What's new in HTTP Live Streaming 2023

Discover the latest features.

Overview

This document highlights new features in HTTP Live Streaming (HLS). These features are applicable to content producers and developers.

Some of these features are described in more detail in the HLS Specification. The current published draft of the HLS Specification is the IETF draft. However, a preliminary version of the next draft is available as a PDF on the streaming landing page. That version should be consulted as well.

Apple Vision Pro support

HLS supports delivery of 3D videos in visionOS. Multivariant playlists can be created that include both 3D and 2D variants, allowing the same playlist to be used with other devices.

The 3D content must be encoded using Multiview HEVC (MV-HEVC) in an MP4 file with a Video Extended Usage atom. Tags describing 3D content in a multivariant playlist must include a REQ-VIDEO-LAYOUT attribute.

For more details see the most recent versions of the HLS specification, the HLS Authoring Specification, and its appendixes.

Localized media names

The media rendition tags in the multivariant playlist contain a name attribute. Because this name can be provided only in a single language, `AVFoundation` typically synthesizes a localized display name based on the attributes listed in the tag.

Some tags do not have enough information to support synthesis of a suitable localized name. The new localization feature allows content producers to supply their own localizations for these renditions. While most audio or subtitle renditions will not need to have a localization supplied, it is useful when you have an alternate audio track, such as director's commentary.

Suppose your multivariant playlist contains session data and a media tag like this:

```
#EXT-X-SESSION-DATA:DATA-ID="_hls.localized-rendition-names",URI=…
#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="AU",NAME="Director’s commentary",LANGUAGE="en",…
```

Localizations are specified in a JSON file given by the session data tag.

```
{
    "Director’s commentary" :
    {
        "en" : "Director’s commentary",
        "de" : "Kommentar des Regisseurs",
        "es" : "Comentario del director"
   }
}
```

With this the player would be able to localize the name in English, German, and Spanish. For an untranslated locale the player would need to use the name without translation.

For more details please see the most recent HLS specification.

Lossless audio attributes

Lossless audio requires some extra information to enable the player to make suitable choices. There are two new attributes - BIT-DEPTH and SAMPLE-RATE. These attributes enable clients to select between lossless renditions to avoid bit depth or sample rate conversions, ensuring the highest possible audio fidelity. They are typically not provided for lossy encodings.

There can be a mix of bit depth and sample rate declarations. It's allowable for some renditions in an audio group to have either one or both of them while others have neither. In practice they are useful only when multiple renditions are present and the content provider wishes to signal that some of them should only be used when the audio output configuration offers special capabilities.

For example, if you have two Apple Lossless renditions encoded at 48 kHz and 96 kHz, declaring the sample rate allows a player to reject the 96 kHz rendition when its audio output hardware does not support that sample rate.

On the other hand, if you want to allow a switch to a high sample rate lossless rendition, even when this would require a sample rate down-conversion, then you can omit the declaration of the sample rate.

For more details please see the most recent HLS specification.

Spatial audio indicators

Spatial audio requires some new identifiers for informed selection and processing of the audio. This is included via the CHANNELS attribute. The third parameter of the attribute contains supplementary indications of special channel usage.

Generally, audio can be dynamically spatialized by the receiving device. Some audio content is already spatitalized when created and would be distorted if passed through a second spatialization process.

Audio may be marked as BINAURAL. This indicates it was recorded or created as binaural content and is optimized for delivery to headphones. Or audio may be marked as IMMERSIVE. This indicates it is pre-processed content, but not restricted to headphones. If neither of these identifiers are present then the content may be spatialized.

In addition, audio may be marked as DOWNMIX. This indicates the audio is a downmix derivative of some other audio. If desired, the player may use the downmix as a substitute for other renditions in the same group with compatible attributes and a greater channel count.

For more details please see the most recent HLS specification.

Inserting URI query parameters in playlists

Playlists may contain variables. In the past these had to be defined in the playlist or inherited by a media playlist from the multivariant playlist. This new feature allows query parameters from the source URI of the playlist to be used as variables in the playlist.

One use case is to provide token authentication to playlists and its associated resources. Content producers formerly did this using real-time generation of playlists to copy the token into the other URIs present in the playlist. With the new mechanism a query parameter can be copied into a playlist variable and used like any other variable. This allows for highly cacheable playlist files, avoids real-time manipulation, and provides better scalability.

Here’s an example of the syntax:

Given a URL: `https://example.com/playlist.m3u8?token=abc123&service="Dingbats"`

And a playlist:

```
EXTM3U
EXT-X-VERSION:11
EXT-X-TARGETDURATION:6
EXT-X-DEFINE:QUERYPARAM="token"
EXT-X-DEFINE:QUERYPARAM="service"
EXTINF 6,
Segment0.mp4?cdn-token={$token}&origin={$service}
```

The resolved segment URL will be `https://example.com/Segment0.mp4?cdn-token=abc123&origin="Dingbats"`

For more details please see the most recent HLS specification.

Marking content containing flashing lights

Beginning in iOS 16.4 and iPadOS 16.4 there is an accessibility setting Dim Flashing Lights. When someone enables this setting on an Apple device, the device automatically dims video when flashes or strobe effects are detected.

As an additional measure, content producers can add markers to their HLS content that identify the regions likely to cause discomfort.

For more information see the section "Dim Flashing Lights" in most recent version of the HLS Authoring Specification appendixes.

Recommendations for I-frame image sequences

Last year, support was added for JPEG-encoded images. This allows you supply an I-frame only playlist as JPEG images inside a fragmented MP4 file. While we recommend using a video codec for I-frames, some lower-end devices from other manufacturers cannot spawn a second video decoder instance, but can decode JPEG in software.

See the section on "I-frame image sequences" in the most recent version of the HLS Authoring Specification appendixes.

Dolby Vision 8.1 content on tvOS

Dolby Vision 8.1 content is backward-compatible with HDR10+ and HDR10. This allows the stream to be played on a Dolby Vision compatible TV or an HDR10 TV.

Properly declaring a Dolby Vision 8.1 variant requires specifying the codec as HEVC/H.265 and adding a SUPPLEMENTAL-CODECS attribute which marks the content at Dolby Vision 8.1.

See the section on "The SUPPLEMENTAL-CODECS attribute" in the HLS Authoring Specification appendixes, as well as the HLS specification.

Downloading HLS audio streams on watchOS

New in watchOS 10.0, the AVAssetDownloadTask family of APIs, which enable persisting HLS streams to disk, is becoming available for usage in Apple Watch applications. This change will enable applications to play audio content when network connectivity is not available or to cache frequently played content in order to start playback faster. As with iOS, HLS downloads occur in background and do not require the application to be running while a download is in progress. FairPlay encrypted audio content is supported.

There are two important differences to consider when using this API on watchOS as compared to iOS or macOS.

1. **Only HLS audio streams.** HLS downloads on watchOS are only intended for audio streams. Attempting to download streams containing other media types will result in undefined behavior.

2. **Only while charging.** Background downloads can be scheduled at any time during the execution of a watchOS application, but they will only be allowed to start when the device is charging. This policy is intended to preserve battery life, as large network downloads would otherwise result in important battery drain.

HLS download support for content steering

HLS download has improved support for content steering. Previously the downloader would would pick a pathway at the start and use only that pathway for downloading. Now the downloader periodically reads the steering manifest and responds to changes in pathway priority. However, when downloading, a failure to fetch a specific file terminates the download, rather than causing a switch to an alternate pathway.

Downloadable content with content steering requires that the pathways have consistent stable IDs for all variants and renditions. In addition, renditions with matching stable IDs are required to have bit-for-bit identical media segments.