Electra Media Player

An overview of how to use the Electra Media Player.

Electra Media Player is a media player plugin for Unreal Engine, designed for both live streams and video on demand (VOD) streams using the [HTTP Live Streaming (HLS)](#) format. With this plugin, you can stream live performances and media into your Unreal project by providing the link to the source. In addition to streaming, Electra also supports `.mp4` progressive download and local playback of `.mp4` files.

Electra is supported on the following platforms:

- Windows 7 or later

- macOS
- iOS 13.0 or later
- Android Lollipop (5.1) or later
- Playstation 4
- Playstation 5
- Xbox One/S/X
- Xbox Series X
- Nintendo Switch

The maximum playable video resolution and frame rate depends on the platform. See the respective platform documentation for their specifications.
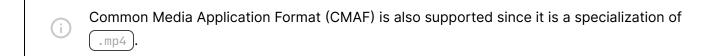
To learn how to use the Electra Media Player to stream videos in your projects, see Play a Video Stream.

Supported Media Formats

Electra Media Player is a streaming media player within Unreal Engine that supports HLS formatted media streams. See HTTP Live Streaming for more details.

Electra Media Player also supports playback of a single, multiplexed `.mp4` (ISO/IEC 14496-12) file if it meets the following specifications:

- Contains at most one video track.
- Contains only video and audio tracks.
- Formatted as **faststart**, which stores the metadata of the file—the **moov atom**—before the frames—the **mdat**.
- Accessed through the `file://` URI scheme only.

> (i) Common Media Application Format (CMAF) is also supported since it is a specialization of `.mp4`.

See the Media Framework Technical Reference for a full list of supported file formats.

Supported Codecs

Electra Media Player only supports media files and streams encoded with the following codecs:

Media Type	Codec
Video Codec	H.264 / AVC (ISO/IEC 14496-10)
Audio Codec	AAC (ISO/IEC 14496-3)

HTTP Live Streaming

Electra Media Player provides streaming functionality in Unreal projects by supporting HLS (RFC-8216). With HLS, the server fragments the media files into segments and creates an index file. This index file contains the list of the media segments and their location that the client reads and uses to format its requests for media playback.

HLS also supports variant streams for the media files with different bitrates so the client can minimize stalling in playback. The variant streams are defined in another index file, called the master playlist. For more details on the HLS architecture, see Apple's HLS documentation.

Variant Streams Restrictions

HLS permits variant streams to use different segment durations as well as different number of segments without a way to identify matching segments. To reduce client overhead, the Electra Media Player requires all video variants to be segmented identically. For VOD presentations, the Electra Media Player then switches variants based on the absolute index of the segment in the variant playlists.

> Live streaming presentations are handled differently since the playlists are constantly changing and contain a rolling window of a few media segments. Use the EXT-X-PROGRAM-DATE-TIME tag to identify matching segments in addition to segmenting media identically.

For optimal performance, Electra Media Player has the following restrictions for playlists and media segments:

- **Variant streams must be formatted as ISO/IEC-14496-12 or CMAF segments:** MPEG2 transport stream segments as per ISO/IEC-13818-1 are not supported.

- **Corresponding segments in variant streams must have the same duration at each segment index:** The video does not have to be segmented equally. For every variant stream *A* and *B*, the segment duration *dur* for segment *n* is the same:

  `dur(segment A(n)) == dur(segment B(n))`

  Different stream types do not need to have identical durations when segmented. Audio can use different segment durations than video segments.

- **Audio streams do not switch:** There must be only one audio stream specified in the master playlist.

- **Segment duration cannot be rounded to the nearest integer in the variant playlists:** Audio and video will segment differently because individual video images and audio sample blocks are not equal. The Electra Media Player synchronizes audio to video by locating the audio segment whose start time best matches the start time of the video segment and discarding any audio samples that are before the video start time. If the segment durations are not specified precisely in the variant playlist, the small error will accumulate and eventually result in selecting the wrong audio segment to start with, which would cause the audio to be out of sync with the video.

- **Audio groups are required in a master playlist:** Because the container format requires a single elementary stream, you must specify an audio group in the master playlist. The following is an example of an audio group media tag in the master playlist.

  `#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="group_audio",NAME="audio_0",DEFAULT=YES,URI="manifest_0.m3u8"`

  `#EXT-X-STREAM-INF:BANDWIDTH=8905600,RESOLUTION=1920x1080,CODECS="avc1.42c028,mp4a.40.2",AUDIO="group_audio"`

- `CODECS` **and** `RESOLUTION` **properties are required in** `EXT-X-STREAM-INF` **tags:** These properties are optional in RFC-8216 but required for Electra Media Player.

- **Use the** `FRAME-RATE` **property in** `EXT-X-STREAM-INF` **tags:** The `FRAME-RATE` property is highly recommended for media playback with Electra Media Player since it helps identify

and filter streams that cannot be played back on platforms that have issues with decoding more than 30 frames per second.

- **Rendition groups other than the required audio group are not supported:** Rendition groups are used for alternate content, such as different viewing angles. Electra Media Player only supports one rendition group, the audio group referenced by the video variant playlists.

- **Video segments must have an Instantaneous Decoder Refresh (IDR) frame as the first frame:** Playback will start on times only where an IDR frame exists in the video stream. This is equivalent to implicitly requiring the use of the `EXT-X-INDEPENDENT-SEGMENTS` tag, which itself is not required to be set but it is strongly recommended.

- **Frame accurate seeking is not supported:** The IDR frame nearest to the specified time where playback should begin will be located and playback starts from there.

- **Only AES-128 encryption is supported for streams with the HLS format:** SAMPLE-AES is not supported.

> (i) Media streams can contain an audio track without a video presentation. It is also possible to only stream a video with no audio.

Unsupported HLS Tags

Electra ignores the following tags in a master and variant playlist. These tags will not cause an error if they are present in the file unless they are required for correct playback.

- `EXT-X-DISCONTINUITY`
- `EXT-X-DISCONTINUITY-SEQUENCE`
- `EXT-X-DATERANGE`
- `EXT-X-I-FRAMES-ONLY`
- `EXT-X-I-FRAME-STREAM-INF`
- `EXT-X-SESSION-DATA`
- `EXT-X-SESSION-KEY`