

What Is New in HTTP Live Streaming

Session 512

Roger Pantos

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

Today

- New HTTP Live Streaming features in iOS 6

Today

- New HTTP Live Streaming features in iOS 6
 - Multinational subtitles

Today

- New HTTP Live Streaming features in iOS 6
 - Multinational subtitles
- New AVFoundation APIs to make your lives easier

Subtitles

Subtitles

- Descriptive/narrative text displayed in time with audio and video
- Used to deliver localization and accessibility
- Displayed explicitly by user, or implicitly by content
- Often added post-production, deployed over time

Subtitle Format—WebVTT

- Web Video Text Tracks (WebVTT)
 - Published by the W3C Web Media Text Tracks Community Group
 - Origins in earlier SubRip (.srt) format
 - Simple text-based files
 - Subtitles are carried as Unicode strings
- Public Specification
<http://dev.w3.org/html5/webvtt/>

WebVTT Example

- Header has magic number and metadata
- Followed by “cues”

```
WEBVTT
```

```
X-TIMESTAMP-MAP=MPEGTS:900000, LOCAL:00:00:00.000
```

```
00:11.000 --> 00:13.000
```

```
Let's play a game
```

```
00:15.300 --> 00:17.900
```

```
You know I don't like games
```

```
(more cues go here)
```


WebVTT Example

- Header has magic number and metadata
- Followed by “cues”

WEBVTT

X-TIMESTAMP-MAP=MPEGTS:900000, LOCAL:00:00:00.000

00:11.000 --> 00:13.000

Let's play a game

00:15.300 --> 00:17.900

You know I don't like games

(more cues go here)

WebVTT Example

- Header has magic number and metadata
- Followed by “cues”

```
WEBVTT
```

```
X-TIMESTAMP-MAP=MPEGTS:900000, LOCAL:00:00:00.000
```

```
00:11.000 --> 00:13.000
```

```
Let's play a game
```

```
00:15.300 --> 00:17.900
```

```
You know I don't like games
```

```
(more cues go here)
```

WebVTT in HTTP Live Streaming

- WebVTT files are divided into segments
- Each segment is a complete WebVTT file, with timestamp metadata
- Segments are collected into a subtitle playlist file (m3u8)
- Subtitle playlists may be static, or live
- Subtitle playlists appear as alternative media in master playlist

`#EXT-X-MEDIA:TYPE=SUBTITLES`

WebVTT in HTTP Live Streaming

Master Playlist

64 kbps
[Audio-Only]

400kbps

800kbps

1500kbps

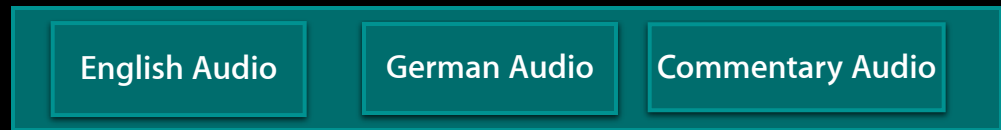
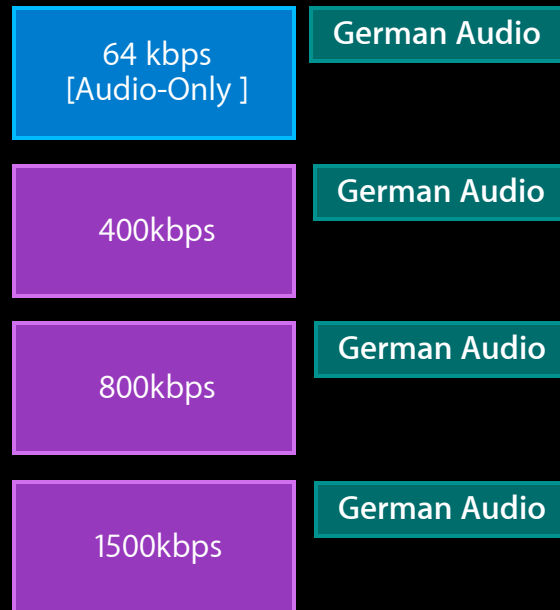
English Audio

German Audio

Commentary Audio

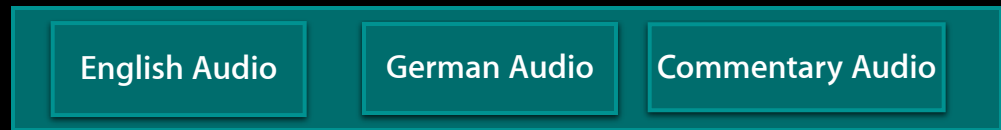
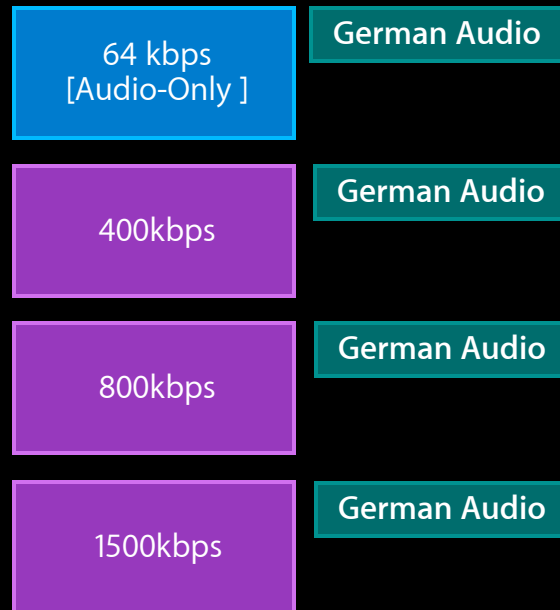
WebVTT in HTTP Live Streaming

Master Playlist



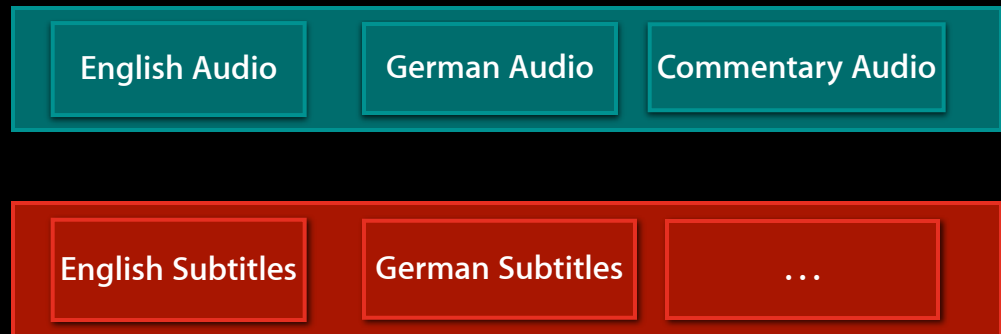
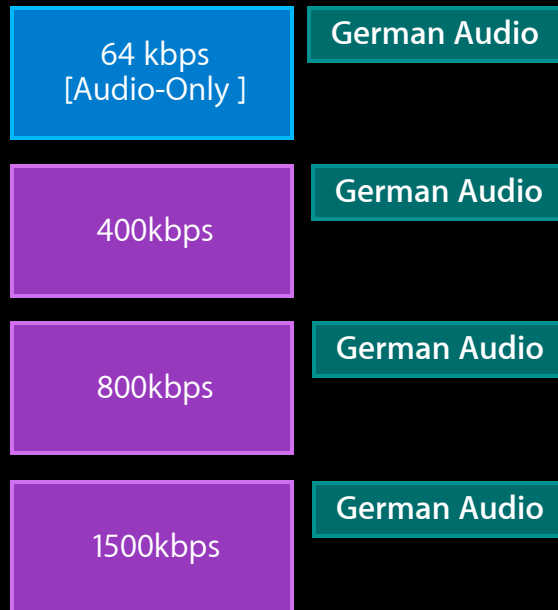
WebVTT in HTTP Live Streaming

Master Playlist



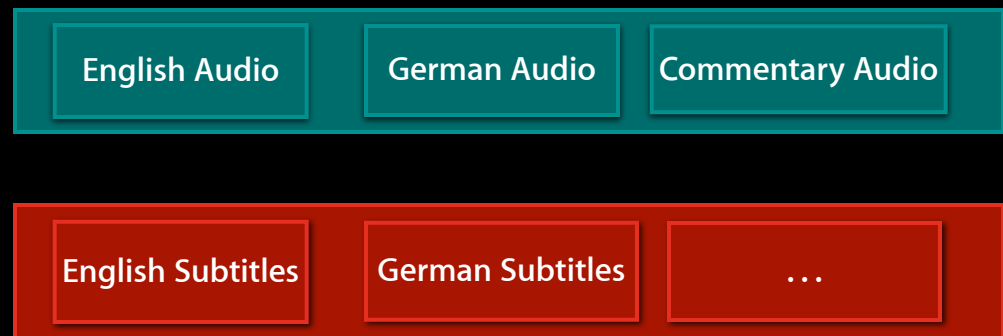
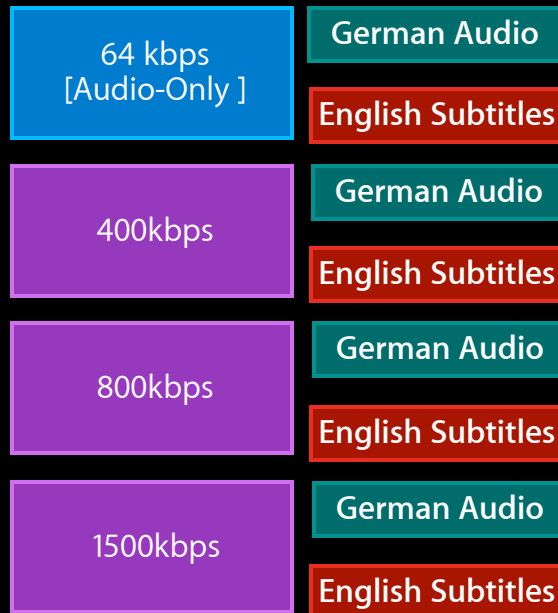
WebVTT in HTTP Live Streaming

Master Playlist



WebVTT in HTTP Live Streaming

Master Playlist



Example: WebVTT in HTTP Live Streaming

- subtitleSegment0.vtt: (0-30s)

WEBVTT

X-TIMESTAMP-MAP=MPEGTS:900000, LOCAL:00:00:00.000

00:11.000 --> 00:13.000

Let's play a game

- subtitleSegment1.vtt: (30-60s)

WEBVTT

X-TIMESTAMP-MAP=MPEGTS:900000, LOCAL:00:00:00.000

00:35.720 --> 00:39.200

Or at least your faith in random chance

Example: WebVTT in HTTP Live Streaming

Example: WebVTT in HTTP Live Streaming

- Subtitle playlist:

```
WEBVTT
X-TIMESTAMP-MAP=MPEGTS:900000,
LOCAL:00:00:00.000
```

```
00:11.000 --> 00:13.000
Let's play a game
```

```
WEBVTT
X-TIMESTAMP-MAP=MPEGTS:900000,
LOCAL:00:00:00.000
```

```
00:28.720 --> 00:32.200
Or at least your faith in
random chance
```

Example: WebVTT in HTTP Live Streaming

- Subtitle playlist:

subtitleSegment0.vtt

subtitleSegment1.vtt

Example: WebVTT in HTTP Live Streaming

- Subtitle playlist:

```
#EXTM3U
#EXT-X-TARGETDURATION:30
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-PLAYLIST-TYPE:VOD
#EXTINF:30,
subtitleSegment0.vtt
#EXTINF:30,
subtitleSegment1.vtt
#EXT-X-ENDLIST
```

Example: WebVTT in HTTP Live Streaming

- Master playlist:

```
#EXTM3U
#EXT-X-TARGETDURATION:30
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-PLAYLIST-TYPE:VOD
#EXTINF:30,
subtitleSegment0.vtt
#EXTINF:30,
subtitleSegment1.vtt
#EXT-X-ENDLIST
```

Example: WebVTT in HTTP Live Streaming

- Master playlist:

```
#EXTM3U
```

```
#EXT-X-MEDIA:TYPE=SUBTITLES,GROUP-ID="subs",NAME="English",  
  DEFAULT=YES,FORCED=NO,URI="vtt/english.m3u8",LANGUAGE="en"
```

```
#EXT-X-MEDIA:TYPE=SUBTITLES,GROUP-ID="subs",NAME="简体中文",  
  DEFAULT=NO,FORCED=NO,URI="vtt/chinese.m3u8",LANGUAGE="zh-Hans"
```

```
#EXT-X-STREAM-INF:BANDWIDTH=2670000,CODECS="mp4a.40.2,avc1.4d401f",  
  RESOLUTION=640x360,SUBTITLES="subs"
```

```
2.5Mbps/prog_index.m3u8
```

Segmentation Note: Time Mapping

- subtitleSegment0.vtt: (0-30s)

WEBVTT

X-TIMESTAMP-MAP=MPEGTS:900000, LOCAL:00:00:00.000

00:11.000 --> 00:13.000

Let's play a game

- subtitleSegment1.vtt: (30-60s)

WEBVTT

X-TIMESTAMP-MAP=MPEGTS:900000, LOCAL:00:00:00.000

00:35.720 --> 00:39.200

Or at least your faith in random chance

Segmentation Note: Time Mapping

- subtitleSegment0.vtt: (0-30s)

WEBVTT

X-TIMESTAMP-MAP=MPEGTS:900000, LOCAL:00:00:00.000

00:11.000 --> 00:13.000

Let's play a game

- subtitleSegment1.vtt: (30-60s)

WEBVTT

X-TIMESTAMP-MAP=MPEGTS:900000, LOCAL:00:00:00.000

00:35.720 --> 00:39.200

Or at least your faith in random chance

Segmentation Note: Time Mapping

- Each WebVTT segment declares a mapping from WebVTT time to MPEG-2 time

`X-TIMESTAMP-MAP:LOCAL=<webVTT timestamp>, MPEGTS=<90000-based timestamp>`

e.g. `X-TIMESTAMP-MAP:LOCAL=00:00:00.000, MPEGTS=900000`

- Should be updated
 - On Discontinuity boundaries
 - When 33-bit MPEG timestamps roll over

Segmentation Note: Time Mapping

- Each WebVTT segment declares a mapping from WebVTT time to MPEG-2 time

`X-TIMESTAMP-MAP:LOCAL=<webVTT timestamp>, MPEGTS=<90000-based timestamp>`

e.g. `X-TIMESTAMP-MAP:LOCAL=00:00:00.000, MPEGTS=900000`

- Should be updated
 - On Discontinuity boundaries
 - When 33-bit MPEG timestamps roll over

Segmentation Note: Segment Durations

Segmentation Note: Segment Durations

- #EXTINF duration indicates timespan of segment data

- Subtitle playlist:

```
#EXTM3U
#EXT-X-TARGETDURATION:30
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-PLAYLIST-TYPE:VOD
#EXTINF:30,
subtitleSegment0.vtt
#EXTINF:30,
subtitleSegment1.vtt
#EXT-X-ENDLIST
```

Segmentation Note: Segment Durations

- #EXTINF duration indicates timespan of segment data
- If timespan has no subtitles, .vtt segment has header but no cues

- Subtitle playlist:

```
#EXTM3U
#EXT-X-TARGETDURATION:30
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-PLAYLIST-TYPE:VOD
#EXTINF:30,
subtitleSegment0.vtt
#EXTINF:30,
subtitleSegment1.vtt
#EXT-X-ENDLIST
```

Segmentation Note: Segment Durations

- #EXTINF duration indicates timespan of segment data
- If timespan has no subtitles, .vtt segment has header but no cues
- Cues that cross segment boundaries are duplicated

- Subtitle playlist:

```
#EXTM3U
#EXT-X-TARGETDURATION:30
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-PLAYLIST-TYPE:VOD
#EXTINF:30,
subtitleSegment0.vtt
#EXTINF:30,
subtitleSegment1.vtt
#EXT-X-ENDLIST
```

Segmentation Note: Target Durations

Segmentation Note: Target Durations

- Live subtitle playlists must use same target duration as other streams

- Subtitle playlist:

```
#EXTM3U
```

```
#EXT-X-TARGETDURATION:30
```

```
#EXT-X-VERSION:3
```

```
#EXT-X-MEDIA-SEQUENCE:0
```

```
#EXT-X-PLAYLIST-TYPE:VOD
```

```
#EXTINF:30,
```

```
subtitleSegment0.vtt
```

```
#EXTINF:30,
```

```
subtitleSegment1.vtt
```

```
#EXT-X-ENDLIST
```

Segmentation Note: Target Durations

- Live subtitle playlists must use same target duration as other streams
- VOD subtitle playlists can have longer target durations
 - Should be a reasonable size (1-5 minutes) for efficient access

- Subtitle playlist:

```
#EXTM3U
```

```
#EXT-X-TARGETDURATION:30
```

```
#EXT-X-VERSION:3
```

```
#EXT-X-MEDIA-SEQUENCE:0
```

```
#EXT-X-PLAYLIST-TYPE:VOD
```

```
#EXTINF:30,
```

```
subtitleSegment0.vtt
```

```
#EXTINF:30,
```

```
subtitleSegment1.vtt
```

```
#EXT-X-ENDLIST
```

Enabling (Non-Forced) Subtitles

- Subtitles appear in an asset as an AVMediaSelectionGroup

```
characteristics = asset.availableMediaCharacteristicsWithMediaSelectionOptions;  
subtitles = [asset mediaSelectionGroupForMediaCharacteristic:  
AVMediaCharacteristicLegible];
```

Enabling (Non-Forced) Subtitles

- Subtitles appear in an asset as an `AVMediaSelectionGroup`

```
characteristics = asset.availableMediaCharacteristicsWithMediaSelectionOptions;  
subtitles = [asset mediaSelectionGroupForMediaCharacteristic:  
AVMediaCharacteristicLegible];
```

- A `MediaSelectionGroup` may contain several variants (e.g. languages)

```
filtered = [AVMediaSelectionGroup  
            mediaSelectionOptionsFromArray:[subtitles options]  
            filteredAndSortedAccordingToPreferredLanguages:  
            [NSLocale preferredLanguages]];
```

Enabling (Non-Forced) Subtitles

- Subtitles appear in an asset as an `AVMediaSelectionGroup`

```
characteristics = asset.availableMediaCharacteristicsWithMediaSelectionOptions;  
subtitles = [asset mediaSelectionGroupForMediaCharacteristic:  
AVMediaCharacteristicLegible];
```

- A `MediaSelectionGroup` may contain several variants (e.g. languages)

```
filtered = [AVMediaSelectionGroup  
           mediaSelectionOptionsFromArray:[subtitles options]  
           filteredAndSortedAccordingToPreferredLanguages:  
           [NSLocale preferredLanguages]];
```

- Set an option from the group on the `AVPlayerItem`

```
[myPlayerItem selectMediaOption:[filtered objectAtIndex:0]  
              inMediaSelectionGroup:subtitles];
```

In Summary

- Create a .vtt file containing all your timed subtitle text
- Divide it into segments; place them on a web server
- Put a URL for each segment into a subtitle playlist file
- Add the subtitle playlist URL to your master playlist
- Discover, select, and enable the subtitle playlist at runtime

Adding Forced Subtitles

- Forced subtitles appear without user intervention

```
#EXT-X-MEDIA:TYPE=SUBTITLES,GROUP-ID="subs",NAME="English (f)",  
DEFAULT=YES,FORCED=YES,URI="forced_subtitles.m3u8"
```



- Non-forced subtitle alternates must include all forced subtitles

Tagging Subtitles with Semantic Info

- Media can be tagged with “characteristics” in playlist
- Characteristics appear in AVMediaSelectionOption
- Subtitles for the Deaf and Hard of Hearing (SDH) should be flagged with:
`#EXT-X-MEDIA:TYPE=SUBTITLES,GROUP-ID="subs",NAME="English (sdh)",
DEFAULT=YES,FORCED=NO,URI="sdh_subtitles.m3u8",
CHARACTERISTICS="public.accessibility.transcribes-spoken-dialog,public.accessibility.describes-music-and-sound"`



Tagging Subtitles with Semantic Info

- Media can be tagged with "characteristics" in playlist
- Characteristics are used by AVMediaSelection
- Subtitles for the

```
#EXT-X-MEDIA:TYPE=SUBTITLES,DEFAULT=YES,FORCED=YES,CHARACTERISTICS="public"
```



be flagged with:

```
(sdh)",
```

```
ity.describes-music-and-sound"
```

Moving Subtitles Out of the Way

- Display position is controlled by cue attributes

`00:00:29.500 --> 00:00:31.300 line:20% align:start`
and developed so many
amazing technologies

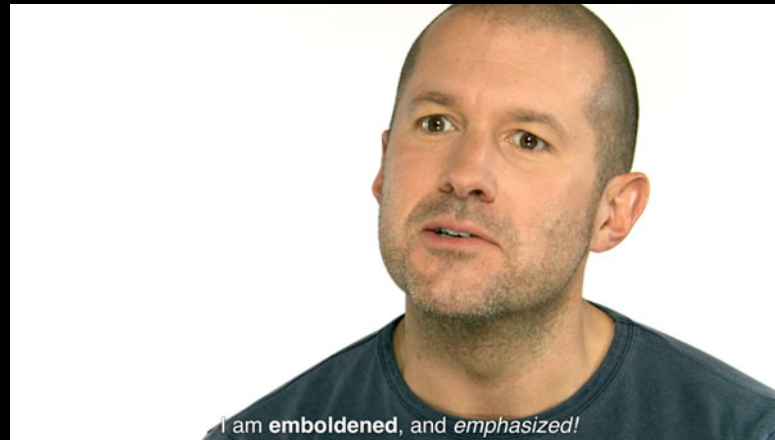


Controlling Built-In Styles

- WebVTT provides built-in Bold, Italic, and Underline styles

00:00:30.000 --> 00:00:35.000

I am `emboldened`, and `<i>emphasized!</i>`



Advanced Styling: WebVTT Classes

- Cue text may be tagged with a CSS class:

00:00:45.000 --> 00:00:49.000

an incredible <c.huge>BREAKTHROUGH</c> product



- The styling attached to a class is specified at runtime:

```
huge = [AVTextStyleRule textStyleRuleWithTextMarkupAttributes:
        [NSDictionary dictionaryWithObject:[NSNumber numberWithInt:1.5]
        forKey:kCMTextMarkupAttribute_RelativeFontSize]
        textSelector:@"huge"];
myPlayerItem.textStyleRules = [NSArray arrayWithObject:huge];
```

Per-Voice Styling

- `<v >` is a special type of class, used to distinguish voices:

```
00:00:52.000 --> 00:00:54.000 line:92% align:start  
<v Jony>...so capable</v>
```

```
00:00:53.000 --> 00:00:56.000 line:95% align:end  
<v Phil>The iPhone was a revolution.</v>
```



Animated Styles

- A cue's timespan can be divided into ranges

00:00:10.000 --> 00:00:15.000

My <00:00:11.000>heart <00:00:12.000>cries <00:00:13.000>for <00:00:14.000>you

- During its display period

The special style :past is applied to ranges prior to current time

The special style :future is applied to ranges after current time

Demo

WebVTT Subtitles

Bill May

For more information on WebVTT

- WebVTT Specification available from W3C

<http://dev.w3.org/html5/webvtt/>

More New Stuff

FF/RW Support in MPMoviePlayerController

- Stream must supply at least one I-frame playlist

FF/RW Support in MPMoviePlayerController

- Stream must supply at least one I-frame playlist



Glitch-Free Audio Switching

- Eliminates pops and gaps when switching between different audio encodings
- Audio timestamps must match exactly for a perfect switch

Glitch-Free Audio Switching

- Eliminates pops and gaps when switching between different audio encodings
- Audio timestamps must match exactly for a perfect switch



High Quality

Glitch-Free Audio Switching

- Eliminates pops and gaps when switching between different audio encodings
- Audio timestamps must match exactly for a perfect switch



High Quality



Low Quality

Glitch-Free Audio Switching

- Eliminates pops and gaps when switching between different audio encodings
- Audio timestamps must match exactly for a perfect switch



High Quality



Low Quality



Switch-up

Glitch-Free Audio Switching

- Eliminates pops and gaps when switching between different audio encodings
- Audio timestamps must match exactly for a perfect switch



High Quality



Low Quality



Switch-up

AC-3 Pass-Through Support

- Pass-through only, using AirPlay or HDMI Adaptor
 - Requires compatible receiver

AC-3 Pass-Through Support

- Pass-through only, using AirPlay or HDMI Adaptor
 - Requires compatible receiver
- Must be accompanied by AAC variant for compatibility
 - AC-3 is automatically selected if output is available

AC-3 Pass-Through Support

- Pass-through only, using AirPlay or HDMI Adaptor
 - Requires compatible receiver
- Must be accompanied by AAC variant for compatibility
 - AC-3 is automatically selected if output is available
- Indicate AC-3 in CODECS attribute

```
#EXT-X-STREAM-INF:BANDWIDTH=3016809, CODECS="ac-3, avc1.4d001e"  
video_plus_ac3/prog_index.m3u8
```

```
#EXT-X-STREAM-INF:BANDWIDTH=2716122, CODECS="mp4a.40.2, avc1.4d001e"  
video_plus_aac/prog_index.m3u8
```

Tuning Bit Rate Switches

- Better Switch-up Algorithm
 - Requires a smaller “bandwidth cushion” if connection is stable
- Better Switch-down Algorithm
 - Identifies rapid bandwidth drops more quickly

Playback Data Starvation Notification

- Sent when playback stalls waiting for network data

```
NSString *const AVPlayerItemPlaybackStalledNotification;
```

Access and Error Log Additions

- New fields in Access log, including:
 - Playback type (VOD, EVENT, LIVE)
 - Minimum, maximum, and standard deviation of observed bandwidth
 - Bandwidth at point of switch
 - Number of segments downloaded slower than real time
 - Number of segments downloaded over cellular
- More details in Error Log when playlist parsing fails

Access and Error Log Notifications

- Sent when new entries are added to a log

```
NSString *const AVPlayerItemAccessLogNewEventAddedNotification;
```

```
NSString *const AVPlayerItemErrorLogNewEventAddedNotification;
```

Easier API for Loading App-Defined URLs

- API on AVURLAsset:

```
@property AVAssetResourceLoader *resourceLoader;
```


Easier API for Loading App-Defined URLs

- API on AVURLAsset:

```
@property AVAssetResourceLoader *resourceLoader;
```

- Application sets delegate on AVAssetResourceLoader:

```
[asset.resourceLoader setDelegate:myDelegate queue:my_queue];
```

Easier API for Loading App-Defined URLs

- API on AVURLAsset:

```
@property AVAssetResourceLoader *resourceLoader;
```

- Application sets delegate on AVAssetResourceLoader:

```
[asset.resourceLoader setDelegate:myDelegate queue:my_queue];
```

- Delegate receives requests from Loader:

```
-(BOOL)resourceLoader:shouldWaitForLoadingOfRequestedResource::;
```

Easier API for Loading App-Defined URLs

- API on AVURLAsset:

```
@property AVAssetResourceLoader *resourceLoader;
```

- Application sets delegate on AVAssetResourceLoader:

```
[asset.resourceLoader setDelegate:myDelegate queue:my_queue];
```

- Delegate receives requests from Loader:

```
-(BOOL)resourceLoader:shouldWaitForLoadingOfRequestedResource::;
```

- Delegate responds to requests:

```
[asset.resourceLoader finishLoadingWithResponse:data:redirect:];
```

Sample-Level Stream Encryption

- Only samples require decryption, which means better battery life
- Encrypted files can be processed by standard tools
- Encryption applied to elementary audio and video streams
 - Which are then carried in regular .ts and .aac files
 - New stream_types for .ts files
 - Codec configuration is carried in the clear
- EXT-X-KEY syntax:
`#EXT-X-KEY:METHOD=SAMPLE-AES,URI="foo://key"`

Summary

- iOS 6 makes it easy to add subtitles to your streams
- Test your apps and streams with the WWDC seed
- Install the seed on Apple TV and test AirPlay

More Information

HTTP Live Streaming Resources

<https://developer.apple.com/resources/http-streaming/>

WebVTT

<http://dev.w3.org/html5/webvtt/>

Eryk Vershen

Media Technology Evangelist
evershen@apple.com

Ask questions in the ADC forum (follow link on Resources page)

Related Sessions

Effective HTTP Live Streaming

Pacific Heights
Tuesday 11:30AM

Labs

HTTP Live Streaming Lab

Graphics, Media & Games Lab B
Wednesday 2:00PM

Q&A

 WWDC2012

