSubtitleLanguage	String	en	ISO-639 language code used with VTT OOB captions
parallelPlaylistDownload	Boolean	True	Optional optimization – download audio and video playlists in parallel for HLS; available starting with version 0.8
parallelPlaylistRefresh	Boolean	True	Optionally disable audio video playlist parallel download for linear (only for HLS)
useAverageBandwidth	Boolean	False	Optional Average bandwidth for ABR switching (version 1.0)
preCachePlaylistTime	Number (s)	-	Optionally enable PreCaching of Playlist and TimeWindow for Cache(minutes) (version 1.0)
progressReportingInterval	Number (s)	1	Optionally change Progress Report Interval (in seconds)
useRetuneForUnpairedDiscontinuity	Boolean	True	Optional unpaired discontinuity retune config (version 1.0)
drmDecryptFailThreshold	Number	10	Maximum number of fragment decrypt failures before reporting playback error (version 1.0)
initialBuffer	Number	-	Optional pre-tune buffering (in seconds) before playback start (version 2.4)
useMatchingBaseUrl	Boolean	False	use DASH main manifest hostname to select from multiple base urls in DASH (when present). By default, will always

			choose first (version 2.4)
initFragmentRetryCount	Number	1	Maximum number of retries for MP4 header fragment download failures (version 2.4)
nativeCCRendering	Boolean	False	Use native ClosedCaption support in AAMP (version 2.6)
langCodePreference	Number	0	Set the preferred format for language codes in other events/APIs (version 2.6) NO_LANGCODE_PREFERENCE = 0, 3_CHAR_BIBLIOGRAPHIC_LANGCODE = 1, 3_CHAR_TERMINOLOGY_LANGCODE = 2, 2_CHAR_LANGCODE = 3
descriptiveTrackName	Boolean	False	Use descriptive audio track naming format which is a combination of <lang>-<role> (version 2.6)
authToken	String	-	Optional field to set AuthService token for license acquisition(version 2.7)
useRetuneForGstInternalError	Boolean	True	Optional Gstreamer error retune config (version 2.7)
reportVideoPTS	Boolean	False	Optional field to enable Video PTS reporting along with progressReport (version 2.10)
propagateUriParameters	Boolean	True	Optional field to disable propagating URI parameters from Main manifest to segment downloads

setDRMConfig(config)

DRM configuration options are passed to AAMP using the setDRMConfig method. Parameter is JSON object with pairs of protectionScheme: licenseServerUrl pairs, along with preferredKeySystem specifying a preferred protectionScheme.

Property	Type	Description
com.microsoft.playready	String	License server endpoint to use with PlayReady DRM. Example: http://test.playready.microsoft.com/service/rightsmanager.asmx
com.widevine.alpha	String	License server endpoint to use with Widevine DRM. Example: https://widevine-proxy.appspot.com/proxy
preferredKeysystem	String	Used to disambiguate which DRM type to use, when manifest advertises multiple supported DRM systems. Example: com.widevine.alpha

Universal Video Engine Player Errors

Error code	Code	Error String
AAMP_TUNE_INIT_FAILED	10	AAMP: init failed
AAMP_TUNE_INIT_FAILED_MANIFEST_DNLD_ERROR	10	AAMP: init failed (unable to download manifest)
AAMP_TUNE_INIT_FAILED_MANIFEST_CONTENT_ERROR	10	AAMP: init failed (manifest missing tracks)
AAMP_TUNE_INIT_FAILED_MANIFEST_PARSE_ERROR	10	AAMP: init failed (corrupt/invalid manifest)
AAMP_TUNE_INIT_FAILED_TRACK_SYNC_ERROR	10	AAMP: init failed (unsynchronized tracks)
AAMP_TUNE_MANIFEST_REQ_FAILED	10	AAMP: Manifest Download failed Playlist refresh failed
AAMP_TUNE_INIT_FAILED_PLAYLIST_VIDEO_DNLD_ERROR	10	AAMP: init failed (unable to download video playlist)
AAMP_TUNE_INIT_FAILED_PLAYLIST_AUDIO_DNLD_ERROR	10	AAMP: init failed (unable to download audio playlist)
AAMP_TUNE_FRAGMENT_DOWNLOAD_FAILURE	10	AAMP: fragment download failures
AAMP_TUNE_INIT_FRAGMENT_DOWNLOAD_FAILURE	10	AAMP: init fragment download failed
AAMP_TUNE_INVALID_MANIFEST_FAILURE	10	AAMP: Invalid Manifest, parse failed
AAMP_TUNE_MP4_INIT_FRAGMENT_MISSING	10	AAMP: init fragments missing in playlist

AAMP_TUNE_CONTENT_NOT_FOUND	20	AAMP: Resource was not found at the URL(HTTP 404)
AAMP_TUNE_AUTHORISATION_FAILURE	40	AAMP: Authorization failure
AAMP_TUNE_UNTRACKED_DRM_ERROR	50	AAMP: DRM error untracked error
AAMP_TUNE_DRM_INIT_FAILED	50	AAMP: DRM Initialization Failed
AAMP_TUNE_DRM_DATA_BIND_FAILED	50	AAMP: InitData-DRM Binding Failed
AAMP_TUNE_DRM_SESSIONID_EMPTY	50	AAMP: DRM Session ID Empty
AAMP_TUNE_DRM_CHALLENGE_FAILED	50	AAMP: DRM License Challenge Generation Failed
AAMP_TUNE_LICENCE_TIMEOUT	50	AAMP: DRM License Request Timed out
AAMP_TUNE_LICENCE_REQUEST_FAILED	50	AAMP: DRM License Request Failed
AAMP_TUNE_INVALID_DRM_KEY	50	AAMP: Invalid Key Error, from DRM
AAMP_TUNE_UNSUPPORTED_STREAM_TYPE	50	AAMP: Unsupported Stream Type Unable to determine stream type for DRM Init
AAMP_TUNE_UNSUPPORTED_AUDIO_TYPE	50	AAMP: No supported Audio Types in Manifest
AAMP_TUNE_FAILED_TO_GET_KEYID	50	AAMP: Failed to parse key id from PSSH
AAMP_TUNE_FAILED_TO_GET_ACCESS_TOKEN	50	AAMP: Failed to get access token from Auth Service
AAMP_TUNE_CORRUPT_DRM_METADATA	50	AAMP: DRM failure due to Bad DRMMetadata in stream
AAMP_TUNE_DRM_DECRYPT_FAILED	50	AAMP: DRM Decryption Failed for Fragments
AAMP_TUNE_DRM_KEY_UPDATE_FAILED	50	AAMP: Failed to process DRM key

AAMP_TUNE_CORRUPT_DRM_DATA	51	AAMP: DRM failure due to Corrupt DRM files
AAMP_TUNE_DEVICE_NOT_PROVISIONED	52	AAMP: Device not provisioned
AAMP_TUNE_HDCP_COMPLIANCE_ERROR	53	AAMP: HDCP Compliance Check Failure
AAMP_TUNE_GST_PIPELINE_ERROR	80	AAMP: Error from gstreamer pipeline
AAMP_TUNE_FAILED_PTS_ERROR	80	AAMP: Playback failed due to PTS error
AAMP_TUNE_PLAYBACK_STALLED	7600	AAMP: Playback was stalled due to lack of new fragments
AAMP_TUNE_FAILURE_UNKNOWN	100	AAMP: Unknown Failure

Inband Closed Caption Management

To use inband closed captions, first register an event listener to discover decoder handle:

```
player.addEventListener("decoderAvailable", decoderHandleAvailable);
```

Along with corresponding event handler to publish the decoder handle to CC subsystem as follows:

```
function decoderHandleAvailable(event) {
   console.log("decoderHandleAvailable " + event.decoderHandle);
   XREReceiver.onEvent("onDecoderAvailable", { decoderHandle: event.decoderHandle });
}
```

Toggle CC display on or off at runtime:

```
XREReceiver.onEvent("onClosedCaptions", { enable: true });
XREReceiver.onEvent("onClosedCaptions", { enable: false });
```

Set CC track at runtime:

```
XREReceiver.onEvent("onClosedCaptions", { setTrack: trackID });
```

Set CC style options at runtime:

```
XREReceiver.onEvent("onClosedCaptions", { setOptions: defaultCCOptions});
```

defaultCCOptions is a JSON object of various style options and its values

When closing stream, detach decoder handle:

```
XREReceiver.onEvent("onDecoderAvailable", { decoderHandle: null });
```

Environments without the XREReceiver JS object may exist in future. Applications may use alternate CC rendering methods to avoid dependency on XREReceiver object.

To use, turn on nativeCCRendering init configuration value to true as follows:

```
player.initConfig( { nativeCCRendering: true } );
```

Toggle CC display on or off at runtime:

```
player.setClosedCaptionStatus(true);
```

```
player.setClosedCaptionStatus(false);
```

Get/Set CC track at runtime:

```
player.getTextTrack();
player.setTextTrack(trackIndex);
```

Get/Set CC style options at runtime:

```
player.getTextStyleOptions();
player.setTextStyleOptions(options);
```

options in a JSON formatted string of style options and its values.

ATSC – Unified Video Engine Features

Support for ATSC-UVE is included from 2.10 version.
A subset of UVE APIs and Events are available when using UVE JS APIs for ATSC playback

API Methods

APIs Supported	Description
load	URI of the Media to be played by the Video Engine. Optional 2nd parameter.
play	Start Playback / Resume playback.
stop	Stop playback and free resources
getAudioTrack	Get the index of the currently selected Audio track
setAudioTrack	Set the index of the Audio track to be selected.
setAudioLanguage	Set the language of the Audio track to be selected
setVideoRect	Set display video rectangle coordinates. Default configuration (0,0,1280,720)
getAvailableAudioTracks	Returns the available audio tracks information in JSON formatted list. Subset of parameters returned • name • language • codec Example: {"name": "English (AC3)","language":"eng","codec":"AC3"}

Events Supported

Events Supported	Value	Description
playbackStarted	1	Tune Success
playbackStateChanged	14	Event when player state changes. AAMP States received with playbackStateChanged : "idle":0, "initializing":1, "initialized":2, "preparing":3, "prepared":4, "buffering":5, "playing":8`

InitConfig

Property	Type	Default Value	Description
preferredAudioLanguage	String	en	ISO-639 audio language preference; for more than one language, provide comma delimited list from highest to lowest priority: '<HIGHEST>,<...>,<LOWEST>'