

Edgecast

# Streaming for Apple Portable Devices

---

**edgecast**

## Disclaimer

Care was taken in the creation of this guide. However, Edgecast cannot accept any responsibility for errors or omissions. There are no warranties, expressed or implied, including the warranty of merchantability or fitness for a particular purpose, accompanying this product.

## Trademark Information

EDGECAST is a registered trademark of Edgecast Inc.

APPLE, IPHONE, IPAD, IPOD TOUCH, MAC OS X, MEDIA STREAM SEGMENTER, AND QUICKTIME are registered trademarks of Apple Inc.

WOWZA MEDIA SERVER is a registered trademark of Wowza Media Systems.

## About This Guide

Streaming for Apple Portable Devices

Version 1.72

6/2/2022

© 2022 Edgecast Inc. All rights reserved.

# Table of Contents

Streaming & iOS Portable Devices .....	1
Overview .....	1
How Does It Work? .....	1
HTTP Live Streaming (HLS) .....	2
On-Demand Streaming (HTTP Progressive Download).....	4
HLS Requirements.....	5
Encoding.....	6
How to Stream a Live Event to iOS Devices through a Wowza Media Server .....	7
Summary .....	7
Live Streaming Configuration Overview .....	7
Setting Up HTTP Rules Engine.....	8
Wowza Media Server Configuration .....	9
Troubleshooting a Customer Origin/Edge CNAME Configuration.....	10
How to Stream On-Demand Content to iOS Devices.....	12
Introduction .....	12
HTTP Progressive Download .....	12
Encode Video .....	12
Upload Video.....	13
Stream Video.....	13
HTTP Live Streaming .....	14
Encode and Segment Video .....	14
Upload Video.....	14
Stream Video.....	14
Appendix .....	16
References .....	16

# Streaming & iOS Portable Devices

---

## Overview

Live and on-demand audio and/or video can be streamed to an iPhone, iPad, or iPod touch over HTTP for playback on its built-in media player application. This built-in media player application natively supports several video formats, which allows the iOS device to play on-demand content by performing a simple HTTP request. Live events, on the other hand, require additional processing (i.e., stream segmentation). Apple identifies the process through which the built-in media player interprets and plays segmented video as HTTP Live Streaming (HLS). In addition to streaming media, it provides support for media encryption, user authentication over HTTPS, and intelligent dynamic stream selection according to the detected network bandwidth. Intelligent dynamic stream selection allows an iPhone, iPad, or iPod touch to automatically switch between different bit rate streams according to the current speed of your client's connection. Dynamically switching between streams can enhance the user experience while switching or connecting through EDGE, 3G, WiFi, and cellular networks.

---

## How Does It Work?

We provide various solutions through which you can stream live and on-demand content to iOS devices. Each streaming solution supports a different feature set.

A brief description is provided below for each streaming solution.

- **HLS (CDN - HLS):** Leverage HTTP Live Streaming without having to purchase & implement third-party server software. Detailed information on how to implement this solution can be found in the **HTTP Streaming Guide**.
- **HLS (Wowza):** A third-party server software that supports HTTP Live Streaming.
- **HLS (CDN – Segmented Content):** A file segmenter can be used to pre-package on-demand content for iOS consumption. Once this package is hosted on an origin server, iOS devices can stream it dynamically via a link to the M3U8 index file.
- **HTTP Progressive Download:** Stream on-demand content to iOS devices using a standard CDN or edge CNAME URL. The iOS device will stream content using sequential byte-range requests.

The following table describes the iOS streaming capabilities of each solution.

Streaming Solution	CDN Only?	Technology	Live Streaming	On-Demand Streaming	Dynamic Streaming
HTTP Live Streaming	Yes	HLS	●	●	●
Wowza Media Server	Requires Wowza Media Server	HLS	●	●	●
File Segmenter	Requires a File Segmenter	HLS		●	●
HTTP Progressive Download	Yes	HTTP		●	

## HTTP Live Streaming (HLS)

In the previous section, a table indicated the various ways through which the HTTP Live Streaming technology can be implemented. This section will provide a high-level overview on how content is streamed for each solution.

---

**Note:** Keep in mind that HTTP Live Streaming can be used to stream both live and on-demand content.

---

### Live Streaming

The process through which a live event can be streamed to iOS devices is indicated below.

1. A streaming server must transform your live event feed into media chunks and package them into a stream.
  - **HTTP Live Streaming (CDN):** Create an event for the live stream.
  - **Third-party streaming server:** A third-party streaming server (e.g., Wowza) should be configured as a customer origin on the HTTP Large platform.
2. Encode the desired audio/video content to an iOS supported format. A media encoder can perform this process for a live event as it occurs.
  - **HTTP Live Streaming (CDN):** Configure the encoder to publish stream(s) to the publishing point URLs defined for the desired event.
  - **Third-party streaming server:** Configure the encoder to publish streams to your third-party streaming server.

3. A media player interprets the index file and then requests the stream files in the order that they are listed. This allows it to play the streaming video without pauses or gaps between stream segments.
  - **HTTP Live Streaming (CDN):** Configure the media player to request the live stream using an HLS-specific playback URL.
  - **Third-party streaming server:** Configure the media player to request the live stream using an edge CNAME URL that points to the streaming server.

### On-Demand Streaming (CDN & Encoded Media)

We offer an HLS streaming solution that allows our servers to dynamically segment encoded video as it is requested by a client. This process is described below.

1. Encode the desired audio/video content to an iOS supported format.
2. Upload the encoded content to CDN storage.
3. Link to the desired content from a web page. This link should use an HLS-specific URL that points to the HTTP Large platform. Our servers will automatically generate an index file which the client will interpret to request video segments.

---

**Note:** For more information on how to setup on-demand streaming for HTTP Live Streaming, please refer to the **HTTP Live Streaming** chapter in the **HTTP Streaming** guide.

---

## On-Demand Streaming (CDN & Segmented Content)

An alternative method to achieve dynamic streaming for on-demand content is to process your encoded video through a file segmenter and then upload the resulting package to a CDN or customer origin server. This process is described below.

1. Encode the desired audio/video content to an iOS supported format.
2. A file segmenter should be used to break up the encoded video into smaller files. It will also generate an index file that lists the order in which stream files should be played.
3. The files generated by the stream segmenter must then be hosted on a CDN origin server or a customer origin server.
4. Link to the M3U8 index file using a CDN or edge CNAME URL that points to the HTTP Large platform. This allows our CDN to deliver the appropriate index file, along with stream files, to clients that request on-demand audio/video.
5. A media player interprets the index file and then requests the stream files in the order that they are listed. This allows it to play the streaming video without pauses or gaps between stream segments.

## On-Demand Streaming (HTTP Progressive Download)

On-demand content can be streamed to iOS devices via HTTP Progressive Download. HTTP Progressive Download refers to a streaming solution provided on the HTTP Large platform that allows a media player to perform sequential requests for on-demand content.

---

**Note:** This streaming solution does not leverage HTTP Live Streaming, nor does it provide dynamic streaming capability.

---

1. Before audio/video content can be streamed to an iOS device, it must be encoded in an iOS supported format.
2. Upload the encoded content to a CDN or customer origin server.
3. Link to the desired content from a web page. This link should use a CDN or edge CNAME URL that points to the HTTP Large platform.

---

**Note:** For more information on how to use HTTP Progressive Download to stream to iOS devices, please refer to the **HTTP Progressive Download** section in the **How to Stream On-Demand Content to iOS Devices** chapter.

---

---

## HLS Requirements

This section describes the requirements for HTTP Live Streaming.

- **Video Format:** Video should be encoded using one of the following: H.264 Baseline Level 3.0, Baseline Level 3.1, and Main Level 3.1
- **Audio Format:** Audio should be encoded using one of the following:
  - HE-AAC or AAC-LC up to 48 kHz, stereo audio
  - MP3 (MPEG-1 Audio Layer 3) 8 kHz to 48 kHz, stereo audio
- **Segmenter:** A segmenter is required to break up a stream or a file into smaller files. The recommended duration for each chunk is 10 seconds. There are two types of segmenters supported by HTTP Live Streaming. The requirements for each are listed below.
  - **Stream segmenter:** A stream segmenter generates .ts files (MPEG-2 streams) and index files as .M3U8 files from a live stream. Detailed information on the structure of the index file format can be found in the IETF Internet-Draft of HTTP Live Streaming (<http://tools.ietf.org/html/draft-pantos-http-live-streaming-04> ). The live streaming solutions discussed in this guide perform this process upon ingesting a stream.
  - **File segmenter:** A file segmenter must be able to convert an encoded media file into stream segments of a MPEG-2 transport stream. This segmentation is automatically performed when leveraging our HTTP Live Streaming solution or when using Wowza Media Server.
- **CDN:** A live event or on-demand content must be streamed over the HTTP Large platform. Additional CDN configuration information for each streaming solution is provided below.
  - **Live Event (Wowza):** Configure the following CDN settings on the HTTP Large platform:
    - **Customer Origin:** It should point to a Wowza Media Server, the HTTP host header should be set to an edge CNAME, and Origin Shield should be enabled.
    - **Edge CNAME:** It should point to the above customer origin.
    - **Cache Settings:** Standard caching is required for compatibility with a Wowza Media Server.
    - **HTTP Rules Engine:** A rule should be created to override Wowza's default caching behavior for the playlist (i.e., m3u8) and media (i.e., ts) files.



- **On-Demand Content (CDN or Customer Origin Server):** The stream segments and the index file need to be available through the HTTP Live Streaming platform. This can be accomplished by hosting the stream segments and index files on a CDN or customer origin server and then using a CDN URL (e.g., `http://wpc.0001.edgecastcdn.net/800001/Streams/MyStream.m3u8`) or an edge CNAME URL when pointing your clients to the stream.
- **On-Demand Content (HTTP Progressive Download):** Encoded video (i.e., H.264) needs to be uploaded to CDN storage. The client should point to this content using a CDN or edge CNAME URL that corresponds to the desired service. The type of URL used to point to the encoded video determines which streaming solution will be used.
- **Client:** HTTP Live Streaming is only supported by the following clients:
  - **Mobile Devices:** iOS 3.0 and later; However, if you would like to use H.264 Baseline 3.1, then all of your customers will need to use an iPad, iPhone 3G, or an iPod touch (2<sup>nd</sup> generation and later).
  - **Mac OS X:** Version 10.6 and later
    - **QuickTime:** Version X or later

---

## Encoding

The first step in distributing your content to iOS devices is to encode your audio and/or video. The recommended encoding method for HTTP Live Streaming is H.264 video (Baseline Level 3.0) and HE-AAC audio. Additionally, you should encode your video with key frames every 3 seconds. The maximum supported bit rate is 1.6 Mbps. Below you will find a set of recommended bit rates for different stream quality levels.

- **Cellular Fallback:** Audio only or audio with still image, 64 Kbps
- **Low:** 96 Kbps video, 64 Kbps audio (Total bit rate: 160 Kbps)
- **Medium:** 450 Kbps video, 96 Kbps audio (Total bit rate: 546 Kbps)
- **High:** 800 Kbps video, 128 Kbps audio (Total bit rate: 928 Kbps)

---

**Note:** When determining the highest quality level for your stream, keep in mind that the screen resolution for older versions of the iPhone may be significantly less than that of iPhone 4 (i.e., 960-by-640-pixel resolution at 326 ppi).

---

# How to Stream a Live Event to iOS Devices through a Wowza Media Server

---

## Summary

Live content can be streamed by a Wowza Media Server to iOS devices (e.g., iPhone, iPad, Mac OS X, etc.) through the use of our worldwide delivery network. The role of the Wowza Media Server is to transform your video feed into media chunks and then package them as a stream that can be interpreted by an iOS device. After which, our CDN will deliver your stream over the HTTP Large platform in an efficient manner to iOS devices around the world.

---

## Live Streaming Configuration Overview

Using HTTP Live Streaming to stream live content to iOS devices requires the proper configuration of a Wowza Media Server. Additionally, the desired stream must take advantage of our data delivery capabilities, which will require you to configure our CDN to recognize your Wowza Media Server. Each step that you must take to set up HTTP Live Streaming to work with our CDN is listed below.

### Live Streaming Configuration Requirements:

- **Customer Origin:** Your Wowza Media Server must be configured as a customer origin for the HTTP Large platform. The hostname specified for that customer origin must include the port used by your Wowza Media Server, which by default is 1935 (e.g., `wowza.mydomain.com:1935`). The **HTTP Host Header** option will need to be configured to point to the hostname used by an edge CNAME.
- **Origin Shield:** It is highly recommended that you have origin shield enabled and properly configured on the Wowza-specific customer origin. If an origin shield is not protecting your Wowza Media Server, it may get overloaded by the number of requests generated by your streams. If you have not purchased the Origin Shield feature, please contact your CDN account manager.
- **Edge CNAME:** Create an edge CNAME that points to the customer origin for your Wowza Media Server. This step is required to instruct the Wowza Media Server to stream data through our CDN. Make sure to create a CNAME record on your DNS server.

This CNAME record should reference your Wowza Media Server hostname to the corresponding CDN URL (i.e., wpc.xxxx.edgecastcdn.net).

- **HTTP Rules Engine:** It is highly recommended that you create a rule on the HTTP Large platform that will override Wowza's default cache behavior for the playlist and the media files.

---

**Note:** Rules Engine must be purchased separately. However, if you have not used up your complimentary rule, then you may be able to perform this configuration for free. For more information, please contact your CDN account manager.

---

- **Cache Settings:** The HTTP Large platform must be configured to use the default caching mechanism, which is called "standard caching." Cache settings can be viewed from the **Query-String Caching** page.
- **Encoder:** An encoder must be configured to send H.264 video to a Wowza Media Server.
- **Wowza Media Server:** Your Wowza Media Server must be properly configured to chunk and package a video for HTTP Live Streaming, which Wowza Media Systems refers to as Cupertino Streaming.

## Setting Up HTTP Rules Engine

It is important to prevent your HTTP Live Streaming playlist (i.e., m3u8) from being cached on our edge servers or by your clients. Additionally, the media files (i.e., ts) should only be cached on our edge servers to ensure optimal data delivery to your clients. The following table describes how a rule should be configured to provide optimal performance for streaming HLS content over a Wowza Media Server.

Match	Feature(s)
URL Path Extension = m3u8	<b>Bypass Cache</b> <ul style="list-style-type: none"><li>• Enabled</li></ul> This ensures that a player doesn't receive an old version of the playlist.
URL Path Extension = ts	<b>Force Internal Max-Age</b> <ul style="list-style-type: none"><li>• Response Code = 200</li><li>• 360 seconds</li></ul> Allows media files to be cached on our edge servers for six minutes. This allows quicker delivery of that content to your clients.

Match	Feature(s)
	<p data-bbox="656 249 873 279"><b>External Max-Age</b></p> <ul data-bbox="704 302 878 331" style="list-style-type: none"> <li data-bbox="704 302 878 331">• -1 seconds</li> </ul> <p data-bbox="656 354 1247 384">This prevents the player from caching media files.</p> <p data-bbox="656 407 938 436"><b>Ignore Origin No-Cache</b></p> <ul data-bbox="704 459 846 489" style="list-style-type: none"> <li data-bbox="704 459 846 489">• Enabled</li> </ul> <p data-bbox="656 512 1414 575">Allows our servers to ignore no-cache header responses sent by the Wowza Media Server.</p> <p data-bbox="656 598 976 627"><b>Expires Header Treatment</b></p> <ul data-bbox="704 651 846 680" style="list-style-type: none"> <li data-bbox="704 651 846 680">• Remove</li> </ul> <p data-bbox="656 703 1317 732">Ensures that an Expires header is not sent to the player.</p>

---

**Reminder:** Rules Engine must be purchased separately. However, if you have not used up your complimentary rule, then you may be able to perform this configuration for free. For more information, please contact your CDN account manager.

---

## Wowza Media Server Configuration

Setting up a Wowza Media Server requires Wowza Push Publishing AddOn 3.5 or higher. The readme provided with this addon provides detailed instructions on how to leverage the RTMP profile for use with our CDN service.

---

## Troubleshooting a Customer Origin/Edge CNAME Configuration

Before you start troubleshooting a customer origin and/or an edge CNAME, you should make sure that your Wowza Media Server has been properly configured. You can verify this by attempting to view your video feed on your iOS device using the Wowza-specific URL. If the video streams properly, then it is possible that you may have to adjust your customer origin or edge CNAME configuration.

### To test HTTP Live Streaming through Wowza Media Server

From an iOS device, open a browser and type the following in the address bar:

```
http://Wowza_Media_Server_2:1935/Application/Stream/playlist.m3u8
```

---

**Note:** If your stream does not play, then you will need to troubleshoot your Wowza Media Server and/or your encoder. For troubleshooting assistance, please reference the documentation provided by Wowza Media Systems or your encoder's manufacturer.

---

### To test HTTP Live Streaming through the CDN

From an iOS device, open a browser and type the following URL in the address bar:

```
http://Edge_CNAME_Hostname/Application/Stream/playlist.m3u8
```

---

**Note:** If the stream plays through Wowza Media Server but not through the CDN, then you will need to troubleshoot your customer origin and/or edge CNAME configuration.

---

## Troubleshooting Checklist

Use the following checklist to troubleshoot issues with live event streaming.

1. Has it been an hour since you created/modified your customer origin or edge CNAME configuration?
  - **Yes:** Proceed to the next step.
  - **No:** Please wait until an hour has elapsed before attempting to stream a video feed to your iOS devices.
2. Can you use a CDN URL  
(<http://wpc.edgecastcdn.net/80xxx/WowzaCustomerOrigin/Path/ManifestFile.m3u8>) to download the manifest file on the Wowza Media Server? When performing this test, make sure that you do not specify the stream name in the URL. This test is simply designed to verify that our servers can retrieve data from the Wowza Media Server.
  - **Yes:** Proceed to the next step.
  - **No:** If you were unable to download the manifest file, then you should perform the following steps:

- i. Verify that the correct URL was specified. You may do this by replacing "wpc.edgecastcdn.net/80xxx/WowzaCustomerOrigin" from the above URL with the hostname for your Wowza Media Server. Were you able to download the manifest file?
    - **Yes:** Proceed to item ii.
    - **No:** You should check the path to the manifest file and the permissions on your Wowza Media Server. Please make the necessary changes. Once you can download the manifest file using the Wowza hostname in the URL, you should try the CDN URL specified above.
  - ii. It is possible that your customer origin has not been properly configured. Please verify that your customer origin has been configured as described in the **To create a customer origin for your Wowza Media Server** procedure above. After making the necessary changes, please wait an hour before trying again. If no changes were required or it still doesn't work, then make sure that the server hosting Wowza has been configured to allow all of the IP addresses listed on the [Customer Origin](#) page.
3. This step should only be attempted if you were able to download the manifest file using a CDN URL. Can you use an edge CNAME URL ([http://Edge\\_CNAME\\_Hostname/Path/ManifestFile.m3u8](http://Edge_CNAME_Hostname/Path/ManifestFile.m3u8)) to download the manifest file on the Wowza Media Server? When performing this test, make sure that you do not specify the stream name in the URL. This test is simply designed to verify that our servers can retrieve data from the Wowza Media Server.
  - **Yes:** From an iOS device, try streaming your content using the following URL:  
[http://Edge\\_CNAME\\_Hostname/Application/Stream/playlist.m3u8](http://Edge_CNAME_Hostname/Application/Stream/playlist.m3u8)
  - **No:** Please verify that your edge CNAME has been configured as described in the **To set up a CNAME to the Wowza Media Server** procedure above. You should pay particular attention that the CNAME record registered on the DNS server matches the name assigned to your edge CNAME. After making the necessary changes, please wait an hour before trying again.

# How to Stream On-Demand Content to iOS Devices

---

## Introduction

The following streaming solutions can be used to deliver on-demand content to iOS devices:

- **HTTP Progressive Download:** Stream content to iOS devices by simply specifying a CDN or an edge CNAME URL that points to the desired audio/video content.
- **HTTP Live Streaming (Segmented Content):** Stream previously segmented content to iOS devices by specifying a CDN or edge CNAME URL that points to an index file.

This chapter describes how to implement the HTTP Progressive Download and HTTP Live Streaming (Segmented Content) streaming solutions.

---

## HTTP Progressive Download

Streaming on-demand content on an iOS device consists of the following steps:

1. Generate encoded video in an iOS supported format.
2. Upload video content to a customer origin server or a CDN origin server.
3. Provide a link to it.

Additional information on each of these steps is provided below.

### Encode Video

Before you can stream on-demand content, it needs to be in a format that is compatible with iOS devices. iOS devices provide native support for a variety of file formats. For detailed information on the file formats and the compression standards supported by iOS devices, please refer to Apple's documentation. A link to it is provided below.

[http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/MediaLayer/MediaLayer.html#//apple\\_ref/doc/uid/TP40007898-CH9-SW4](http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/MediaLayer/MediaLayer.html#//apple_ref/doc/uid/TP40007898-CH9-SW4)

## Upload Video

Once you have verified that your video complies with the above requirements, you should make sure that it can be accessed via the HTTP Large platform. This will require that you upload the desired content to a customer origin server or a CDN origin server.

## Stream Video

Once the desired content is hosted on our CDN, you are ready to stream on-demand content. The quickest way to do so is to link to the desired content using a CDN or edge CNAME URL. The iOS device will play the video in full screen.

An alternative way to stream on-demand content is to take advantage of iOS support of the HTML 5 video embed tag. A sample HTML tag is provided below.

```
<video width="320" height="240" controls="controls">  
    <source src="http://wpc.xxxx.edgecastcdn.net/yyxxx/Path/FileName.ext" />  
</video>
```

Simply replace the URL defined in the src parameter with the CDN or edge CNAME URL to your on-demand content. If you plan on using a CDN URL, then you will need to replace the following items from the sample URL:

- **xxxx:** This should be replaced with your CDN account number. Your CDN account number can be found in the upper-right hand corner of the MCC.
- **yy:** This should be replaced with the identifier corresponding to the type of location where your stream files can be found. This identifier will either be 00 (CDN origin server) or 80 (customer origin server).
- **Path:** This term should be replaced with the path to the folder where the desired on-demand content can be found. If you are using a customer origin server, then this path will always start with the folder assigned to the customer origin created for the HTTP Large platform.
- **Filename.ext:** This term should be replaced by the filename of the desired video content.

The above video tag must be added to the web page where your stream will play.



---

## HTTP Live Streaming

Similar to live events, on-demand content can be made available to your clients through HTTP Live Streaming. HTTP Live Streaming requires the creation of multiple versions of the desired content at variable bit rates. This allows the iOS device to choose the bit rate that is best suited for the current bandwidth.

### Encode and Segment Video

The first step that you should perform is to encode the desired content into H.264 video (Baseline Level 3.0) and HE-AAC audio. After which, it is ready to be split into video segments using a stream segmenter (e.g., Apple Media Stream Segmenter). Although Apple provides the Apple Media Stream Segmenter with Mac OS X, version 10.6 and later, it is recommended that you download the latest version from the Apple Developer website.

The Apple Media Stream Segmenter can split a stream into MPEG-2 stream segments (TS files). It will also generate an M3U8 index file that identifies the order in which those stream segments should be played. This segmenter is able to receive an MPEG-2 transport stream or a properly encoded video file.

### Upload Video

In order to stream on-demand content to iOS devices, it will need to be accessed via the HTTP Large platform. This means that the corresponding stream segments and the index file generated by the stream segmenter must be hosted on a CDN or customer origin server.

### Stream Video

Once the desired content can be accessed via the CDN, you can stream it by simply providing a link to the playlist. This link must leverage a CDN or an edge CNAME URL. The iOS device will play the desired on-demand content in full screen mode.

Another method through which you can stream this content is to create a media player that points to the index file using a CDN or edge CNAME URL. One way of accomplishing this task is to take advantage of iOS support of the HTML 5 video embed tag. A sample HTML tag is provided below.

```
<video width="320" height="240" controls="controls">
  <source src="http://wpc.xxxx.edgecastcdn.net/yyxxxx/Path/FileName.m3u8" />
</video>
```

In the above tag, you will need to replace the following items:

- **xxxx:** This should be replaced with your CDN account number. Your CDN account number can be found in the upper-right hand corner of the MCC.
- **yy:** This should be replaced with the identifier corresponding to the type of location where your stream files can be found. This identifier will either be 00 (CDN origin server) or 80 (customer origin server).
- **Path:** This term should be replaced with the path to the folder where your index file can be found. If you are using a customer origin server, then this path will always start with the folder assigned to the customer origin created for the HTTP Large platform.
- **Filename:** This term should be replaced with the name of the index file generated for your stream by the stream segmenter.

---

**Note:** If you have already uploaded the index file to a CDN origin server, then you can view the exact URL that you should use from the **File Manager** page in the MCC. Simply navigate to the index file, select it, and then make sure that the **HTTP Large** option is selected. The appropriate URL should appear to the left of the selected option.

---

The above tag must be added to the web page where your stream will play.

# Appendix

---

## References

Portions of this document are based on the information found in the HTTP Live Streaming section of the iPhone OS Reference Library. For more detailed information on the implementation of HTTP Live Streaming, please refer to the following website:

<http://developer.apple.com/iphone/library/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/Introduction/Introduction.html>