Measuring and Optimizing HLS Performance

Session 502

Emil Andriescu, Media Streaming Engineer
Best User Experience for HLS
Best User Experience for HLS

Content authoring → playlist authoring, media formats, codecs
Best User Experience for HLS

Content authoring → playlist authoring, media formats, codecs

Application design → using the AVFoundation API
Best User Experience for HLS

Content authoring → playlist authoring, media formats, codecs

Application design → using the AVFoundation API

Delivery → server configuration and performance
Best User Experience for HLS

Content authoring → playlist authoring, media formats, codecs

Application design → using the AVFoundation API

Delivery → server configuration and performance

Understand and quantify user experience in conjunction with your changes
Best User Experience for HLS

Content authoring → playlist authoring, media formats, codecs

Application design → using the AVFoundation API

Delivery → server configuration and performance

Understand and quantify user experience in conjunction with your changes

Measure to find the optimal configuration
What Is This Session About?
What Is This Session About?

Establish a common language for streaming Quality of Service
What Is This Session About?

Establish a common language for streaming Quality of Service

Objectively measure the streaming QoS
What Is This Session About?

Establish a common language for streaming Quality of Service
Objectively measure the streaming QoS
Identify and solve problems that impair QoS
What Is This Session About?

Establish a common language for streaming Quality of Service
Objectively measure the streaming QoS
Identify and solve problems that impair QoS
Get master playlists right
Anatomy of an HLS Playback Session

- Not playing
- Loading playlist
- 2 Mbps download
- 1 Mbps download
- Playout duration
Anatomy of an HLS Playback Session

- Not playing
- Loading playlist
- 2 Mbps download
- 1 Mbps download
- Playout duration
Anatomy of an HLS Playback Session

- Not playing
- Loading playlist
- 2 Mbps download
- 1 Mbps download
- Playout duration
Anatomy of an HLS Playback Session

- Not playing
- Loading playlist
- 2 Mbps download
- 1 Mbps download
- Playout duration
Anatomy of an HLS Playback Session

Startup Time

AVPlayerItem.isPlaybackLikelyToKeepUp == true
Anatomy of an HLS Playback Session

Startup Time

AVPlayerItem.isPlaying == true

Not playing
Loading playlist
2 Mbps download
1 Mbps download
Playout duration
Anatomy of an HLS Playback Session

Startup Time

AVPlayerItem.isPlaybackLikelyToKeepUp == true
Anatomy of an HLS Playback Session

Startup Time

AVPlayerItem.isPlaybackLikelyToKeepUp == true
Anatomy of an HLS Playback Session

- **Startup Time**
- **Stall Duration**

AVPlayerItem.isPlaybackLikelyToKeepUp == true

- Not playing
- Loading playlist
- 2 Mbps download
- 1 Mbps download
- 1 Mbps download
- Playout duration

1 Mb
2 Mb
3
4
5
6
7
8
Anatomy of an HLS Playback Session

AVPlayerItem.isPlaybackLikelyToKeepUp == true
Quantifying the User Experience
Key Performance Indicators (KPIs) for HLS

| Startup Time | Playback at 2 Mbps | 1 Mbps | Stall Duration | 1 Mbps |
How much time did the user spend waiting for playback to start?
Quantifying the User Experience
Key Performance Indicators (KPIs) for HLS

- Startup Time
- Playback at 2 Mbps
- 1 Mbps
- Stall Duration
- Playback at 2 Mbps
- 1 Mbps

How much time did the user spend waiting for playback to start?
How often was playback interrupted?
How long was playback interrupted?
Quantifying the User Experience
Key Performance Indicators (KPIs) for HLS

Startup Time
Playback at 2 Mbps
1 Mbps
Stall Duration
1 Mbps

How much time did the user spend waiting for playback to start?
How often was playback interrupted?
How long was playback interrupted?
What was the overall media quality?
Quantifying the User Experience
Key Performance Indicators (KPIs) for HLS

How much time did the user spend waiting for playback to start?
How often was playback interrupted?
How long was playback interrupted?
What was the overall media quality?
How often did the stream end due to an error?
Startup Time
How much time did users spend waiting for playback to start?
Startup Time
How much time did users spend waiting for playback to start?

AVPlayer.status == .readyToPlay
Startup Time

How much time did users spend waiting for playback to start?

- AVPlayerItem.isPlayingLikelyToKeepUp == true
- AVPlayerItem.status == .readyToPlay
- AVPlayer.status == .readyToPlay
Startup Time
How much time did users spend waiting for playback to start?

- kCMTimebaseNotification_EffectiveRateChanged
- CMTimebaseGetRate(AVPlayerItem.timebase)
- AVPlayer.timeControlStatus == .playing
- AVPlayerItem.isPlaybackLikelyToKeepUp == true
- AVPlayerItem.status == .readyToPlay
- AVPlayer.status == .readyToPlay
Stall Rate
How often was playback interrupted?

- Startup Time
- Playback at 2 Mbps
- 1 Mbps
- Stall Duration
- 1 Mbps
Stall Rate
How often was playback interrupted?

Startup Time
Playback at 2 Mbps 1 Mbps

Stall Duration 1 Mbps

AVPlayerItemPlaybackStalled
Stall Rate
How often was playback interrupted?

KPI | Unit
---|---
Stall rate (normalized to duration watched) | stalls / hour watched
Stall Duration
How long was playback interrupted?

Startup Time
- Playback at 2Mbps
- 1 Mbps
- Stall Duration
- 1 Mbps

AVPlayerItemPlaybackStalled
Stall Duration
How long was playback interrupted?

- Startup Time
- Playback at 2Mbps
- 1 Mbps
- Stall Duration
- 1 Mbps

AVPlayerItemPlaybackStalled

kCMTIMEbaseNotification_EffectiveRateChanged
Stall Duration
How long was playback interrupted?

Startup Time
Playback at 2Mbps
1 Mbps
Stall Duration
1 Mbps

AVPlayerItemPlaybackStalled

kCMT imebaseNotification_EffectiveRateChanged

KPI
Unit
Stall duration to duration watched ratio
total stall duration / total duration watched
//Computing total duration watched from AVPlayerItem’s access log

var totalDurationWatched = 0.0
if let accessLog = playerItem.accessLog() {
    for event in accessLog.events {
        if event.durationWatched > 0 {
            totalDurationWatched += event.durationWatched
        }
    }
}
//Computing total duration watched from AVPlayerItem’s access log

var totalDurationWatched = 0.0
if let accessLog = playerItem.accessLog() {
    for event in accessLog.events {
        if event.durationWatched > 0 {
            totalDurationWatched += event.durationWatched
        }
    }
}
//Computing total duration watched from AVPlayerItem’s access log

var totalDurationWatched = 0.0
if let accessLog = playerItem.accessLog() {
    for event in accessLog.events {
        if event.durationWatched > 0 {
            totalDurationWatched += event.durationWatched
        }
    }
}
// Computing total duration watched from AVPlayerItem’s access log

var totalDurationWatched = 0.0
if let accessLog = playerItem.accessLog() {
    for event in accessLog.events {
        if event.durationWatched > 0 {
            totalDurationWatched += event.durationWatched
        }
    }
}
AVPlayerItemAccessLog
Provides a history of the playback session

Startup Time | Playback at 2 Mbps | 1 Mbps | Stall Duration | 1 Mbps
AVPlayerItemAccessLog

Provides a history of the playback session

Startup Time → Playback at 2 Mbps → 1 Mbps → Stall Duration → 1 Mbps

AVPlayerItemNewAccessLogEntry

AVPlayerItemAccessLog

AVPlayerItemAccessLogEvent

- URI = nil
- indicatedBitrate = -1
- observedBitrate = -1
- numberOfBytesTransferred = -1
- durationWatched = -1
- numberOfStalls = -1
AVPlayerItemAccessLog
Provides a history of the playback session

AVPlayerItemNewAccessLogEntry
AVPlayerItemAccessLogEvent

Startup Time
Playback at 2 Mbps
1 Mbps
Stall Duration
1 Mbps

AVPlayerItemAccessLog

AVPlayerItemAccessLogEvent

URI = <2 Mbps variant URI>
indicatedBitrate = 2 Mbps
observedBitrate = 1.2 Mbps
numberOfBytesTransferred = 8MB
durationWatched = 10s
numberOfStalls = 0

Playback at 2 Mbps
AVPlayerItemAccessLog

Provides a history of the playback session

AVPlayerItemNewAccessLogEntry

AVPlayerItemAccessLog

AVPlayerItemAccessLogEvent

URI = <2 Mbps variant URI>
indicatedBitrate = 2 Mbps
observedBitrate = 1.2 Mbps
numberOfBytesTransferred = 8MB
durationWatched = 10s
numberOfStalls = 0
AVPlayerItemAccessLog
Provides a history of the playback session

AVPlayerItemNewAccessLogEntry

AVPlayerItemAccessLog

AVPlayerItemAccessLogEvent

URI = <1 Mbps variant URI>
indicatedBitrate = 1 Mbps
observedBitrate = 1 Mbps
numberOfBytesTransferred = 5MB
durationWatched = 5s
numberOfStalls = 0
Time-Weighted Indicated Bitrate

What was the overall media quality?

- Startup time
- Playback at 2Mbps
- 1Mbps
- Stall duration
- 1Mbps
Time-Weighted Indicated Bitrate

What was the overall media quality?

Startup time

Playback at 2Mbps

1Mbps

Stall duration

1Mbps
Time-Weighted Indicated Bitrate

What was the overall media quality?

- Playback at 2Mbps
- 1Mbps
- 1Mbps
Time-Weighted Indicated Bitrate

What was the overall media quality?

- Playback at 2Mbps
- 1Mbps for 10 seconds
- 1Mbps for 5 seconds

\[(\frac{10s}{15s}) \times 2\text{ Mbps}\]
Time-Weighted Indicated Bitrate

What was the overall media quality?

(10s / 15s) * 2 Mbps + (3s / 15s) * 1 Mbps + (2s / 15s) * 1 Mbps
Time-Weighted Indicated Bitrate

What was the overall media quality?

Playback at 2Mbps

(10s / 15s) * 2 Mbps + (3s / 15s) * 1 Mbps + (2s / 15s) * 1 Mbps

1.66 Mbps
**Time-Weighted Indicated Bitrate**

What was the overall media quality?

\[
\text{KPI: Time-weighted indicated bitrate} \quad \text{Unit: Mbps}
\]
/Computing time-weighted indicated bitrate from AVPlayerItem’s access log

var timeWeightedIBR = 0.0
if let accessLog = playerItem.accessLog(), totalDurationWatched > 0 {
    for event in accessLog.events {
        if event.durationWatched > 0 && event.indicatedBitrate > 0 {
            let eventTimeWeight = event.durationWatched / totalDurationWatched
            timeWeightedIBR += event.indicatedBitrate * eventTimeWeight
        }
    }
}
//Computing time-weighted indicated bitrate from AVPlayerItem’s access log

var timeWeightedIBR = 0.0

if let accessLog = playerItem.accessLog(), totalDurationWatched > 0 {
    for event in accessLog.events {
        if event.durationWatched > 0 && event.indicatedBitrate > 0 {
            let eventTimeWeight = event.durationWatched / totalDurationWatched
            timeWeightedIBR += event.indicatedBitrate * eventTimeWeight
        }
    }
}
//Computing time-weighted indicated bitrate from AVPlayerItem's access log

var timeWeightedIBR = 0.0
if let accessLog = playerItem.accessLog(), totalDurationWatched > 0 {
    for event in accessLog.events {
        if event.durationWatched > 0 && event.indicatedBitrate > 0 {
            let eventTimeWeight = event.durationWatched / totalDurationWatched
            timeWeightedIBR += event.indicatedBitrate * eventTimeWeight
        }
    }
}
//Computing time-weighted indicated bitrate from AVPlayerItem's access log

var timeWeightedIBR = 0.0
if let accessLog = playerItem.accessLog(), totalDurationWatched > 0 {
    for event in accessLog.events {
        if event.durationWatched > 0 && event.indicatedBitrate > 0 {
            let eventTimeWeight = event.durationWatched / totalDurationWatched
            timeWeightedIBR += event.indicatedBitrate * eventTimeWeight
        }
    }
}
//Computing time-weighted indicated bitrate from AVPlayerItem’s access log

```swift
var timeWeightedIBR = 0.0

if let accessLog = playerItem.accessLog(), totalDurationWatched > 0 {
    for event in accessLog.events {
        if event.durationWatched > 0 && event.indicatedBitrate > 0 {
            let eventTimeWeight = event.durationWatched / totalDurationWatched
            timeWeightedIBR += event.indicatedBitrate * eventTimeWeight
        }
    }
}
```
Computing time-weighted indicated bitrate from AVPlayerItem’s access log

```swift
var timeWeightedIBR = 0.0
if let accessLog = playerItem.accessLog(), totalDurationWatched > 0 {
    for event in accessLog.events {
        if event.durationWatched > 0 && event.indicatedBitrate > 0 {
            let eventTimeWeight = event.durationWatched / totalDurationWatched
            timeWeightedIBR += event.indicatedBitrate * eventTimeWeight
        }
    }
}
```
Playback Failure Percentage
How often did the stream end due to an error?

- Startup Time
- Playback at 2 Mbps
Playback Failure Percentage
How often did the stream end due to an error?

- Startup Time
- Playback at 2 Mbps

❌
Playback Failure Percentage
How often did the stream end due to an error?

Startup Time → Playback at 2 Mbps → AVPlayerItem.status == .failed

AVPlayerItem.error // Examine a playback failure
Playback Failure Percentage

How often did the stream end due to an error?

- **AVPlayerItem.status** == `.failed`
- **AVPlayerItem.error** // Examine a playback failure

<table>
<thead>
<tr>
<th>KPI</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Playback failure percentage</td>
<td>Percent of total sessions</td>
</tr>
</tbody>
</table>
AVPlayerItemErrorLog
Conveys failures with varying degrees of user impact
AVPlayerItemErrorLog

Conveys failures with varying degrees of user impact

Startup Time

Playback at 2 Mbps

1 Mbps

Stall Duration

1 Mbps

AVPlayerItemErrorLogEvent

AVPlayerItemErrorLogEvent

AVPlayerItemErrorLogEvent

AVPlayerItemNewErrorLogEntry
AVPlayerItemErrorLog
Conveys failures with varying degrees of user impact

- Startup Time
- Playback at 2 Mbps
- 1 Mbps
- Stall Duration
- 1 Mbps

date = 2018-06-16 15:04:41
errorStatusCode = -12889
errorDomain = "CoreMediaErrorDomain"
errorComment = "No response for media file in 9.9767s"
The User Experience
Key performance indicators for HLS

<table>
<thead>
<tr>
<th>User Experience</th>
<th>Metric</th>
<th>KPI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# The User Experience

Key performance indicators for HLS

<table>
<thead>
<tr>
<th>User Experience</th>
<th>Metric</th>
<th>KPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waiting for playback to start</td>
<td>Startup time</td>
<td>Startup time per session</td>
</tr>
</tbody>
</table>
# The User Experience

Key performance indicators for HLS

<table>
<thead>
<tr>
<th>User Experience</th>
<th>Metric</th>
<th>KPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waiting for playback to start</td>
<td>Startup time</td>
<td>Startup time per session</td>
</tr>
<tr>
<td>Playback stalls</td>
<td>Stall count</td>
<td>Stall rate (normalized to duration watched)</td>
</tr>
<tr>
<td>User Experience</td>
<td>Metric</td>
<td>KPI</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>Waiting for playback to start</td>
<td>Startup time</td>
<td>Startup time per session</td>
</tr>
<tr>
<td>Playback stalls</td>
<td>Stall count</td>
<td>Stall rate (normalized to duration watched)</td>
</tr>
<tr>
<td>Waiting for playback to resume</td>
<td>Stall duration</td>
<td>Stall duration to duration watched ratio</td>
</tr>
</tbody>
</table>
## The User Experience

Key performance indicators for HLS

<table>
<thead>
<tr>
<th>User Experience</th>
<th>Metric</th>
<th>KPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waiting for playback to start</td>
<td>Startup time</td>
<td>Startup time per session</td>
</tr>
<tr>
<td>Playback stalls</td>
<td>Stall count</td>
<td>Stall rate (normalized to duration watched)</td>
</tr>
<tr>
<td>Waiting for playback to resume</td>
<td>Stall duration</td>
<td>Stall duration to duration watched ratio</td>
</tr>
<tr>
<td>Overall stream quality</td>
<td>Indicated bitrate</td>
<td>Time-weighted indicated bitrate</td>
</tr>
</tbody>
</table>
## The User Experience

Key performance indicators for HLS

<table>
<thead>
<tr>
<th>User Experience</th>
<th>Metric</th>
<th>KPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waiting for playback to start</td>
<td>Startup time</td>
<td>Startup time per session</td>
</tr>
<tr>
<td>Playback stalls</td>
<td>Stall count</td>
<td>Stall rate (normalized to duration watched)</td>
</tr>
<tr>
<td>Waiting for playback to resume</td>
<td>Stall duration</td>
<td>Stall duration to duration watched ratio</td>
</tr>
<tr>
<td>Overall stream quality</td>
<td>Indicated bitrate</td>
<td>Time-weighted indicated bitrate</td>
</tr>
<tr>
<td>Playback fails</td>
<td>Error</td>
<td>Playback failure percentage</td>
</tr>
</tbody>
</table>
The User Experience

Key performance indicators for HLS

KPIs are not always comparable between device models and usage scenarios
The User Experience

Key performance indicators for HLS

KPIs are not always comparable between device models and usage scenarios

Gather context information along with your streaming metrics
The User Experience
Key performance indicators for HLS

KPIs are not always comparable between device models and usage scenarios
Gather context information along with your streaming metrics
Partition your playback sessions in classes that make sense for your application
More Information

https://developer.apple.com/library/content/navigation/
Improving HLS Performance

Zhenheng Li, Media Streaming Engineer
Ways to Improve User Experience

Reduce startup time
Investigate stalls
Investigate errors
What Can Delay Start of Playback?

Network Activities

- Playlists
  - Default audio and video
  - Decryption keys
  - Spanish audio
What Can Delay Start of Playback?

- AVAsset
- Network Activities
- Playlists
- Default audio and video
- Decryption keys
- Spanish audio
What Can Delay Start of Playback?

- AVAsset
- Inspection
- m3u8
- Network Activities
- Playlists
- Default audio and video
- Decryption keys
- Spanish audio
What Can Delay Start of Playback?

- AVAsset
- AVPlayer
- AVPlayerItem
- Inspection

Network Activities:
- Playlists
- Default audio and video
- Decryption keys
- Spanish audio
What Can Delay Start of Playback?

Network Activities

AVAsset → AVPlayer → AVPlayerItem → Inspection → Buffering → Network Activities

- m3u8
- Decryption keys
- Spanish audio
- Default audio and video
- Playlists
What Can Delay Start of Playback?

Network Activities

- AVAsset
- AVPlayer
- AVPlayerItem

Inspection | Buffering | Fairplay Key Fetching

Network Activities

- m
- u
- 8
- 3
- u
- 8
- 1
- m
- u
- 8
- 3
- u
- 8

Playlists
- Default audio and video
- Decryption keys
- Spanish audio
What Can Delay Start of Playback?

- AVAsset
- AVPlayer
- AVPlayerItem

- Inspection
- Buffering
- Fairplay Key Fetching
- Buffering

Network Activities:

1. Fairplay Key Fetching
2. Buffering
3. Fairplay Key Fetching

Playlists:
- Default audio and video
- Decryption keys
- Spanish audio
What Can Delay Start of Playback?

- AVAsset
- AVPlayer
- AVPlayerItem
- seek

Activities:
- Inspection
- Buffering
- Fairplay Key Fetching
- Buffering

Network Activities:
- m3u8
- 1
- 2
- 3

Additional notes:
- Playlists
- Default audio and video
- Decryption keys
- Spanish audio
What Can Delay Start of Playback?

- AVAsset
- AVPlayer
- AVPlayerItem
- seek

Activities:
- Inspection
- Buffering
- Fairplay Key Fetching
- Buffering
- Rebuffering

Network Activities:
- m3u8
- 1
- m3u8
- 2
- 3
- 100
- 101

Playlists:
- Default audio and video
- Decryption keys
- Spanish audio
What Can Delay Start of Playback?

- AVAsset
- AVPlayer
- AVPlayerItem
- select(AVMediaSelectionOption)

Activities:
- Inspection
- Buffering
- Fairplay Key Fetching
- Buffering
- Rebuffering

Network Activities:
- m3u8
- 1
- 2
- 3
- 100
- 101

Playlists:
- Default audio and video
- Decryption keys
- Spanish audio
What Can Delay Start of Playback?

- AVAsset
- AVPlayer
- AVPlayerItem
- select(AVMediaSelectionOption)
- seek

Inspection | Buffering | Fairplay Key Fetching | Buffering | Rebuffering

Network Activities

Playlists
- Default audio and video
- Decryption keys
- Spanish audio
What Can Delay Start of Playback?

- AVAsset
- AVPlayer
- AVPlayerItem
- select(AVMediaSelectionOption)
- seek
- AVPlayerItem.isPlaybackLikelyToKeepUp == true

Network Activities:
- Inspection
- Buffering
- Fairplay Key Fetching
- Buffering
- Rebuffering

Playlists:
- Default audio and video
- Decryption keys
- Spanish audio
What Can Delay Start of Playback?

 AVAsset  AVPlayer  AVPlayerItem  select(AVMediaSelectionOption)  seek  play
Inspection  Buffering  Fairplay Key Fetching  Buffering  Rebuffering

AVPlayerItem.isPlaybackLikelyToKeepUp == true

Network Activities

Playlists
Default audio and video
Decryption keys
Spanish audio
What Can Delay Start of Playback?

AVAsset → AVPlayer → AVPlayerItem

Inspection | Buffering | Fairplay Key Fetching | Buffering | Rebuffering | Playing

seek
select(AVMediaSelectionOption)
play

AVPlayerItem.isPlaybackLikelyToKeepUp == true

Network Activities:
- Playlists
- Default audio and video
- Decryption keys
- Spanish audio
What Can Delay Start of Playback?

AVAsset → AVPlayer → AVPlayerItem

Inspection → Buffering → Fairplay Key Fetching → Buffering → Rebuffering → Playing

select(AVMediaSelectionOption)

seek

play

AVPlayerItem.isPlaybackLikelyToKeepUp == true

Network Activities

Playlists
Default audio and video
Decryption keys
Spanish audio
How Do I Reduce Startup Time?
How Do I Reduce Startup Time?

Multiple time-consuming operations
How Do I Reduce Startup Time?

Multiple time-consuming operations

Measure time cost
How Do I Reduce Startup Time?

Multiple time-consuming operations

Measure time cost

• Time spent between API calls and Player/PlayerItem status change notifications
How Do I Reduce Startup Time?

Multiple time-consuming operations

Measure time cost

• Time spent between API calls and Player/PlayerItem status change notifications
• AVPlayerItemAccessLogEvent.startupTime
How Do I Reduce Startup Time?

Multiple time-consuming operations

Measure time cost

- Time spent between API calls and Player/PlayerItem status change notifications
- `AVPlayerItemAccessLogEvent.startupTime`
- Time from starting media download to the first `isPlaybackLikelyToKeepUp == true`
Reduce Startup Time—AVAsset

AVAsset → AVPlayer → AVPlayerItem

Inspection → Buffering → Fairplay Key Fetching → Buffering → Rebuffering → Playing

select(AVMediaSelectionOption)

seek

play

AVPlayerItem.isPlaybackLikelyToKeepUp == true

Playlists
Default audio and video
Decryption keys
Spanish audio
Reduce Startup Time—AVAsset

AVPlayerItem

AVPlayer

Buffering

Fairplay Key Fetching

Buffering

Rebuffering

Playing

select(AVMediaSelectionOption)

seek

play

AVPlayerItem.isPlaybackLikelyToKeepUp == true

Playlists
Default audio and video
Decryption keys
Spanish audio
Reduce Startup Time—AVAsset

AVPlayerItem

AVPlayer

Buffering

select(AVMediaSelectionOption)

seek

play

AVPlayerItem.isPlaybackLikelyToKeepUp == true

Playlists
Default audio and video
Decryption keys
Spanish audio
Reduce Startup Time—AVAsset

AVPlayerItem.isPlaybackLikelyToKeepUp == true
Reduce Startup Time—AVAsset

AVPlayerItem.isPlaybackLikelyToKeepUp == true
Reduce Startup Time—AVAsset

AVAsset

Inspection  Fairplay Key Fetching

m3u8 m3u8

Network Activities

iPhone

Jumanji: Welcome to the Jungle

Also Included In

Jumanji 2 Movie Collection

Trailers

Details  Reviews  Related
Reduce Startup Time—AVAsset

Create and inspect AVAsset before user decides to play
Reduce Startup Time—AVAsset

Create and inspect AVAsset before user decides to play

Prefetch decryption keys

• Adopt AVContentKeySession!
Reduce Startup Time—AVPlayer and AVPlayerItem

AVPlayerItem

select (AVMediaSelectionOption)

Seek

Buffering

Rebuffering

Playing

AVPlayerItem.isPlayingLikelyToKeepUp == true

Default audio and video
Spanish audio
Reduce Startup Time—AVPlayer and AVPlayerItem

AVPlayerItem isPlaybackLikelyToKeepUp == true

select (AVMediaSelectionOption)

Buffering
Playing

AVPlayer Item
AVPlayer

Playing

Default audio and video
Spanish audio

Default audio and video
Spanish audio
Reduce Startup Time—AVPlayer and AVPlayerItem

AVPlayerItem.isPlaybackLikelyToKeepUp == true

Default audio and video
Spanish audio
Reduce Startup Time—AVPlayer and AVPlayerItem

AVPlayerItem.isPlaybackLikelyToKeepUp == true

Buffering

Playing

Default audio and video
Spanish audio
Reduce Startup Time—AVPlayer and AVPlayerItem

AVPlayerItem.isPlaybackLikelyToKeepUp == true
Reduce Startup Time—AVPlayer and AVPlayerItem

AVPlayerItem.isPlaybackLikelyToKeepUp == true

Buffering  Playing

Default video and spanish audio
Reduce Startup Time—AVPlayer and AVPlayerItem

Network Activities

Default video and Spanish audio
Reduce Startup Time—AVPlayer and AVPlayerItem

AVPlayerItem

Network Activities

Default video and Spanish audio
Reduce Startup Time—AVPlayer and AVPlayerItem

AVPlayerItem seek

---------------------------------------- Network Activities

Default video and spanish audio
Reduce Startup Time—AVPlayer and AVPlayerItem

AVPlayerItem seek select (AVMediaSelectionOption)

Network Activities

Default video and spanish audio
Reduce Startup Time—AVPlayer and AVPlayerItem

Default video and Spanish audio
Reduce Startup Time—AVPlayer and AVPlayerItem

AVPlayerItem seek select (AVMediaSelectionOption)

Network Activities

AVPlayer play

Default video and spanish audio
Reduce Startup Time—AVPlayer and AVPlayerItem

AVPlayerItem seek select (AVMediaSelectionOption)

Network Activities

AVPlayer play replaceCurrentItem

Default video and Spanish audio
Reduce Startup Time—AVPlayer and AVPlayerItem

AVPlayerItem → seek → select (AVMediaSelectionOption)

Buffering

100 → 101 → Network Activities

AVPlayer → play → replaceCurrentItem

Default video and Spanish audio
Reduce Startup Time—AVPlayer and AVPlayerItem

AVPlayerItem ⬇️ seek ⬆️ select (AVMediaSelectionOption)

AVPlayer.timeControlStatus == .playing

Buffering

100  101

Network Activities

AVPlayer ⬇️ play ⬆️ replaceCurrentItem

Default video and Spanish audio
Reduce Startup Time—AVPlayer and AVPlayerItem

Set up AVPlayerItem before buffering

Set AVPlayer rate before setting player item

AVPlayerItem

Set AVPlayerItem before buffering

Network Activities

Default video and spanish audio
Reduce Startup Time—AVPlayerItem Transition
Reduce Startup Time—AVPlayerItem Transition
Reduce Startup Time—AVPlayerItem Transition

AVQueuePlayer → AVPlayerItem

(1) and (2)

Network Activities

Network Activities
Reduce Startup Time—AVPlayerItem Transition

AVQueuePlayer

AVPlayerItem
(1) and (2)

Item 1

Network Activities

Network Activities
Reduce Startup Time—AVPlayerItem Transition

AVQueuePlayer

AVPlayerItem (1) and (2)

Item 1

Network Activities

Network Activities
Reduce Startup Time—AVPlayerItem Transition

AVQueuePlayer

AVPlayerItem (1) and (2)

AVPlayerItem.timeControlStatus == .playing

Item 1

Item 2

Network Activities

Network Activities
Reduce Startup Time—AVPlayerItem Transition

Use AVQueuePlayer to play multiple items
Enqueue second AVPlayerItem well in advance
What Determines Network Buffering Time?
What Determines Network Buffering Time?

Choice of variant
What Determines Network Buffering Time?

Choice of variant

Content bitrate
What Determines Network Buffering Time?

Choice of variant
Content bitrate
Playlist target duration
What Determines Network Buffering Time?

Choice of variant
Content bitrate
Playlist target duration
Network bandwidth
Network Buffering Time—Typical Case

#EXTM3U
#EXT-X-VERSION:7
#EXT-X-STREAM-INF:AVERAGE-BANDWIDTH=4972158,BANDWIDTH=7365128,CODECS="avc1.640028,mp4a.40.2",RESOLUTION=1920x1080
video_high.m3u8
Network Buffering Time—Typical Case

Network bandwidth: avg 6Mbps
Target duration: 10s
Initial tier: avg 4972158
Video quality: 1080p
Network Buffering Time—Typical Case

#EXTM3U
#EXT-X-VERSION:7
#EXT-X-STREAM-INF:AVERAGE-BANDWIDTH=4972158,BANDWIDTH=7365128,CODECS="avc1.640028,mp4a.40.2",RESOLUTION=1920x1080
video_high.m3u8

Network bandwidth: avg 6Mbps
Target duration: 10s
Initial tier: avg 4972158
Video quality: 1080p

AVPlayerItem.isPlaybackLikelyToKeepUp == true

8s
10s
Network Buffering Time—Download Slower Than 1x

#EXTM3U
#EXT-X-VERSION:7
#EXT-X-STREAM-INF:AVERAGE-BANDWIDTH=4972158,BANDWIDTH=7365128,CODECS="avc1.640028,mp4a.40.2",RESOLUTION=1920x1080
video_high.m3u8

Network bandwidth: avg 6Mbps
Target duration: 10s
Initial tier: avg 4972158
Video quality: 1080p
Network Buffering Time—Download Slower Than 1x

Network bandwidth: avg 6Mbps
Target duration: 10s
Initial tier: avg 4972158
Video quality: 1080p

AVPlayerItem.isPlaybackLikelyToKeepUp == true
Network Buffering Time—Download Slower Than 1x

Network bandwidth: avg 6Mbps
Target duration: 10s
Initial tier: avg 4972158
Video quality: 1080p

```
#EXTM3U
#EXT-X-VERSION:7
#EXT-X-STREAM-INF:AVERAGE-BANDWIDTH=4972158,BANDWIDTH=7365128,CODECS="avc1.640028,mp4a.40.2",RESOLUTION=1920x1080
video_high.m3u8

#EXT-X-STREAM-INF:AVERAGE-BANDWIDTH=2884141,BANDWIDTH=4125794,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=1280x720
video_mid.m3u8
```

AVPlayerItem.isPlaybackLikelyToKeepUp == true
Network Buffering Time—Download Slower Than 1x

#EXTM3U
#EXT-X-VERSION:7
#EXT-X-STREAM-INF:AVERAGE-BANDWIDTH=4972158,BANDWIDTH=7365128,CODECS="avc1.640028,mp4a.40.2",RESOLUTION=1920x1080
video_high.m3u8

#EXT-X-STREAM-INF:AVERAGE-BANDWIDTH=2884141,BANDWIDTH=4125794,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=1280x720
video_mid.m3u8

AVPlayerItem.isPlaybackLikelyToKeepUp == true

Network bandwidth: avg 6Mbps
Target duration: 10s
Initial tier: avg 4972158
Video quality: 1080p
Network Buffering Time—Initial Variant Choice

#EXTM3U
#EXT-X-VERSION:7
#EXT-X-STREAM-INF:AVERAGE-BANDWIDTH=2884141,BANDWIDTH=4125794,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=1280x720
video_mid.m3u8
#EXT-X-STREAM-INF:AVERAGE-BANDWIDTH=4972158,BANDWIDTH=7365128,CODECS="avc1.640028,mp4a.40.2",RESOLUTION=1920x1080
video_high.m3u8
Network Buffering Time—Initial Variant Choice

#EXTM3U
#EXT-X-VERSION:7
#EXT-X-STREAM-INF:AVERAGE-BANDWIDTH=2884141,BANDWIDTH=4125794,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=1280x720
video_mid.m3u8

#EXT-X-STREAM-INF:AVERAGE-BANDWIDTH=4972158,BANDWIDTH=7365128,CODECS="avc1.640028,mp4a.40.2",RESOLUTION=1920x1080
video_high.m3u8

Network bandwidth: avg 6Mbps
Target duration: 10s
Initial tier: avg 2884141
Video quality: 720p
Network Buffering Time—Initial Variant Choice

Network bandwidth: avg 6Mbps
Target duration: 10s
Initial tier: avg 2884141
Video quality: 720p

AVPlayerItem.isPlaybackLikelyToKeepUp == true

10s

4.6s
Network Buffering Time—Initial Variant Choice

Network bandwidth: avg 6Mbps
Target duration: 10s
Initial tier: avg 2884141
Video quality: 720p
Reducing Network Buffering Time
Reducing Network Buffering Time

Wise choice of initial variant

• Lower content bitrate means shorter buffering time for startup
• But it is a tradeoff of initial video quality
Reducing Network Buffering Time

Wise choice of initial variant
• Lower content bitrate means shorter buffering time for startup
• But it is a tradeoff of initial video quality

Be mindful of all media formats
• Consider initial variant for each flavor of media format
• HDR / SDR, HEVC / H.264, Stereo / DD / DD+
Why Does the Playback Stall?
Why Does the Playback Stall?

Stalls can happen!
Why Does the Playback Stall?

Stalls can happen!

How to investigate stalls
Why Does the Playback Stall?

Stalls can happen!

How to investigate stalls

How to avoid stalls
Stall Investigation
Stall Investigation

Listen to stall notifications
Listen to stall notifications

```swift
notificationCenter.addObserver(self,
    selector: #selector(handlePlaybackStalled(_:)),
    name: .AVPlayerItemPlaybackStalled,
    object: playerItem)
```
Stall Investigation

Listen to stall notifications

```swift
notificationCenter.addObserver(self,
    selector: #selector(handlePlaybackStalled(_:)),
    name: .AVPlayerItemPlaybackStalled,
    object: playerItem)
```

Check AVPlayerItem status
Listen to stall notifications

```swift
notificationCenter.addObserver(
    self,
    selector: #selector(handlePlaybackStalled(_:)),
    name: .AVPlayerItemPlaybackStalled,
    object: playerItem)
```

Check AVPlayerItem status

```swift
AVPlayerItem.isPlaybackLikelyToKeepUp
```
Stall Investigation

Listen to stall notifications

```swift
notificationCenter.addObserver(self,
    selector: #selector(handlePlaybackStalled(_:)),
    name: .AVPlayerItemPlaybackStalled,
    object: playerItem)
```

Check AVPlayerItem status

```swift
AVPlayerItem.isPlayingLikelyToKeepUp
```

AVPlayerItemErrorLog
Stall Investigation

Listen to stall notifications

```
notificationCenter.addObserver(self,
    selector: #selector(handlePlaybackStalled(_:)),
    name: .AVPlayerItemPlaybackStalled,
    object: playerItem)
```

Check AVPlayerItem status

```
AVPlayerItem.isPlaybackLikelyToKeepUp
```

AVPlayerItemErrorLog

AVPlayerItemAccessLog
AVPlayerItemErrorLog

AVPlayerItemErrorLogEvent

URI = http://example.com/VOD/4k/prog_index.m3u8
errorStatusCode = -16830
errorDomain = CoreMediaErrorDomain
errorComment = "Media file not received in 15s"
Stall Example—Network Bandwidth Condition

AVPlayerItemErrorLogEvent
- URI = http://example.com/VOD/4k/prog_index.m3u8
- errorStatusCode = -16830
- errorDomain = CoreMediaErrorDomain
- errorComment = "Media file not received in 15s"

AVPlayerItemAccessLogEvent
- playbackType = VOD
- URI = http://example.com/VOD/4k/prog_index.m3u8
- indicatedBitrate = 36 Mbps
- observedBitrate = 2.8 Mbps
Reducing Stalls—Dealing With Variable Networks
Reducing Stalls—Dealing With Variable Networks

Provide a full set of bit rate tiers
Reducing Stalls—Dealing With Variable Networks

Provide a full set of bit rate tiers

• Each codec combination needs its own set of tiers
AVPlayerItemErrorLog

AVPlayerItemErrorLogEvent

URI = http://example.com/LIVE/prog_index.m3u8
errorStatuscode = -12888
errorDomain = CoreMediaErrorDomain
errorComment = “Playlist File unchanged for 2 consecutive reads”
Stall Example—Content Authoring / Delivering

AVPlayerItemErrorLog

AVPlayerItemErrorLogEvent
- URI: http://example.com/LIVE/prog_index.m3u8
- errorStatusCode: -12888
- errorDomain: CoreMediaErrorDomain
- errorComment: “Playlist File unchanged for 2 consecutive reads”

AVPlayerItemAccessLog

AVPlayerItemAccessLogEvent
- playbackType: LIVE
- URI: http://example.com/LIVE/prog_index.m3u8
- indicatedBitrate: 404 kbps
- observedBitrate: 3.7 Mbps
Reducing Stalls—Content Delivery
Reducing Stalls—Content Delivery

Your content server and CDN must
Reducing Stalls—Content Delivery

Your content server and CDN must

- Deliver media playlists, segments, and keys without delay
Reducing Stalls—Content Delivery

Your content server and CDN must

- Deliver media playlists, segments, and keys without delay
- Update live playlists at least every target duration
Reducing Stalls—Content Delivery

Your content server and CDN must

- Deliver media playlists, segments, and keys without delay
- Update live playlists at least every target duration
- Synchronize discontinuity sequence numbers between playlists
Reducing Stalls—Content Delivery

Your content server and CDN must
- Deliver media playlists, segments, and keys without delay
- Update live playlists at least every target duration
- Synchronize discontinuity sequence numbers between playlists
- Indicate server-side failures clearly
How to Investigate Errors
How to Investigate Errors

Error and access logs
How to Investigate Errors

Error and access logs

Error properties
How to Investigate Errors

Error and access logs

Error properties

Media validation
Streaming Errors—AVPlayerItemErrorLog
Streaming Errors—AVPlayerItemErrorLog

Can be triggered by problems with the network or content format
Streaming Errors—AVPlayerItemErrorLog

Can be triggered by problems with the network or content format

May be nonfatal
Streaming Errors—AVPlayerItemErrorLog

Can be triggered by problems with the network or content format

May be nonfatal

Details are in the error comments

AVPlayerItemErrorLog

AVPlayerItemErrorLogEvent

errorComment = "Media file not received in 15s"
Streaming Errors—AVPlayerItemErrorLog

Can be triggered by problems with the network or content format

May be nonfatal

Details are in the error comments

```
errorComment = "HTTP 404: File Not Found"
```
Streaming Errors—AVPlayerItemErrorLog

Can be triggered by problems with the network or content format

May be nonfatal

Details are in the error comments
Streaming Errors—AVPlayerItemErrorLog

Can be triggered by problems with the network or content format

May be nonfatal

Details are in the error comments

- AVPlayerItemErrorLog
  - AVPlayerItemErrorLogEvent
    - errorComment = "Media file not received in 15s"
    - errorComment = "HTTP 404: File Not Found"
    - errorComment = "Segment exceeds specified bandwidth for variant"
    - errorComment = "unsupported crypt format"
Streaming Errors—AVPlayerItemErrorLog

Can be triggered by problems with the network or content format

May be nonfatal

Details are in the error comments

```swift
AVPlayerItemErrorLogEvent
errorComment = "Media file not received in 15s"

AVPlayerItemErrorLogEvent
errorComment = "HTTP 404: File Not Found"

AVPlayerItemErrorLogEvent
errorComment = "Segment exceeds specified bandwidth for variant"

AVPlayerItemErrorLogEvent
errorComment = "unsupported crypt format"
```
Streaming Errors—AVPlayer, AVPlayerItem
Streaming Errors—AVPlayer, AVPlayerItem

Observe AVPlayerItem.status and AVPlayerItem.error
Streaming Errors—AVPlayer, AVPlayerItem

Observe AVPlayerItem.status and AVPlayerItem.error

AVPlayerItem errors end playback, remove item from player queue
/Streaming Errors : AVPlayer, AVPlayerItem

playerItemObserver = playerItem?.observe(
    AVPlayerItem.status, options: [.new, .initial])
{ [weak self] (item, _) in
    guard let strongSelf = self else { return }
    if item.status == .failed {
        let error = item.error
        print("AVPlayerItem Error: \(String(describing: error?.localizedDescription))")
        let errorLog = item.errorLog()
        let lastErrorEvent = errorLog?.events.last
        print("ErrorLog: \(String(describing: lastErrorEvent?.description))")
    }
}
//Streaming Errors : AVPlayer, AVPlayerItem

playerItemObserver = playerItem?.observe(.AVPlayerItem.status, options: [.new, .initial])
{
    [weak self] (item, _) in

        guard let strongSelf = self else { return }

        if item.status == .failed {

            let error = item.error
            print("AVPlayerItem Error: \(String(describing: error?.localizedDescription))")

            let errorLog = item.errorLog()

            let lastErrorEvent = errorLog?.events.last
            print("ErrorLog: \(String(describing: lastErrorEvent?.description))")

        }

    }
}
//Streaming Errors : AVPlayer, AVPlayerItem

playerItemObserver = playerItem?.observe(AVPlayerItem.status, options: [.new, .initial])
{ [weak self] (item, _) in
    guard let strongSelf = self else { return }
    if item.status == .failed {
        let error = item.error
        print("AVPlayerItem Error: \(String(describing: error?.localizedDescription))")
        let errorLog = item.errorLog()
        let lastErrorEvent = errorLog?.events.last
        print("ErrorLog: \(String(describing: lastErrorEvent?.description))")
    }
}
/* Streaming Errors : AVPlayer, AVPlayerItem */

playerItemObserver = playerItem?.observe(
    AVPlayerItem.status, options: [.new, .initial])
{
    [weak self] (item, _) in
        guard let strongSelf = self else { return }
        if item.status == .failed {
            let error = item.error
            print(“AVPlayerItem Error: \(String(describing: error?.localizedDescription))”)
            let errorLog = item.errorLog()
            let lastErrorEvent = errorLog?.events.last
            print(“ErrorLog: \(String(describing: lastErrorEvent?.description))”)
        }
}
// Streaming Errors: AVPlayer, AVPlayerItem

playerItemObserver = playerItem?.observe(AVPlayerItem.status, options: [.new, .initial])

{ [weak self] (item, _) in

    guard let strongSelf = self else { return }

    if item.status == .failed {

        let error = item.error
        print("AVPlayerItem Error: \(String(describing: error?.localizedDescription))")
        let errorLog = item.errorLog()
        let lastErrorEvent = errorLog?.events.last
        print("ErrorLog: \(String(describing: lastErrorEvent?.description))")
    }

}
// Streaming Errors: AVPlayer, AVPlayerItem

Error Domain=AVFoundationErrorDomain Code=-11800 "See -[AVPlayerItem errorLog] for 2 events"
UserInfo= {
    NSLocalizedDescription=The operation could not be completed,
    NSUnderlyingError=0x108342b0 {
        Error Domain=NSOSStatusErrorDomain Code=-12645
    },
    NSDebugDescription=See -[AVPlayerItem errorLog] for 2 events,
    NSLocalizedFailureReason=An unknown error occurred (-12645)
}
//Streaming Errors: AVPlayer, AVPlayerItem

Error Domain=AVFoundationErrorDomain Code=-11800 "See -[AVPlayerItem errorLog] for 2 events"
UserInfo= {
    NSLocalizedDescription=The operation could not be completed,
    NSUnderlyingError=0x108342ba0 {
        Error Domain=NSOSStatusErrorDomain Code=-12645
    },
    NSDebugDescription=See -[AVPlayerItem errorLog] for 2 events,
    NSLocalizedFailureReason=An unknown error occurred (-12645)
}
//Streaming Errors: AVPlayer, AVPlayerItem

Error Domain=AVFoundationErrorDomain Code=-11800 "See -[AVPlayerItem errorLog] for 2 events"
UserInfo= {
    NSLocalizedDescription=The operation could not be completed,
    NSUnderlyingError=0x108342ba0 {
        Error Domain=NSOSStatusErrorDomain Code=-12645
    },
    NSDebugDescription=See -[AVPlayerItem errorLog] for 2 events,
    NSLocalizedFailureReason=An unknown error occurred (-12645)
}
//Streaming Errors: AVPlayerItemErrorLog

ErrorLog: <AVPlayerItemErrorLog: 0x10832d370>
#Version: 1.0
#Software: AppleCoreMedia/1.0.0.15F79 (iPhone; U; CPU OS 11_0 like Mac OS X; en_us)
#Date: 2018/05/23 11:34:34.034
#Fields: date time uri status domain comment cs-iftype
2018/05/23 11:34:34.034 http://example.com/master.m3u8 -12885
"CoreMediaErrorDomain" "unsupported crypt format." wifi-infra
ErrorLog: <AVPlayerItemErrorLog: 0x10832d370>
#Version: 1.0
#Software: AppleCoreMedia/1.0.0.15F79 (iPhone; U; CPU OS 11_0 like Mac OS X; en_us)
#Date: 2018/05/23 11:34:34.034
#Fields: date time uri status domain comment cs-iftype
2018/05/23 11:34:34.034 http://example.com/master.m3u8 -12885
"CoreMediaErrorDomain" "unsupported crypt format." wifi-infra
//Streaming Errors: AVPlayerItemErrorLog

ErrorLog: <AVPlayerItemErrorLog: 0x10832d370>
#Version: 1.0
#Software: AppleCoreMedia/1.0.0.15F79 (iPhone; U; CPU OS 11_0 like Mac OS X; en_us)
#Date: 2018/05/23 11:34:34.034
#Fields: date time uri status domain comment cs-iftype
2018/05/23 11:34:34.034 http://example.com/master.m3u8 -12885
"CoreMediaErrorDomain" "unsupported crypt format." wifi-infra
//Streaming Errors: AVPlayerItemErrorLog

ErrorLog: <AVPlayerItemErrorLog: 0x10832d370>
#Version: 1.0
#Software: AppleCoreMedia/1.0.0.15F79 (iPhone; U; CPU OS 11_0 like Mac OS X; en_us)
#Date: 2018/05/23 11:34:34.034
#Fields: date time uri status domain comment cs-iftype
2018/05/23 11:34:34.034 http://example.com/master.m3u8 -12885
"CoreMediaErrorDomain" "unsupported crypt format." wifi-infra
Streaming Errors—HDCP
Streaming Errors—HDCP

`AVPlayer.isOutputObscuredDueToInsufficientExternalProtection == true`
Streaming Errors—HDCP

Current item requires external protection
Device does not meet protection level
User observes video loss

AVPlayer.isOutputObscuredDueToInsufficientExternalProtection == true
Streaming Errors—HDCP

Current item requires external protection
Device does not meet protection level
User observes video loss

Have a variant which doesn’t require HDCP protection
App user interface should reflect property value changes
Streaming Errors—HDCP

AVPlayer.isOutputObscuredDueToInsufficientExternalProtection == true

Current item requires external protection
Device does not meet protection level
User observes video loss

Have a variant which doesn’t require HDCP protection
App user interface should reflect property value changes
Streaming Errors—Media Validation
Streaming Errors—Media Validation

Media stream validator
Streaming Errors—Media Validation

Media stream validator

Available on developer website
Master Playlists

Getting it right
“The right thing to do is to give us everything you’ve got.”

Roger Pantos, in a meeting
How do I make sure the player chooses the right stream?
# The simplest possible master playlist. Don't do this!

```extm3u
#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=1500000
http://example.com/video.m3u8
```
# Slightly better master playlist. Still, don't do this!

#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=1500000
http://example.com/video1.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=2500000
http://example.com/video2.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=3500000
http://example.com/video3.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=6500000
http://example.com/video4.m3u8
## Attributes For Variants

**EXT-X-STREAM-INF tag**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>QoM Question</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BANDWIDTH</strong></td>
<td>Peak bits per second</td>
<td>Worst case data rate</td>
</tr>
<tr>
<td><strong>AVERAGE-BANDWIDTH</strong></td>
<td>Average bits per second</td>
<td>Expected data rate</td>
</tr>
<tr>
<td><strong>CODECS</strong></td>
<td>Media formats used</td>
<td>Will I be able to decode this?</td>
</tr>
<tr>
<td><strong>RESOLUTION</strong></td>
<td>Video resolution</td>
<td>Is this an appropriate size?</td>
</tr>
<tr>
<td><strong>HDCP-LEVEL</strong></td>
<td>NONE, TYPE-0, or TYPE-1</td>
<td>Will this play on my output device?</td>
</tr>
<tr>
<td><strong>VIDEO-RANGE</strong></td>
<td>SDR or PQ</td>
<td>Will this play on my output device?</td>
</tr>
<tr>
<td><strong>FRAME-RATE</strong></td>
<td>Maximum frame rate</td>
<td>Will this play on my output device?</td>
</tr>
</tbody>
</table>
# Simple master playlist with attributes

```m3u8
#EXTM3U
#EXT-X-VERSION:4
#EXT-X-STREAM-INF:BANDWIDTH=1500000,AVERAGE-BANDWIDTH=1000000,CODECS="avc1.64001f, mp4a.40.2",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976
http://example.com/video1.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=2500000,AVERAGE-BANDWIDTH=2000000,CODECS="avc1.64001f, mp4a.40.2",RESOLUTION=960x540,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976
http://example.com/video2.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=3500000,AVERAGE-BANDWIDTH=3000000,CODECS="avc1.64001f, mp4a.40.2",RESOLUTION=1280x720,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976
http://example.com/video3.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=6500000,AVERAGE-BANDWIDTH=6000000,CODECS="avc1.64001f, mp4a.40.2",RESOLUTION=1920x1080,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976
http://example.com/video4.m3u8
```
Why don't I see images in fast-forward or in the Apple TV scrubber bar?
What's an I-frame?

Normal (full) video segments

segment 1
segment 2
segment 3
What's an I-frame?

Independently decodable frames
I-frame Video Segments

Consist of a single frame
High Density I-frames
Evenly spaced, closer together, better user experience
# Master Playlist with I-frame playlists

```m3u8
#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=1500000,AVERAGE-BANDWIDTH=1000000,CODECS="avc1.64001f, mp4a.40.2",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976
http://example.com/video1.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=2500000,AVERAGE-BANDWIDTH=2000000,CODECS="avc1.64001f, mp4a.40.2",RESOLUTION=960x540,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976
http://example.com/video2.m3u8

#EXT-X-I-FRAME-STREAM-INF:BANDWIDTH=187500,AVERAGE-BANDWIDTH=125000,CODECS="avc1.64001f", RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,URI="http://example.com/iframe1.m3u8"

#EXT-X-I-FRAME-STREAM-INF:BANDWIDTH=312500,AVERAGE-BANDWIDTH=250000,CODECS="avc1.64001f", RESOLUTION=960x540,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,URI="http://example.com/iframe2.m3u8"
```
# Master Playlist with I-frame playlists

# EXT M3U

# EXT-X-STREAM-INF:BANDWIDTH=1500000,AVERAGE-BANDWIDTH=1000000,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976
http://example.com/video1.m3u8

# EXT-X-STREAM-INF:BANDWIDTH=2500000,AVERAGE-BANDWIDTH=2000000,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=960x540,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976
http://example.com/video2.m3u8

# EXT-X-I-FRAME-STREAM-INF:BANDWIDTH=187500,AVERAGE-BANDWIDTH=125000,CODECS="avc1.64001f",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,URI="http://example.com/iframe1.m3u8"

# EXT-X-I-FRAME-STREAM-INF:BANDWIDTH=312500,AVERAGE-BANDWIDTH=250000,CODECS="avc1.64001f",RESOLUTION=960x540,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,URI="http://example.com/iframe2.m3u8"
# Master Playlist with I-frame playlists

EXTM3U

EXT-X-STREAM-INF:BANDWIDTH=1500000,AVERAGE-BANDWIDTH=1000000,CODECS="avc1.64001f, mp4a.40.2",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976
http://example.com/video1.m3u8

EXT-X-STREAM-INF:BANDWIDTH=2500000,AVERAGE-BANDWIDTH=2000000,CODECS="avc1.64001f, mp4a.40.2",RESOLUTION=960x540,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976
http://example.com/video2.m3u8

EXT-X-I-FRAME-STREAM-INF:BANDWIDTH=187500,AVERAGE-BANDWIDTH=125000,CODECS="avc1.64001f", RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,URI="http://example.com/iframe1.m3u8"

EXT-X-I-FRAME-STREAM-INF:BANDWIDTH=312500,AVERAGE-BANDWIDTH=250000,CODECS="avc1.64001f", RESOLUTION=960x540,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,URI="http://example.com/iframe2.m3u8"
“The peak segment bit rate of a Media Playlist is the largest bit rate of any contiguous set of segments whose total duration is between 0.5 and 1.5 times the target duration.”
How do I support multiple languages?
<table>
<thead>
<tr>
<th>LANGUAGE</th>
<th>SOUND</th>
<th>SPEAKER</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>✓ Full Dynamic Range</td>
<td></td>
</tr>
<tr>
<td>English AD</td>
<td>Reduce Loud Sounds</td>
<td></td>
</tr>
<tr>
<td>French (Canada)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spanish (Latin America)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tvOS**

**iOS**

---

**Audio & Subtitles**

- **Audio**
  - English (U.S.)
  - English (U.S.) AD
  - Spanish (Latin America)
# Master Playlist with audio in two languages

#EXTM3U

#EXT-X-STREAM-INF:BANDWIDTH=1500000,AVERAGE-BANDWIDTH=1000000,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="aud"
http://example.com/video1.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=2500000,AVERAGE-BANDWIDTH=2000000,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=960x540,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="aud"
http://example.com/video2.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=3500000,AVERAGE-BANDWIDTH=3000000,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=1280x720,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="aud"
http://example.com/video3.m3u8

#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="aud",NAME="English",LANGUAGE="en",DEFAULT=YES,AUTOSELECT=YES,CHANNELS="2",URI="http://example.com/en_audio.m3u8"

#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="aud",NAME="Français",LANGUAGE="fr",DEFAULT=NO,AUTOSELECT=YES,CHANNELS="2",URI="http://example.com/fr_audio.m3u8"
# Master Playlist with audio in two languages

#EXTM3U

#EXT-X-STREAM-INF:BANDWIDTH=1500000,AVERAGE-BANDWIDTH=1000000,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="aud"
http://example.com/video1.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=2500000,AVERAGE-BANDWIDTH=2000000,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=960x540,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="aud"
http://example.com/video2.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=3500000,AVERAGE-BANDWIDTH=3000000,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=1280x720,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="aud"
http://example.com/video3.m3u8

#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="aud",NAME="English",LANGUAGE="en",DEFAULT=YES,AUTOSELECT=YES,CHANNELS="2",URI="http://example.com/en_audio.m3u8"

#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="aud",NAME="Français",LANGUAGE="fr",DEFAULT=NO,AUTOSELECT=YES,CHANNELS="2",URI="http://example.com/fr_audio.m3u8"
# Master Playlist with audio in two languages

#EXTM3U

#EXT-X-STREAM-INF:BANDWIDTH=1500000,AVERAGE-BANDWIDTH=1000000,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="aud" http://example.com/video1.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=2500000,AVERAGE-BANDWIDTH=2000000,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=960x540,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="aud" http://example.com/video2.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=3500000,AVERAGE-BANDWIDTH=3000000,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=1280x720,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="aud" http://example.com/video3.m3u8

#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="aud",NAME="English",LANGUAGE="en",DEFAULT=YES,AUTOSELECT=YES,CHANNELS="2",URI="http://example.com/en_audio.m3u8"

#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="aud",NAME="Français",LANGUAGE="fr",DEFAULT=NO,AUTOSELECT=YES,CHANNELS="2",URI="http://example.com/fr_audio.m3u8"
What's the difference between DEFAULT and AUTOSELECT?
This rendition can be automatically selected

Generally you want this set to YES

Set to NO for things like a commentary track
DEFAULT

Player chooses this rendition unless the user chooses something else

Only one member of a group can be DEFAULT=YES

Must have AUTOSELECT=YES

Has nothing to do with the default variant
How do I support multichannel audio?
## Multichannel Audio

Gives us stereo also

<table>
<thead>
<tr>
<th></th>
<th>Group 1</th>
<th>Group 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>English stereo</td>
<td>English 5.1</td>
</tr>
<tr>
<td>French</td>
<td>French stereo</td>
<td>French 5.1</td>
</tr>
<tr>
<td>Chinese</td>
<td>Chinese stereo</td>
<td>Chinese 5.1</td>
</tr>
</tbody>
</table>
Multichannel Audio
Gives us stereo also

<table>
<thead>
<tr>
<th>Language</th>
<th>Group 1</th>
<th>Group 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>English stereo</td>
<td>English 5.1</td>
</tr>
<tr>
<td>French</td>
<td>French stereo</td>
<td></td>
</tr>
<tr>
<td>Chinese</td>
<td>Chinese stereo</td>
<td>Chinese 5.1</td>
</tr>
</tbody>
</table>
# Multichannel Audio

Gives us stereo also

<table>
<thead>
<tr>
<th></th>
<th>Group 1</th>
<th>Group 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>English stereo</td>
<td>English 5.1</td>
</tr>
<tr>
<td>French</td>
<td>French stereo</td>
<td>French Stereo</td>
</tr>
<tr>
<td>Chinese</td>
<td>Chinese stereo</td>
<td>Chinese 5.1</td>
</tr>
</tbody>
</table>
# Master Playlist with audio in two languages, two formats

#EXTM3U

#EXT-X-STREAM-INF:BANDWIDTH=1500000,AVERAGE-BANDWIDTH=1000000,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="aac"
http://example.com/video1.m3u8

#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="aac",NAME="English",LANGUAGE="en",DEFAULT=YES,AUTOSELECT=YES,CHANNELS="2",URI="http://example.com/en_audio.m3u8"

#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="aac",NAME="Français",LANGUAGE="fr",DEFAULT=NO,AUTOSELECT=YES,CHANNELS="2",URI="http://example.com/fr_audio.m3u8"
# Master Playlist with audio in two languages, two formats

```m3u8
#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=1500000,AVERAGE-BANDWIDTH=1000000,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="aac"
http://example.com/video1.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=1700000,AVERAGE-BANDWIDTH=1200000,
CODECS="avc1.64001f,ac-3,mp4a.40.2",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,
FRAME-RATE=23.976,AUDIO="ac3"
http://example.com/video1.m3u8
#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="aac",NAME="English",LANGUAGE="en",DEFAULT=YES,
AUTOSELECT=YES,CHANNELS="2",URI="http://example.com/en_audio.m3u8"
#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="aac",NAME="Français",LANGUAGE="fr",DEFAULT=NO,
AUTOSELECT=YES,CHANNELS="2",URI="http://example.com/fr_audio.m3u8"
#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="ac3",NAME="English",LANGUAGE="en",DEFAULT=YES,
AUTOSELECT=YES,CHANNELS="6",URI="http://example.com/en_surround.m3u8"
#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="ac3",NAME="Français",LANGUAGE="fr",DEFAULT=NO,
AUTOSELECT=YES,CHANNELS="2",URI="http://example.com/fr_audio.m3u8"
```
# Master Playlist with audio in two languages, two formats

#EXTM3U

#EXT-X-STREAM-INF:BANDWIDTH=1500000,AVERAGE-BANDWIDTH=1000000,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="aac"
http://example.com/video1.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=1700000,AVERAGE-BANDWIDTH=1200000,
CODECS="avc1.64001f,ac-3,mp4a.40.2",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,
FRAME-RATE=23.976,AUDIO="ac3"
http://example.com/video1.m3u8

#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="aac",NAME="English",LANGUAGE="en",DEFAULT=YES,
AUTOSELECT=YES,CHANNELS="2",URI="http://example.com/en_audio.m3u8"

#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="aac",NAME="Français",LANGUAGE="fr",DEFAULT=NO,
AUTOSELECT=YES,CHANNELS="2",URI="http://example.com/fr_audio.m3u8"

#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="ac3",NAME="English",LANGUAGE="en",DEFAULT=YES,
AUTOSELECT=YES,CHANNELS="6",URI="http://example.com/en_surround.m3u8"

#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="ac3",NAME="Français",LANGUAGE="fr",DEFAULT=NO,
AUTOSELECT=YES,CHANNELS="2",URI="http://example.com/fr_audio.m3u8"
How do I support several audio bit rates?
## Multiple Audio Bit Rates

Give us every language in every bit rate

<table>
<thead>
<tr>
<th>Language</th>
<th>HE-AAC 32 kb</th>
<th>AAC-LC 96 kb</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>English low bit rate stereo</td>
<td>English stereo</td>
</tr>
<tr>
<td>French</td>
<td>French low bit rate stereo</td>
<td>French stereo</td>
</tr>
<tr>
<td>Chinese</td>
<td>Chinese low bit rate stereo</td>
<td>Chinese stereo</td>
</tr>
</tbody>
</table>
Multiple Audio Bit Rates and Formats
Always fill out the whole matrix

<table>
<thead>
<tr>
<th>Language</th>
<th>HE-AAC 32 kb</th>
<th>AAC-LC 96 kb</th>
<th>AC-3 384 kb</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>English low bit rate stereo</td>
<td>English stereo</td>
<td>English 5.1</td>
</tr>
<tr>
<td>French</td>
<td>French low bit rate stereo</td>
<td>French stereo</td>
<td>French 5.1</td>
</tr>
<tr>
<td>Chinese</td>
<td>Chinese low bit rate stereo</td>
<td>Chinese stereo</td>
<td>Chinese 5.1</td>
</tr>
</tbody>
</table>
# Master Playlist with multiple bit rate audio

#EXTM3U

#EXT-X-STREAM-INF:BANDWIDTH=1500000,AVERAGE-BANDWIDTH=1000000,CODECS="avc1.64001f,mp4a.40.5",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="low"
http://example.com/video1.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=2500000,AVERAGE-BANDWIDTH=2000000,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=960x540,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="high"
http://example.com/video2.m3u8
# Master Playlist with multiple bit rate audio and multiple formats

#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=1500000,AVERAGE-BANDWIDTH=1000000,CODECS="avc1.64001f,mp4a.40.5",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="low"
http://example.com/video1.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=2500000,AVERAGE-BANDWIDTH=2000000,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=960x540,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="high"
http://example.com/video2.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=1800000,AVERAGE-BANDWIDTH=1300000,CODECS="avc1.64001f,ac-3",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="ac3"
http://example.com/video1.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=2700000,AVERAGE-BANDWIDTH=2200000,CODECS="avc1.64001f,ac-3",RESOLUTION=960x540,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="ac3"
http://example.com/video2.m3u8
# Master Playlist with multiple bit rate audio and multiple formats

```m3u
#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=1500000,AVERAGE-BANDWIDTH=1000000,CODECS="avc1.64001f,mp4a.40.5",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="low" http://example.com/video1.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=2500000,AVERAGE-BANDWIDTH=2000000,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=960x540,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="high" http://example.com/video2.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=1800000,AVERAGE-BANDWIDTH=1300000,CODECS="avc1.64001f,ac-3",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="ac3" http://example.com/video1.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=2700000,AVERAGE-BANDWIDTH=2200000,CODECS="avc1.64001f,ac-3",RESOLUTION=960x540,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="ac3" http://example.com/video2.m3u8
```
How do I want HEVC or Dolby Vision?
## Video Formats

Don't skimp on variants

<table>
<thead>
<tr>
<th>Video (tier 1)</th>
<th>H.264</th>
<th>HEVC</th>
<th>Dolby Vision</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>540p SDR H.264</td>
<td>540p SDR HEVC</td>
<td>540p HDR Dolby Vision</td>
</tr>
<tr>
<td>Video (tier 2)</td>
<td>720p SDR H.264</td>
<td>720p SDR HEVC</td>
<td>720p HDR Dolby Vision</td>
</tr>
<tr>
<td>Video (tier 3)</td>
<td>1080p SDR H.264</td>
<td>1080p SDR HEVC</td>
<td>1080p HDR Dolby Vision</td>
</tr>
<tr>
<td>Video (tier 4)</td>
<td></td>
<td>2160p SDR HEVC</td>
<td>2160p HDR Dolby Vision</td>
</tr>
</tbody>
</table>
# Master Playlist with multiple video formats (part 1 of 3)

#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=2259085,AVERAGE-BANDWIDTH=1477534,CODECS="avc1.64001f,ac-3",
RESOLUTION=960x540,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="audio-ac3"
http://example.com/video_a1.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=4860763,AVERAGE-BANDWIDTH=3175979,CODECS="avc1.640020,ac-3",
RESOLUTION=1280x720,HDCP-LEVEL=TYPE-0,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="audio-ac3"
http://example.com/video_a2.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=6463929,AVERAGE-BANDWIDTH=4165530,CODECS="avc1.640028,ac-3",
RESOLUTION=1920x1080,HDCP-LEVEL=TYPE-0,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="audio-ac3"
http://example.com/video_a3.m3u8
# Master Playlist with multiple video formats (part 2 of 3)

#EXT-X-STREAM-INF:BANDWIDTH=1856750,AVERAGE-BANDWIDTH=1194136,CODECS="hvc1.2.4.L123.B0,ac-3",RESOLUTION=960x540,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="audio-ac3",
http://example.com/video_h1.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=3955841,AVERAGE-BANDWIDTH=2488906,CODECS="hvc1.2.4.L123.B0,ac-3",RESOLUTION=1280x720,HDCP-LEVEL=TYPE-0,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="audio-ac3",
http://example.com/video_h2.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=3954219,AVERAGE-BANDWIDTH=2487757,CODECS="hvc1.2.4.L123.B0,ac-3",RESOLUTION=1920x1080,HDCP-LEVEL=TYPE-0,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="audio-ac3",
http://example.com/video_h3.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=8545569,AVERAGE-BANDWIDTH=5269090,CODECS="hvc1.2.4.H150.B0,ac-3",RESOLUTION=3840x2160,HDCP-LEVEL=TYPE-1,VIDEO-RANGE=SDR,FRAME-RATE=23.976,AUDIO="audio-ac3",
http://example.com/video_h4.m3u8
# Master Playlist with multiple video formats (part 3 of 3)

EXT-X-STREAM-INF:BANDWIDTH=2208349,AVERAGE-BANDWIDTH=1394149,CODECS="dvh1.05.01,ac-3", RESOLUTION=960x540,HDCP-LEVEL=TYPE-1,VIDEO-RANGE=PQ,FRAME-RATE=23.976,AUDIO="audio-ac3", http://example.com/video_d1.m3u8

EXT-X-STREAM-INF:BANDWIDTH=4662645,AVERAGE-BANDWIDTH=2926283,CODECS="dvh1.05.01,ac-3", RESOLUTION=1280x720,HDCP-LEVEL=TYPE-1,VIDEO-RANGE=PQ,FRAME-RATE=23.976,AUDIO="audio-ac3", http://example.com/video_d2.m3u8

EXT-X-STREAM-INF:BANDWIDTH=4659386,AVERAGE-BANDWIDTH=2929051,CODECS="dvh1.05.03,ac-3", RESOLUTION=1920x1080,HDCP-LEVEL=TYPE-1,VIDEO-RANGE=PQ,FRAME-RATE=23.976,AUDIO="audio-ac3", http://example.com/video_d3.m3u8

EXT-X-STREAM-INF:AVERAGE-BANDWIDTH=6327675,BANDWIDTH=10277026,CODECS="dvh1.05.06,ac-3", RESOLUTION=3840x2160,HDCP-LEVEL=TYPE-1,VIDEO-RANGE=PQ,FRAME-RATE=23.976,AUDIO="audio-ac3", http://example.com/video_d4.m3u8
How do I handle subtitles?
# Master Playlist with subtitles and closed captions

```m3u8
#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=1500000,AVERAGE-BANDWIDTH=1000000,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAME-RATE=23.976,SUBTITLES="subtitles",CLOSED-CAPTIONS="cc"
http://example.com/video1.m3u8

#EXT-X-MEDIA:TYPE=SUBTITLES,GROUP-ID="subtitles",LANGUAGE="cmn-Hans",NAME="中文（简体)",AUTOSELECT=YES,FORCED=NO,URI="http://example.com/subtitles_cmn-Hans.m3u8"

#EXT-X-MEDIA:TYPE=SUBTITLES,GROUP-ID="subtitles",LANGUAGE="es-419",NAME="Español (Latinoamérica) (forced)",AUTOSELECT=YES,FORCED=YES,URI="http://example.com/subtitles_es-419_forced.m3u8"

```
# Master Playlist with subtitles and closed captions

#EXTM3U

#EXT-X-STREAM-INF:BANDWIDTH=1500000,AVERAGE-BANDWIDTH=1000000,CODECS="avc1.64001f,mp4a.40.2",RESOLUTION=768x432,HDCP-LEVEL=NONE,VIDEO-RANGE=SDR,FRAMERATE=23.976,SUBTITLES="subtitles",CLOSED-CAPTIONS="cc"

http://example.com/video1.m3u8

#EXT-X-MEDIA:TYPE=SUBTITLES,GROUP-ID="subtitles",LANGUAGE="cmn-Hans",NAME="中文（简体)",AUTOSELECT=YES,FORCED=NO,URI="http://example.com/subtitles_cmn-Hans.m3u8"

#EXT-X-MEDIA:TYPE=SUBTITLES,GROUP-ID="subtitles",LANGUAGE="es-419",NAME="Español (Latinoamérica) (forced)",AUTOSELECT=YES,FORCED=YES,URI="http://example.com/subtitles_es-419_forced.m3u8"

“The right thing to do is to give us everything you’ve got.”

Roger Pantos, in a meeting
Summary

Measure key performance indicators (KPIs)

Reduce startup time

Investigate and resolve stalls and errors

Tell us everything you can in your master playlists
More Information


<table>
<thead>
<tr>
<th>HTTP Live Streaming Lab</th>
<th>Technology Lab 3</th>
<th>Tuesday 2:00PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP Live Streaming Lab</td>
<td>Technology Lab 4</td>
<td>Thursday 9:00AM</td>
</tr>
</tbody>
</table>