AAMP UVE – API Documentation

V4.4

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 • 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	3	Oct 2020 Release update.

		• Updated getAvailableAudioTracks / getAvailableTextTracks API: • setAudioLanguage Configuration: • propagateUriParameters • reportVideoPTS ATSC – UVE Features Added .
10	3.1	Jan 2021 Release update. ATSC New APIs / Events API: • getAvailableThumbnailTracks • setThumbnailTrack • getThumbnail Configuration: • sslVerifyPeer • persistBitrateOverSeek • setLicenseCaching • maxPlaylistCacheSize • enableSeekableRange
11	3.2	Mar 2021 Release update API • setPreferredAudioLanguage • setAudioTrack Configuration: • livePauseBehavior • limitResolution
12	3.3	May 2021 Release update Configuration: • useAbsoluteTimeline • asyncTune Events : • Updated bitrateChanged for ATSC

13	3.4	Events : • audioTracksChanged • textTracksChanged • seeked • vttCueDataListener • id3Metadata
14	3.5	Aug 2021 Release update API • load (updated) • setPreferredAudioLanguage (updated) • getAvailableAudioTracks (updated) • getAvailableTextTracks (updated) • downloadBuffer default value(updated) Events : • id3Metadata
15	3.6	Sept 2021 Release update Configuration • disable4K • sharedSSL • preferredAudioRendition • preferredAudioCodec Events: • mediaMetadata (updated)
16	4.1	Jan 2022 Release update API • subscribeResponseHeaders Configuration • supportTLS • maxInitFragCachePerTrack • fragmentDownloadFailThreshold • tsbInterruptHandling • sslVerifyPeer (updated)

		Events: • AAMP_TUNE_UNSUPPORTED_AUDIO_TYPE (updated error code) • AAMP_TUNE_UNSUPPORTED_STREAM_TYPE (updated error code) • AAMP_EVENT_CONTENT_GAP • AAMP_EVENT_HTTP_RESPONSE_HEADER
17	4.2	Feb 2022 Release update API • getAudioTrackInfo • getPreferredAudioProperties Configuration • Updated asyncTune default state to True • useSecManager Events: • AAMP_EVENT_WATERMARK_SESSION_UPDATE
18	4.3	Mar 2022 Release update API • getPlaybackStatistics Configuration • customLicenseData • Updated asyncTune default state to False
19	4.4	Apr 2022 Release update Support for AC4 Audio API • getAvailableVideoTracks • setVideoTracks Configuration • disableAC4 • asyncTune default state to True

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
AAMP.version	number	Global variable for applications to get UVE API version without creating a player instance. Value will be same as player.version.

METHODS:

load(uri, autoplay, tuneParams)

- 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.
tuneParams	Object	optional 3rd parameter The tuneParams Object includes four elements contentType, traceId, isInitialAttempt and isFinalAttempt. Details provided in below table

Name	Type	Description
contentType	String	Content Type of the asset taken for playback. Eg: CDVR, VOD, LINEAR_TV, IVOD, EAS, PPV, OTT, OTA, HDMI_IN, COMPOSITE_IN, SLE
traceId	String	Trace ID which is unique for a tune.
isInitialAttempt	Boolean	Flag indicates if it's the first tune initiated, tune is neither a retry nor a rollback.
isFinalAttempt	Boolean	Flag indicates if the current tune is the final retry attempt, count has reached the maximum tune retry limit.

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.

release()

- Release memory of native player.
- If this API is not called , then garbage collector takes care of memory release

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

State Name	Value	Semantics	Remarks
			and metadata processing.
	3	eSTATE_PREPARING	Create internal resources required 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
accessibilityType	String	Accessibility value for descriptive, visually impaired signaling e.g: description, captions

```
Example:

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

Reference:

<AdaptationSet id="3" contentType="audio" segmentAlignment="true" bitstreamSwitching="true" lang="ger">
<Role schemeIdUri="urn:mpeg:dash:role:2011" value="german"/>
<Accessibility schemeIdUri="urn:mpeg:dash:role:2011" value="description" />
<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.
accessibilityType	String	Accessibility value for descriptive, visually impaired signaling e.g: description, captions

Example:

```
{
    "name": "caption_en",
    "type": "SUBTITLES",
    "language":     "en",
```

```
          "accessibiltyType": "description",

          "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>
        <Accessibility schemeIdUri="urn:mpeg:dash:role:2011" value="captions" />
      </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.

getAudioTrackInfo()

- Supported UVE version 4.2 and above.
- Returns the list of audio tracks available in the stream , in JSON format .

getPreferredAudioProperties ()

- Supported UVE version 4.2 and above.
- Returns the list of preferred language, codecs , rendition and audio type selected by user , in JSON format.

getPlaybackStatistics ()

- Supported UVE version 4.3 and above.
- Returns the playback statistics in JSON format during playback.
- Refer appendix for full JSON format

Example:

{"timeToTopProfile":42,"timeInTopProfile":1096,"duration":1359,"profileStepDown_Network":4,"displayWidth":3840,"displayHeight":2160,"profileCappingPresent":0,"mediaType":"DASH",
“playbackMode":"VOD","totalError":0,"numOfGaps":0,"languageSupported":\{"audio1":"en"}

,"main":{"profiles":{"0":{"manifestStat":{"latencyReport":{"timeWindow_0":1}

,"sessionSummary":{"200":1},"info":{"DownloadTimeMs":287,"ParseTimeMs":6,"PeriodCount":1,"Size":20277}}}}},"video":{"profiles":{"1807164":{"fragmentStat":{"media":{"latencyReport":{"timeWindow_0":3},"sessionSummary":{"200":3}},

"init":{"latencyReport":{"timeWindow_0":1},"sessionSummary":{"200":1}}},"width":960,"height":540},"4532710":{"fragmentStat":{"media":{"latencyReport":{"timeWindow_0":128},"sessionSummary":{"200":128}},

"init":{"latencyReport":{"timeWindow_0":1},"sessionSummary":{"200":1}}},"width":1280,"height":720},"7518491":{"fragmentStat":{"media":{"latencyReport":{"timeWindow_0":548}

getAvailableVideoTracks ()

- Supported UVE version 4.4 and above.
- Returns the video profile information from the manifest.
- Refer appendix for full JSON format

setVideoTracks ()

- Supported UVE version 4.4 and above.
- This API will set the Video track(s) to select for playback .
- This will override the TV resolution based profile selection and MinBitRate/MaxBitRate based profiles

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

setAudioTrack(trackDescriptorObj)

- Supported UVE version 3.2 and above.
- Set the audio track by language and rendition from available audio track list.
- "language" match always takes precedence over "rendition" match.
- While playing passively to new periods with different track order/availability, or when tuning to new locator, heuristic for track selection is automatically re-applied.
- Note that for now, "best" codec (ATMOS > DD+ > Stereo) is always selected, subject to filtering configuration.

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

```
Example:
var trackDescriptorObject =
{
      "language": "ger",
      "rendition": "commentary"
}
playerInstance.setAudioTrack( trackDescriptorObject );
```

setPreferredAudioLanguage(languages, rendition, accessibility)

- Supported UVE version 3.2 and above.
- Set the audio track preference by languages, rendition and accessibility
- This is functionally equivalent to passing a trackDescriptorObject to setAudioTrack above.
- May be called pre-tune or post tune.

Name	Type	Description
languages	String	ISO-639 audio language preference; for more than one language, provide comma delimited list from highest to lowest priority: '<HIGHEST>,<...>,<LOWEST>'
rendition	String	Optional preferred rendition for automatic audio selection.
accessibility	String	Optional preferred accessibility type for descriptive audio.

setAudioLanguage(language)

- Supported UVE version 3.0 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

getAvailableThumbnailTracks ( )

- Returns json array of each thumbnail track's metadata

Name	Type	Description
Resolution	String	String indicating the width x height of the thumbnail images.
Bandwidth	String	Decimal-Integer encoding - bits per second. Represents bit rate of the thumbnail track.

Example:

```
[{
        "RESOLUTION":   "416x234",
        "BANDWIDTH":    71416
  }, {
        "RESOLUTION":   "336x189",
        "BANDWIDTH":    52375
  }, {
        "RESOLUTION":   "224x126",
```

```
            "BANDWIDTH":    27413
        }]
```

setThumbnailTrack(index)

- Set the desired thumbnail track from the list of available thumbnail track metadata.
- Returns Boolean value true or false to indicate Success or Failure configuring the thumbnail track.

Name	Type	Description
Index	Number	Index value based on the available thumbnail tracks.

getThumbnail(startPosition, endPosition)

- Get the thumbnail data for the time range “startPosition” till “endPosition”.

Name	Type	Description
startPosition	Number	Start value from which the thumbnail data is fetched.
endPosition	Number	End value till which the thumbnail data is fetched.
baseUrl	String	The base url which is appended to tile url to fetch the required thumbnail image.
raw_w	String	Original width of the thumbnail sprite sheet.
raw_h	String	Original height of the thumbnail sprite sheet.
width	String	Width of each thumbnail tile present in the sprite sheet.
height	String	Height of each thumbnail tile present in the sprite sheet.
tile	String	JSON array of multiple thumbnail tile information.
url	String	Url for each tile, which is appended with base url to form complete url.
t	String	Presentation time for each tile.
d	String	Duration value of each tile.
x	String	X co-ordinate position to locate the tile from sprite sheet.
y	String	Y co-ordinate position to locate the tile from sprite sheet.

Example:

```
{
  "baseUrl":      "https://g004-c-13a10c-
peacockvodstg.s.llnwi.net/pub/global/aOb/kIc/PCK_1604349987778_01/cmaf_thumbtest_segtime_d/mpeg_2sec/
images/416x234/",
  "raw_w":        3744,
  "raw_h":        3978,
  "width":        416,
  "height":       234,
  "tile": [{
                  "url":  "pckimage-1.jpg",
                  "t":    328.0,
                  "d":    2,
                  "x":    832,
                  "y":    234
          }]
}
```

subscribeResponseHeaders(headerNames)

- Supported UVE version 4.1 and above.
- Subscribe http response headers from manifest download

Name	Type	Description
headerNames	String Array	List of tag names of interest. Examples: C-XServerSignature

		X-Powered-By X-MoneyTrace

EVENTS

Event Number	Event Name	Event Payload	Description
1	playbackStarted		- Supported UVE version 0.7 and above. - fired when playback starts
2	playbackFailed	shouldRetry: boolean, code: number, description: string	- Supported UVE version 0.7 and above. - fired when an error occurs
3	playbackSpeedChanged	speed: number, reason: string	- Supported UVE version 0.7 and above.
4	playbackCompleted		- Supported UVE version 0.7 and above. - fired when there is nothing left to play
5	playlistIndexed		- fired after manifest / playlist parsing completed
6	playbackProgressUpdate	durationMiliseconds: number, positionMiliseconds: number, playbackSpeed: number, startMiliseconds: 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)

		endMiliseconds: number, currentPTS: number, videoBufferedMiliseconds : number	
7	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
8	jsEvent		Generic event for jsbinding . to be deprecated
9	mediaMetadata	durationMiliseconds: number, languages: string[], bitrates: number[], playbackSpeeds: number[], width: number, height: number, hasDrm: boolean isLive: boolean programStartTime: DRM: string[]	- 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

10	enteringLive		- Supported UVE version 0.7 and above. - fired when entering live point of a live playlist during/after a seek/trickplay operation
11	bitrateChanged	time: number, bitRate: number, description: string, width: number, height: number, framerate: number position: number cappedProfile:bool displayWidth:number displayHeight:number	- Supported UVE version 0.7 and above. - fired when video profile is switched by ABR with the metadata associated with newly selected profile.
12	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
13	bulkTimedMetadata	content: string,	Combine all timedMetadata and fire single event if bulkTimedMetadata is enabled
14	playbackStateChanged	state: number	- Supported UVE version 0.7 and above. - fired as state changes across play/pause seek/not-seek quadruplet

15	speedsChanged	playbackSpeeds: number[]	- Supported UVE version 0.7 and above. - fired when supported playback speeds changes (based on iframe availability)
16	seeked	Position: number	fired when Seek is triggered with a position
17	tuneProfiling	microdata:string	Tune profiling data
18	bufferingChanged	buffering: bool	- Supported UVE version 0.8 and above. - fired when AAMP encounters buffering mid-playback, buffering flag indicates buffer status FALSE -> No buffer for playback(Underflow) TRUE -> Buffer available for playback
19	durationChanged		To be deprecated
20	audioTracksChanged		fired when Audio track is changed during playback
21	textTracksChanged		fired when Text track is changed during playback
22	contentBreaksChanged		To be deprecated
23	contentStarted		To be deprecated
24	contentCompleted		To be deprecated
25	drmMetadata	code: number,	- Supported UVE version 0.7 and above.

		description: string	- fired when there is a change in DRM metadata (especially expiration of DRM auth data)
26	anomalyReport	severity: string description:string	fired for any anomaly during playback
27	vttCueDataListener	start : number duration: number text:string	- This event is fired for VTT cue parsed from the WebVTT playlist
28	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.
29	reservationStart	adbreakId: string, time: number	- Supported UVE version 0.8 and above. - Sent upon playback into an ad break (one or more ads).
30	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
31	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.
32	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.
33	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.
34	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.
35	metricsData	metricType:string traceID:string metricData:string	Playback data after video end.
36	id3Metadata	schemeIdUri : string value : string timescale : number presentationTime : number eventDuration : number id : number timestampOffset : number data : array	This event is fired when ID3Metadata is parsed from the stream playlist

		length: number	
37	drmMessage	data:string	Drm challenge data after individualization
38	blocked	reason:string	Event with reason for video blocked. *used only in ATSC playback.
39	contentGap	time:number duration:number	Event with Content gap information in TSB due to network interruption
40	httpResponseHeader	header:string response:string	http header fields received in manifest download
41	watermarkSessionUpdate	sessionHandle:string status:string system:string	Watermarking session information

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	4	max amount of time to download ahead of playhead (fragments) example: - with a downloadBuffer of 4 (default) there will be 4 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)
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 3.0)
propagateUriParameters	Boolean	True	Optional field to disable propagating URI parameters from Main manifest to segment downloads
enableSeekableRange	Boolean	False	Optional field to enable reporting of seekable range for linear scrubbing
maxPlaylistCacheSize	Number	0	Optional field to configure maximum cache size in Kbytes to store different profile HLS VOD playlist
setLicenseCaching	Boolean	True	Optional field to disable License Caching in player . By default 3 DRM Sessions are Cached .
persistBitrateOverSeek	Boolean	False	To enable AAMP persisting video profile during Seek/Trickplay/Audio switching operation
sslVerifyPeer	Boolean	True	Optional field to enable/disable SSL peer verification .Default enabled
livePauseBehavior	Number	0	Optional field to configure player live pause behavior on linear streams when live window touches eldest position.

			Options: 0 – Autoplay immediate 1 – Live immediate 2 – Autoplay defer 3 – Live defer Default – Autoplay immediate
limitResolution	Boolean	False	Optional field to set maximum video profile resolution based on TV display resolution setting . Default Off.
asyncTune	Boolean	True	Optional field to enable asynchronous player API processing. Application / UI caller threads returned immediately without any processing delays.
useAbsoluteTimeline	Boolean	False	Optional field to enable progress reporting based on Availability Start Time of stream (DASH Only)
sharedSSL	Boolean	True	Optional field to disable sharing SSL context for all download sessions, across manifest, playlist and segments .
disable4K	Boolean	False	Optional field to disable 4K profile playback and restrict only to non-4k video profiles.
preferredAudioRendition	String		Optional field to set preferred Audio rendition setting DASH : caption,subtitle,main HLS : GROUP-ID
preferredAudioCodec	String		Optional field to set preferred Audio Codec 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
tsbInterruptHandling	Boolean	False	Optional field to enable support for Network interruption handling with TSB. Network failures will be ignored and TSB will continue building
supportTLS	Number	6	Default set to CURL_SSLVERSION_TLSv1_2 (value of 6 , uses CURLOPT_SSLVERSION values)
maxInitFragCachePerTrack	Number	5	Number of init header file cached per player instance per track type. Use cached data instead of network re-download
fragmentDownloadFailThreshold	Number	10	Maximum number of fragment download failures before reporting playback error
useSecManager	Boolean	True	Optional field to enable /disable usage of SecManager for

			Watermarking functionality
disableEC3	Boolean	False	Optional field to disable selection of EC3/AC3 audio track
disableATMOS	Boolean	False	Optional field to disable selection of ATMOS audio track
disableAC4	Boolean	False	Optional field to disable selection of AC4 audio track

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
customLicenseData	String	Optional field to provide Custom data for license request

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	60	AAMP: Unsupported Stream Type Unable to determine stream type for DRM Init
AAMP_TUNE_UNSUPPORTED_AUDIO_TYPE	60	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 3.0 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.
	Examples for new URLs Supported:
	`live:///75 – ATSC Channel`
	`hdmiin://localhost/deviceid/0 – HDMI Input`
	`cvbsin://localhost/deviceid/0 – Composite Input`
	`tune://atsc?frequency=5700000&serviceid=3 – Direct tune to ATSC Channel`

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.

setAudioTrack	Set the Audio track to be selected by Language and Rendition.
	JSON formatted argument.
	• language
	• rendition
	`Example:`
	`{"language":"ger","rendition":"commentary"}`

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"}
setClosedCaptionStatus	Set the Closed Caption rendering to on/off.
getAvailableTextTracks	Returns the available text track (CC) information in JSON formatted list. Subset of parameters returned • name • type • language • instreamId Example: [{"name":"English (Closed Caption)","type":"CLOSED-CAPTIONS","language":"eng",”instreamId":"CC1"}, {"name":"Spanish (Closed Caption)","type":"CLOSED-CAPTIONS","language":"spa","instreamId":"CC2"}, {"name":"English (Closed Caption)","type":"CLOSED-CAPTIONS","language":"eng","instreamId":"SERVICE1"}, {"name":"Spanish (Closed Caption)","type":"CLOSED-CAPTIONS","language":"spa","instreamId":"SERVICE2"}]
getTextTrack	Get the Index of the currently selected Text track
setTextTrack	Set the Index of the Text track to be selected.
getTextStyleOptions	Returns the JSON formatted string of current ClosedCaption style options and values.
setTextStyleOptions	Set the ClosedCaption style options to be used for rendering.

New Set of APIs added for ATSC Parental Control Settings

disableContentRestrictions (until)

- Temporarily disable content restriction based on the control input provided by the ‘until’ parameter.
- Can be used for unlocking a locked channel (Channel locked due to Restrictions set)

Name	Type	Description
until {"time": < seconds>} Or {"programChange":true};	Json Object	Provides control for automatic re-locking conditions. ▪ If 'time' is set, the seconds will be considered as relative to current time until which the program will be unlocked. ▪ If 'programChange' is set, the program will be unlocked, but re-enable restriction handling on next program boundary ▪ If neither specified, parental control locking will be disabled until settop reboot, or explicit call to enableContentRestrictions(). ▪ If both specified, parental control locking will be re-enabled depending on which condition occurs first.

enableContentRestrictions ()

- To re-enable parental control locks based on restrictions

Events Supported

Events Supported	Value	Description
playbackStarted	1	Tune Success [OTA , HDMIIN,COMPOSITE IN]
playbackStateChanged	14	Event when player state changes. Valid AAMP States for ATSC OTA Playback: : "idle":0, "initializing":1, "initialized":2, "preparing":3, "prepared":4, "playing":8, "blocked":14 Valid AAMP States for HDMIIN : "playing":8,"stopped":10
blocked	38	Blocked event is generated when player status switches to eSTATE_BLOCKED. The Event payload is a string to describe the reason for the blocked state. Event Payload: Type : reason – string Reason for restriction Eg: "reason": (ATSC Playback) "STATUS|Low or No Signal" "Service Pin Locked" "STATUS|Unable to decode" If a program is Blocked due to Restrictions set by the Application, the 'blocked' event's reason will be "Service Pin Locked" "reason": (HDMIIN) "NO_SIGNAL" "UNSTABLE_SIGNAL" "NOT_SUPPORTED_SIGNAL"

bitrateChanged	11	Event notified when bitrate change happens. The event payload provides video stream info Will be notified after first tuned event for OTA and and after display settings change. Event Payload: time: number, bitRate: number, description: string, width: number, height: number, framerate: number, position: number(not used), cappedProfile:number (not used), displayWidth:number (not used), displayHeight:number (not used), progressive:bool, aspectRatioWidth:number, aspectRatioHeight:number

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>'
nativeCCRendering	Boolean	False	Use native Closed Caption support in AAMP

APPENDIX

getPlaybackStatistics

Description: Returns the playback statistics from beginning of playback till the time API is called.
JSON description:

```
{
        "mediaType": {
            type: string,
            description: Stream type like HLS, DASH etc
        },
        "playbackMode": {
            type: string,
            description: Playback modes like linear, vod etc
        },
        "liveLatency": {
            type: integer,
            description:  The time from current position to end of seek range,
                        this field would only be available for Linear playback
            unit: milliseconds
        },
        "totalError": {
            type: integer,
            description: Total number of download errors so far (as of now, but will include other
errors too in future)
                across profiles/manifests etc
        },
        "numOfGaps": {
            type: integer,
            description: total number of gaps between periods,
                        for Live this would be the number of gaps so far,
                        but for vod this would be the number of gaps across all periods,
                        this field would only be available for DASH
        },
        "timeInTopProfile": {
```

```
            type: integer,
            description: Duration of the media playback stayed in the top file profile from beginning
of the playback,
            unit: seconds
        },
        "timeToTopProfile": {
            type: integer,
            description: time to reach top profile first time after tune. Provided initial tune
bandwidth is not a top bandwidth
            unit: seconds
        },
        "duration": {
            type: integer,
            description: Duration of the playback so far,
            unit: seconds
        },
        "profileStepDown_Network": {
            type: integer,
            description: Number of profile step downs due to network issues,
        },
        "profileStepDown_Error": {
            type: integer,
            description: Number of profile step downs due to playback errors,
        },
        "displayWidth": {
            type: integer,
            description: display width,
        },
        "displayHeight": {
            type: integer,
            description: display height,
        },
        "profileCappingPresent": {
            type: integer,
            description: profile capping status,
        },
        "tsbAvailable": {
            type: integer,
            description: will be 1 if tsb is employed for the playback,
        },
        "languagesSupported": {
            type: object,
```

```
            description: lists the supported audio tracks. with name audio1, audio2 etc
             properties: {
                 "audio1": {
                    type: string,
                    description: language corresponds to audio1,
                }
            }
        },
        "main"/"audio1"/"audio2" etc/"video1"/"subtitle"/"iframe"/"ad_audio"/"ad_video"etc: {
            type: object,
            description: different track types ("main" to specify main manifest)
             properties: {
                "profiles": {
                    type: object,
                    description: for each profile in the given track type,
                     properties: {
                        "0/<profile number> etc": {
                            type: object,
                            description: details of a specific profile played, for the main/master
manifest, this would be "0",
                             properties: {
                                "manifestStat": {
                                    type: object,
                                    description: For HLS, here it describes the stats of playlist
manifest corresponds to each profile,
                                                 for DASH this field be only available for main manifest,
                                     properties: {
                                        "latencyReport": {
                                           type: object,
                                            description: field indicating the latency of the download,
                                             properties: {
                                                "timeWindow_0/timeWindow_1 etc": {
                                                    type: integer,
                                                    description: gives the number of downloads on an
item comes under 0th, 1th, 2nd bucket,
                                                                 with a bucket of duration 250ms
                                                }
                                            }
                                        },
                                        "sessionSummary": {
                                            type: object,
```

```
                description: Shows the session information so far
(400/404,206), number of various error
                             occurrences along with successful (200,206 etc)
cases
                 properties: {
                    "200": {
                        type: integer,
                        description: number of http 200 cases
                    }
                }
            },
            "info": {
                type: object,
                description: This field will only be present for for
manifests.
                    For linear DASH playback, this field will represent
the details of the last refreshed manifest
                 properties: {
                    "downloadTimeMs": {
                        type: integer,
                        description: Manifest download time,
                        unit: milliseconds
                    },
                    "parseTimeMs": {
                        type: integer,
                        description: Manifest parse time, this field will
only be available for main/master manifests,
                        unit: Milliseconds
                    },
                    "size": {
                        type: integer,
                        description: Downloaded manifest size in bytes,
                        unit: bytes
                    },
                    "periodCount": {
                        type: integer,
                        description: number of periods, only applicable
for DASH
                    }

                }
            }
```

```
                                    }
                                },
                                "fragmentStat": {
                                    type: object,
                                    description: Statistics of the downloaded fragments, would not be
applicable for profile "0" (as is is for manifest),
                                     properties: {
                                        "media": {
                                            type: object
                                            description: Stats of media (non-init) fragments,
                                             properties: {
                                                "latencyReport": {
                                                    //similar as in manifest stats
                                                },
                                                "sessionSummary": {
                                                    //similar as in manifest stats
                                                }
                                            }
                                        }
                                        "init": {
                                            type: object
                                            description: Stats of init fragments,
                                             properties: {
                                                "latencyReport": {
                                                    //similar as in media
                                                },
                                                "sessionSummary": {
                                                    //similar as in media
                                                }
                                            }
                                        },
                                        "lastFailedUrl": {
                                            type: string,
                                            description: url of the last failed fragment ( could be
normal fragment/ init  )
                                        },
                                        "width": {
                                            type: integer,
                                            description: video width specified for the particular
bitrate (ony for the track type video),
                                        },
                                        "height": {
```

```
                                            type: integer,
                                            description: video height specified for the particular
bitrate (ony for the track type video),
                                        }
                                    }
                                },
                                "licenseStat": {
                                    type: object,
                                    description: license related stats
                                    properties: {
                                        "totalClearToEncrypted": {
                                            type: integer,
                                            description: Total number of clear to encrypted content
switches
                                        },
                                        "totalEncryptedToClear": {
                                            type: string,
                                           description: Total number encrypted to clear content
switches
                                        }
                                    }
                                },

                            }
                        }
                    }
                }
            }
        },
        "version": {
            type: string,
            description: Version of this document
        },
        "creationTime": {
            type: string,
            description: UTC timestamp of this document creation
        }
}
```

getAvailableVideoTracks


```
AVAILABLE VIDEO TRACKS: [{
                "bandwidth":    5000000,
                "width":          1920,
                "height":       1080,
                "framerate":    25,
                 "codec":    "avc1.4d4028",
                 "enabled":  1
        }, {
                "bandwidth":    2800000,
                "width":          1280,
                "height":       720,
                "framerate":    25,
                 "codec":    "avc1.4d401f",
                 "enabled":  1
        }, {
               "bandwidth":    1400000,
                "width":          842,
                "height":       474,
                "framerate":    25,
                 "codec":       "avc1.4d401e",
                 "enabled":   1
        } ]
```