Edgecast

Dynamic Cloud Packaging

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

About This Guide

Dynamic Cloud Packaging Administration Guide
Version 1.61
11/19/2021

Table of Contents

Dynamic Cloud Packaging

Introduction

Dynamic Cloud Packaging is designed to:

- Stream live events and on-demand content over the HTTP Large platform.
- Playback live and on-demand content using HLS and MPEG-DASH.
- Ingest live events via RTMP.

Requirements

This section describes the various requirements for streaming a live event and on-demand content through our network.

Requirement	Description
Source	**Live Streaming Only:** Use any encoder that can push an encoded audio/video feed over RTMP. **Flash Media Live Encoder:** It is strongly recommended to encode your audio/video feed as a single bit rate stream. The use of multiple bit rate streams (a.k.a. dynamic streaming or adaptive bit rate streaming) is unsupported and should only be used at your own discretion. Recommended encoders are listed below. • Wirecast • Open Broadcaster Software (OBS) **On-Demand Streaming Only:** Stream on-demand content from the following sources: • An external web server (customer origin server) • CDN storage (CDN origin server)
File Format	**On-Demand Streaming Only:** MPEG-4 (MP4)

Requirement	Description
Video Codec	Video should be encoded using one of the following H.264 codecs: - Baseline Level 3.0 - Baseline Level 3.1 - Main Level 3.1
Audio Codec	Audio should be encoded using the following codecs: - HE-AAC or AAC-LC up to 48 kHz, stereo audio **Note:** An additional plug-in may be required for AAC support on the Windows version of Flash Media Live Encoder.
Content Delivery Network	Dynamic Cloud Packaging is designed to stream live and on-demand content via our global network.
Media Player	**HLS Only:** HLS playback requires a media player to meet one of the following requirements: - iOS 3.0 or later (including iPad and Apple TV) - Any computer with Safari 4.0 or later installed - Roku 3 Limited support for: - Android 4.0 (Ice Cream Sandwich) - Android 4.1+ (Jelly Bean) **MPEG-DASH Only:** Any MPEG-DASH-compatible media player may be used for playback.

Note: For more detailed service information and additional recommendations on HLS, please refer to Apple's documentation on HTTP Live Streaming.

Interaction with the HTTP Large Platform

Dynamic Cloud Packaging leverages the HTTP Large platform to stream live and on-demand content. Specifically, it leverages the HTTP Large platform to achieve efficient worldwide delivery. This allows it to take advantage of the services and features that are available on the HTTP Large platform.

Learn how to:

- Generate Dynamic Cloud Packaging-specific reports.
- Configure Rules Engine to be compatible with Dynamic Cloud Packaging.
- Secure streams.

Reports and Analytics

Traffic generated by the Dynamic Cloud Packaging service is logged under the HTTP Large platform. This report data may be viewed via our Analytics product. Detailed information on how this data is reported for each module is provided below.

Module	Description
Core Reports	Core Reports does not distinguish or segregate Dynamic Cloud Packaging traffic from other HTTP Large traffic. The following reports include data for all HTTP Large traffic: • Traffic Summary • "All Platforms" reports • "HTTP Large" reports
Custom Reports	If Dynamic Cloud Packaging is streamed via an edge CNAME, then a custom report may be generated to view hits and data transferred statistics. **Tip:** Enable the logging of the number of hits and total data transferred for a specific edge CNAME by enabling its **Custom Reports** option.
Real-Time Statistics	Real-Time Statistics does not distinguish or segregate Dynamic Cloud Packaging traffic from other HTTP Large traffic. The following reports include data for all HTTP Large traffic: • Overview dashboard • "HTTP Large Object" dashboard

Module	Description
Advanced Content Analytics	The following HTTP Large reports track usage by origin directory: • By File • By Directory • By Download • By 404 Errors
Edge Performance Analytics	The following HTTP Large reports track usage by origin directory: • URLs • NONE Details Report • CONFIG_NOCACHE Details Report • UNCACHEABLE Details Report • TCP_HIT Details Report • TCP_MISS Details Report • TCP_EXPIRED_HIT Details Report • TCP_EXPIRED_MISS Details Report • TCP_CLIENT_REFRESH_MISS_Details Report • 404 Errors Report • 403 Errors Report • 4xx Errors Report • 504 Errors Report • 502 Errors Report • 5xx Errors Report The Origins (HTTP Large) report tracks usage by origin name. Dynamic Cloud Packaging uses the following labels: • **Live Streaming:** EdgeCast Live DCP Playback (Dynamic Cloud Packaging) • **On-Demand Streaming - CDN Storage:** EdgeCast On-Demand DCP (Dynamic Cloud Packaging) • **On-Demand Streaming - Customer Origin:** *CustomerOrigin* DCP (Dynamic Cloud Packaging)

To track Dynamic Cloud Packaging usage via Custom Reports

1. Create an edge CNAME that points to an origin directory.
2. Set the edge CNAME's **Custom Reports** option to "Enabled."
3. Wait an hour after creating the above edge CNAME.
4. From your DNS service provider, create or update a CNAME record to point the edge CNAME's hostname to the CDN hostname (e.g., wpc.0001.edgecastcdn.net).

To generate a custom report on Dynamic Cloud Packaging traffic

1. Verify the following:
    - An edge CNAME that points to Dynamic Cloud Packaging has been created.
    - The **Custom Reports** option has been enabled on this edge CNAME configuration.
    - Traffic is being served over this edge CNAME.
2. Navigate to the HTTP Large custom report.
3. Statistics on the Dynamic Cloud Packaging-specific edge CNAME may be viewed in the table displayed below the bar graph.

Rules Engine

Rules Engine can impact the functionality of Dynamic Cloud Packaging. For example, a rule that defines a cache policy on all requests will prevent a player from retrieving an updated manifest file. In turn, this will prevent the player from properly streaming the requested media.

Under most circumstances, it is recommended to modify each rule's match conditions (e.g., CDN Origin, Customer Origin, Edge CNAME, etc.) to exclude this service. An exception takes place when securing manifest files (e.g., via the Token Auth feature).

Tips for defining or reviewing rules:

- Most rules that apply to all requests will impact this service. Please carefully review all such rules.
- The best way to ensure that a rule will not apply to this service is to match by origin server or by URL.

Origin Directories

Dynamic Cloud Packaging uses the following origin identifiers:

Type	Origin Directory
Live Streaming – Playback	/24*AN* (Dynamic Cloud Packaging Live Playback Instance)
Live Streaming - Publishing	/20*AN* (Dynamic Cloud Packaging Live Playback Instance) **Note:** Report data is not generated for publishing points.
On-Demand Streaming – CDN Storage	/04*AN* (Dynamic Cloud Packaging CDN Origin)
On-Demand Streaming – Customer Origin	/84*AN*/*CustomerOrigin*

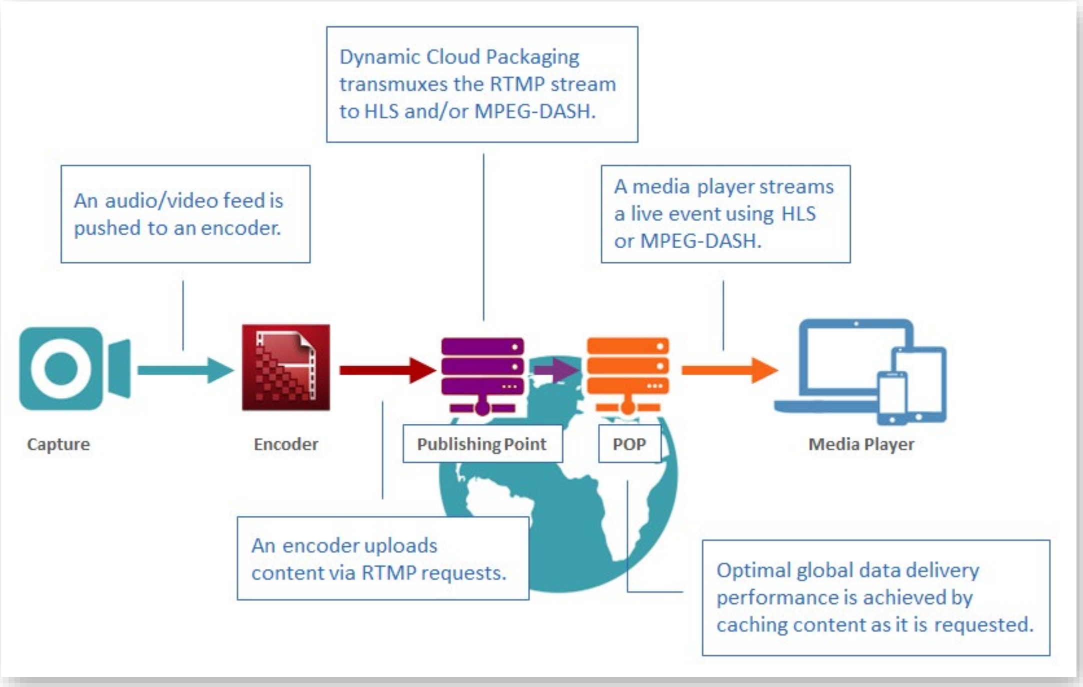
Live Streaming

Introduction

Dynamic Cloud Packaging can stream live content to any standard HLS or MPEG-DASH player. It is able to do so by ingesting a live stream that was pushed by an encoder to our services via RTMP.

How Does It Work?

The entire process through which an audio/video feed is transmuxed and then streamed to clients around the world is illustrated below.


Dynamic Cloud Packaging transmuxes the RTMP stream to HLS and/or MPEG-DASH.
An audio/video feed is pushed to an encoder.
A media player streams a live event using HLS or MPEG-DASH.
Capture
Encoder
Publishing Point
POP
Media Player
An encoder uploads content via RTMP requests.
Optimal global data delivery performance is achieved by caching content as it is requested.

How Does Live Streaming Work?

Transmuxing Workflow

Dynamic Cloud Packaging prepares an audio/video feed for playback through the following workflow:

- **Encoder:** An encoder pushes an encoded audio/video feed via RTMP to the closest publishing point.
- **Dynamic Cloud Packaging:** Dynamic Cloud Packaging transmuxes the video feed to HLS and/or MPEG-DASH.

Live Streaming Workflow

Dynamic Cloud Packaging streams a live event through the following workflow:

1. **Request:** A media player requests the playback of the live stream via HLS or MPEG-DASH.
   This request identifies:
   1. A streaming service (i.e., Dynamic Cloud Packaging).
   2. A streaming technology (i.e., HLS or MPEG-DASH).
   3. A live stream.
2. **CDN (HTTP Large):** This request is routed to the POP closest to the viewer.
   A check will be performed to see whether the requested content has been previously cached.
   - **Cached:** If the requested dynamic stream has been previously cached, then it will be streamed directly from the edge of our network to the viewer.
   - **Not Cached:** The request will be forwarded to the publishing point where it was ingested and transmuxed into HLS and/or MPEG-DASH.
3. **Dynamic Cloud Packaging:** A Dynamic Cloud Packaging publishing point will deliver the requested stream via the POP closest to the viewer.

Setup

The following tasks must be performed to stream a live event:

i. Set up a Live Authentication key.

ii. Create an instance.

iii. Set up stream broadcasting.

iv. Implement a media player.

Live Authentication Key

Important: Live Authentication is a mandatory security measure for streaming a live event via Dynamic Cloud Packaging. It is provided for your protection and it prevents unauthorized streams from being hosted on your account.

Our CDN service prevents unauthorized users from publishing a live stream via your account by requiring that an encoder authenticate to the publishing server through the use of a live authentication key. This type of key is defined on the **Live Auth** page.

There are two types of live authentication keys, which are:

Type	Description
Global Key	Authorizes publishing to any location that has not been secured by a stream key. **Tip:** This type of live authentication key makes it easy to secure a single live event that contains multiple streams. **Note:** Only a single global key may be defined at any given time.
Stream Key	Authorizes publishing for a specific stream. **Note:** Zero or more stream keys may be defined.

Requirements

Make sure that a live authentication key meets the following requirements:

- **(Global/Stream) Key Value:** 256 alphanumeric characters or less
- **Stream Path (Stream Key Only):** A stream-specific live authentication key must identify an instance and a stream name that will be paired with a key. This value is known as a stream path.
  This stream path must consist of:
  - Alphanumeric characters
  - A single forward slash that delimits the instance name from the stream name.

  Optionally, an asterisk (*), which represents one or more characters, may be appended to the stream name.

  Tip: Do not specify a pattern after an asterisk as it will be ignored.

Setup

Setting up live authentication consists of performing the following steps:

1. Define one or more live authentication key(s).
2. Configure an encoder to pass a stream name along with a live authentication key.
3. Use the following syntax when setting up FMLE's **Stream** option.
   StreamName?*LiveAuthenticationKey*

Note: It may take up to 1 hour for changes to your live authentication configuration to take effect.

Global Key Setup

Perform the following steps to define a global key:

1. Navigate to the **Live Authentication** page.
2. Set the **Global Key** option to the desired value.
3. Click **Update**.

Stream Key Setup

Stream key setup requires defining a stream path and a key. A stream path, which is defined within the **Stream Path** option, identifies a stream using the following syntax:

InstanceName/StreamName

To define a stream key

Perform the following steps to define a stream key:

1. Navigate to the **Live Authentication** page.
2. Click **Create Key**.
3. Set the **Stream Path** option to a value that identifies the stream that will be published.
4. Set the **Stream Key** option to the desired value.
5. Click **Add**.

Multiple Streams

If the encoder is publishing multiple streams of varying bit rate quality, then perform either of the following steps:

- Create a stream key that authorizes all streams that have the same base stream name.
  Syntax:
  *InstanceName/BaseStreamName**

- Create a stream key for each unique stream (e.g., mystream100, mystream250, mystream700, etc.).

To authorize multiple streams via stream keys

1. Decide upon a naming convention for each stream that will be broadcast by your encoder.
   Example: myevent250, myevent500, and myevent750.

2. Create a stream key whose stream path is set to:
   *InstanceName/BaseStreamName**
   Example:
   myinstance/myevent*

3. Specify each stream name in your encoder.

   - Make sure that this stream name does not include parameters (e.g., %i, %v, %a, etc.).

   - Use a delimiter between each stream name/stream key combination.
     Adobe Flash Media Live Encoder uses semi-colons to delimit stream names.

Authorizing Live Stream Publishing

Authorize an encoder to publish a live stream by appending a live authentication key to the stream name. Use the following syntax when defining the name of the stream being published.

Encoder	Syntax / Example
Single Stream	**Syntax:** *StreamName*?*LiveAuthenticationKey* **Example:** mystream?mykey123
Multiple Streams	**Syntax:** *StreamName*%b?*LiveAuthenticationKey* **Note:** A global or stream key may be used to authorize multiple streams. However, a single stream key may only authorize multiple streams if its stream path contains an asterisk. An alternative approach is to create a stream key for each desired stream name. **Example:** mystream%b?mykey123

Note: The purpose of a Live Authentication key is to authenticate that a stream is authorized for publishing. It should not be used for live stream playback.

Best Practices - Live Streaming Security

Live authentication keys should be treated like any other security credential or password. It is paramount to keep these keys as secure as possible. We have observed incidents in which customers lost control of their Live Authentication keys and then experienced unauthorized streaming on their account.

Although Live Authentication keys may be exposed as plain text in an encoder or streaming tool configuration, cautionary steps should be taken to prevent them from falling into the wrong hands. The following precautions are recommended:

- Ensure that Live Authentication keys are never shared outside of your organization. For example, they should not be inadvertently posted on an online support form.
- Use stream keys whenever possible.
- Periodically change your global and stream keys.
- Perform general administrative security tasks on a regular basis. These tasks include:
    - Remove old user accounts.
    - Change passwords on a regular basis.
    - Reminder users to use complex passwords.

Instances

An instance allows:

- An encoder to push streams to our network.
- Our servers to determine how a stream will be published.
- A media player to optimize the viewing experience by dynamically requesting streams of varying quality.

Key information:

- The following actions may take up to an hour before they take effect:
  - Creating an instance.
  - Deleting an instance.
- A set of unique publishing point URLs and playback URLs are assigned to an instance upon its creation.
- All CDN URLs, including publishing point and player URLs, are case-sensitive.

> **Tip:** It is recommended to copy and paste these URLs when setting up an encoder and media player(s).

- An instance may not be modified after it has been created.
- An instance may be configured to automatically archive live streams to CDN storage. This capability is known as Server-Side Archiving (SSA).

Creating an Instance

An instance may be created for each live event. Alternatively, once a live event is over, an instance may be reused to stream a different live event.

To create an instance configuration

1. Navigate to the **Dynamic Cloud Packaging - Live** page.
2. Click **New Instance**.
3. In the **Instance Name** option, type the name of the instance that will be created.
4. **HLS Only:** Use the **Encrypt HLS** option to determine whether streams generated for this instance should be encrypted.
5. Use the **DVR** option to determine whether DVR will be enabled for this instance. Upon enabling this option, define the length, in minutes, of the DVR window.
6. Click **Add** to create a new instance configuration.

7. View the desired publishing point and playback URLs.
   i. Expand the instance configuration by clicking on it.
   ii. From the **Publish URL** option, select the location closest to the encoder. The corresponding publishing point URL will be displayed directly to the right of that option.
   iii. From the **Playback URL** option, select either "HLS" or "MPEG-DASH." The corresponding playback URL will be displayed directly to the right of that option.

Reminder: It may take up to an hour before all clients can connect to your newly created instance.

Deleting an Instance

Delete an instance by performing the following steps:

1. Navigate to the **Dynamic Cloud Packaging - Live** page.
2. Expand the desired instance by clicking on it.
3. Click **Delete**.
4. When prompted, confirm the deletion of the instance.

Important: An instance configuration can be deleted even if there are clients connected to it. The deletion of an instance will cause all connections to that stream to be dropped.

Reusing an Instance

A single instance configuration may be reused to stream different live events. Before pointing an encoder to a previously used instance configuration, make sure that the following conditions have been met:

- The instance is not currently in use by another encoder.
- The instance's DVR window has elapsed.
- **Rules Engine:** If a custom cache policy has been applied to the manifest files or the file chunks generated for an instance, then either of the following actions must take place:
  - The TTL for the corresponding manifest files and file chunks must expire.
  - The corresponding manifest files and file chunks must be purged.

Note: Attempting to reuse an instance that does not meet the above conditions may result in playback issues.

Settings

Each instance configuration setting is described below.

Option	Description
Instance Name	Defines the name that will be assigned to the streaming configuration. **Note:** This name will be incorporated into both the publishing point and playback URL.
Enable SSA	Determines whether the live streams generated from an instance will be archived to CDN storage. This capability is known as Server-Side Archiving (SSA). **Important:** Both Encrypted HLS and Encrypted Key Rotation are incompatible with Server-Side Archiving. **Note:** The **Enable SSA** option is only available when the Server-Side Archiving feature has been activated on your account. If this option is not available when creating an instance, please contact your CDN account manager to activate this feature.
Encrypt HLS	Determines whether streams associated with this instance will undergo AES-128 encryption. Encrypted streams can only be decrypted by players that support encrypted HLS (e.g., iOS devices, QuickTime, and Android devices). **Important:** Both Encrypted HLS and Encrypted Key Rotation are incompatible with Server-Side Archiving. **Note:** The **Encrypt HLS** option is only available when the Encrypted HLS feature has been activated on your account. If this option is not available when creating an instance, please contact your CDN account manager to activate this feature. **Note:** This feature does not require additional CDN or player configuration. A player that supports encrypted HLS can automatically play encrypted streams. **Note:** This option only affects HLS streams. Regardless of this option, MPEG-DASH streams will be served in an unencrypted format.

Option	Description
Encrypt Key Rotation	Determines whether the encryption key generated for encrypted HLS will be rotated. If enabled, it also determines the time interval, in seconds, at which the key will be rotated. **Important:** Both Encrypted HLS and Encrypted Key Rotation are incompatible with Server-Side Archiving. **Note:** The **Encrypt Key Rotation** option is only available when the Encrypted Key Rotation feature has been activated on your account. If this option is not available when creating an instance, please contact your CDN account manager to activate this feature.
DVR	Determines whether DVR will be enabled on the current instance and the length of the DVR window (in minutes). For the purpose of this document, DVR provides a viewer with the capability to pause and rewind a live stream. The length of time from the present moment that a viewer can rewind a live stream is known as the DVR window. The length of this DVR window can be set from 5 to 180 minutes (i.e., 3 hours). Upon the completion of a live event, viewers may continue to view the live stream on delay for whichever of the following two conditions is shorter: • DVR window length • Total live stream length **Note:** A live stream must have a minimum buffer window of 60 seconds. This window exists regardless of whether DVR has been enabled on a stream. **Note:** Disconnecting a media encoder from our service does not affect an instance's DVR window. This ensures that viewers watching a live stream on delay may view it in its entirety.

Option	Description
Segment Size	**Important:** This is an advanced setting that requires careful planning. Changing the default value for this setting may cause player incompatibility and playback issues. For more information, please consult your media player's documentation. Determines the size of the segments that will be generated for the instance. Segment size, which is defined in seconds, can be set from 1 - 20 seconds. **Default value:** 10 seconds **Note:** Apple recommends segments of 10 seconds to achieve a balance between latency, startup time, and network overhead.

Broadcasting Encoded Media (Encoder)

Once an instance configuration has been created, an encoder (i.e., Adobe Flash Media Live Encoder 3.2) may publish one or more streams to it.

Basic encoder configuration consists of:

- Defining the audio/video source from which the live feed will be generated.
- Defining the audio/video output (e.g., output format, audio/video codec, bit rate levels, etc.).
- Defining the CDN ingest location and stream name(s).

Tip: Please refer to your encoder's documentation for detailed setup information.

Defining an Encoder's Output

The video output generated by an encoder defines the viewing experience. Key settings that require special attention are:

- **Video Format:** Video should be encoded using one of the following H.264 codecs:
  - Baseline Level 3.0
  - Baseline Level 3.1
  - Main Level 3.1
- **Audio Format:** HE-AAC or AAC-LC up to 48 kHz, stereo audio
- **Bit Rate Levels:** Define each bit rate level that will be generated by the encoder.

  Flash Media Live Encoder: It is strongly recommended to encode your audio/video feed as a single bit rate stream. The use of multiple bit rate streams (a.k.a. dynamic streaming or adaptive bit rate streaming) is unsupported and should only be used at your own discretion.

Tip: When defining bit rate levels, try to strike a balance between providing high quality feeds and the amount of bandwidth supported by the computer hosting your encoder. The minimum bandwidth required by an encoder can be calculated by summing up all of the bit rate levels being generated by it.

Defining the CDN Ingest Location & Stream Name(s)

The **Dynamic Cloud Packaging - Live** page provides a set of publishing point URLs for each instance configuration. Use these URLs when pushing an encoder's media output to our service.

Key information:

- In order to prevent unauthorized streams, a global or stream key is required to authorize each stream published by an encoder to our servers. This live authentication key must be appended after the stream name.
  Example: MyStream?MyLiveAuthenticationKey
  - **Stream Keys:** Unlike a global key, a unique stream key is required for each bit rate stream (e.g., MyStream750, MyStream500, and MyStream250).
- Encoding multiple bit rates requires that the name of each stream include its total bit rate value.
  The easiest way to specify a unique name for each stream is to include the %b parameter in the stream name. However, this requires the use of a global live authentication key.
  Example: MyStream%b?MyLiveAuthenticationKey
- A stream name is case-sensitive and should not contain spaces. Additionally, it should not include other special characters (e.g., !@#). Notable exceptions are listed below.
  - **?:** A question mark should delimit the stream name from a live authentication key.
  - **%:** FMLE supports the use of a percentage symbol to specify a variable in the stream name.

To define an encoder's output location

1. Navigate to the **Dynamic Cloud Packaging - Live** page.
2. Find the desired instance and then copy the publishing point URL.
3. Perform the following steps to define the CDN ingest location:
   1. Paste the publishing point URL into the **FMS URL** option.
   2. Delete "/<streamName>?<Live Authentication Key>" from the **FMS URL** option.
      The **FMS URL** option should now look like:
      rtmp://fso.dca.0001.edgecastcdn.net/200001/myinstance
4. Perform the following steps to define the streams that will be generated:
   1. From the **Stream** option, type the name by which the stream will be identified.

      Multi-bit Rate: Please append %b to the stream name when encoding multiple bit rates.

   2. Append a question mark to the stream name.
   3. Append a Live Authentication key directly after the question mark.
      The **Stream** option should now look like:
      MyStream%b?MyLiveAuthenticationKey
5. Double-check your encoder's media output settings and then start encoding the live stream.

Media Player

Setting up a HLS or MPEG-DASH-compatible media player requires pointing it to the live stream via a playback URL. The two types of playback URLs are:

- **CDN URL:** Instance-specific playback URLs are provided on the **Dynamic Cloud Packaging - Live** page.
- **Edge CNAME URL:** Use this type of URL to generate a friendlier playback URL.

Tip: Add SSL support to an edge CNAME to allow live streams to be served over HTTPS.

To construct a playback URL (CDN URL)

1. Navigate to the **Dynamic Cloud Packaging - Live** page.
2. Copy a base playback URL from the desired instance.
3. If the encoder was configured to publish to a relative path, append the same relative path to the playback URL.
4. Replace "<streamName>" with the stream name defined in the encoder's publishing session.
   Multiple Streams: Use the following syntax to playback multiple streams of varying quality (i.e., bit rates):
   BaseStreamName,*BitRate1*,*BitRate2*,*BitRateN*,.*FilenameExtension*

   Note: If a stream name contains a suffix, then it should be appended after the last comma.

5. Verify that the playback URL looks similar to one of the following URLs:
   HTTP Live Streaming (Dynamic Streaming):
   http://wpc.0001.edgecastcdn.net/240001/myinstance/mystream,750,500,250,.m3u8
   MPEG-DASH (Dynamic Streaming):
   http://wpc.0001.edgecastcdn.net/240001/myinstance/mystream,750,500,250,.mpd

To construct a playback URL (Edge CNAME URL)

1. Create an edge CNAME configuration.

    i. Make sure that the **Origin Directory** option is set to:
    /24*xxxx* (Dynamic Cloud Packaging Live Playback Instance).

    ii. Wait an hour for this edge CNAME configuration to take effect.

    iii. Add the corresponding CNAME record via a DNS service provider.

2. Construct a playback URL using the following syntax:
http://*EdgeCNAME*/*InstanceName*/*StreamName*.*ext*

    i. Replace *EdgeCNAME* with the name (e.g., live.mydomain.com) of the edge CNAME configuration created in the previous step.

    ii. Replace *InstanceName* with the name of the desired instance.

    Tip: View a list of instance names from the **Dynamic Cloud Packaging - Live** page.

    iii. If the encoder was configured to publish to a relative path, then insert it into the playback URL as indicated below.
    http://*EdgeCNAME*/*InstanceName*/*RelativePath*/*StreamName*.*ext*

    iv. Replace *StreamName* with the stream name defined in the encoder's publishing session.

    Note: If a stream name contains a suffix, then it should be appended after the last comma.

    v. Replace *ext* with the filename extension corresponding to the desired streaming solution.

        - **HLS:** .m3u8
        - **MPEG-DASH:** .mpd

3. Verify that the playback URL looks similar to one of the following URLs:
HTTP Live Streaming (Dynamic Streaming):
http://live.mydomain.com/myinstance/mystream,750,500,250,.m3u8
MPEG-DASH (Dynamic Streaming):
http://live.mydomain.com/myinstance/mystream,750,500,250,.mpd

HTTPS Streaming

Live streams may be served over HTTPS. The two main benefits of streaming content over HTTPS are:

- It allows end-to-end encryption. This ensures secure communication between the viewer and the CDN.
- It improves the user experience by eliminating the web browser-generated security warning that pops up when a HTTPS web page contains content delivered over HTTP (e.g., audio/video).

An important caveat to HTTPS streaming is that only certain media players (e.g., iOS devices and QuickTime) support this capability. Refer to the media player's documentation to find out whether it supports streaming over HTTPS.

Tip: Use encrypted HLS to apply additional security to your live streams.

Live Streaming over HTTPS Setup

Setting up HTTPS streaming involves the following steps:

1. Contact your CDN account manager to request an SSL certificate.
2. Once your CDN account manager has informed you that the SSL certificate has been installed on our network, create or update an edge CNAME for the HTTP Large platform.
3. Set the edge CNAME's **Origin Directory** option to:
   /24 (Dynamic Cloud Packaging Live Playback Instance)
4. From your DNS service provider, update a CNAME record to point the edge CNAME to the hostname provided by your CDN account manager.
5. Point the desired media player to a playback URL that leverages the above edge CNAME.

Basic Playback URL Information

Basic information on playback URLs is described below.

- A unique playback URL is assigned to each instance.
- View an instance's playback URL from the **Dynamic Cloud Packaging - Live** page.
- An encoder may publish to either of the following locations:

Publishing Location	Description
Root Folder	An encoder may publish to the default publishing point location as defined by an instance's publishing point URL. **Sample publishing point URL:** rtmp://fso.oxr.0001.edgecastcdn.net/240001/myinstance
Relative Path	An encoder may publish to a subdirectory of the location defined by the instance's publishing point URL. This type of setup requires that an identical relative path be appended to the playback URL. **Sample publishing point URL:** rtmp://fso.oxr.0001.edgecastcdn.net/240001/myinstance/sales/videos **Sample playback URL:** http://wpc.0001.edgecastcdn.net/240001/myinstance/sales/videos

- The stream name defined in the playback URL varies according to whether the media player will play back a single or multiple streams.
- The player's URL filename extension determines the streaming solution that will be leveraged by the media player.
  HTTP Live Streaming:
  http://wpc.*AN*.edgecastcdn.net/24*AN*/*Instance*/*Path*/*StreamName*.m3u8
  MPEG-DASH:
  http://wpc.*AN*.edgecastcdn.net/24*AN*/*Instance*/*Path*/*StreamName*.mpd

Single Stream

Configure a media player to playback a specific stream by modifying the desired instance's playback URL to point to the stream generated by the encoder. This stream name should exclude the specified live authentication key.

Encoder's Stream Name:

StreamName?*LiveAuthenticationKey*

Playback URL's Stream Name:

StreamName.*FilenameExtension*

Multiple Streams (Dynamic Streaming)

Encoding multiple streams of varying quality and then referencing them within a playback URL allows a media player to vary the stream's bit rate quality to provide an optimal viewing experience. Specifically, the media player will analyze a user's environment at frequent intervals and dynamically choose the stream that provides the highest bit rate quality that the client can support without causing buffering or stuttering.

This capability requires the following:

- An encoder must publish several streams of varying quality.
- The base stream name defined in the encoder must be the same for all streams. Use the %b parameter to identify each stream by its total bit rate level.
- Each bit rate level must be indicated in the player URL. A media player's initial request will be for the first bit rate specified in the player URL.

Syntax

The player URL for multiple streams must identify each stream generated by the encoder. Multiple streams can be specified within a single URL by following the file naming conventions described below.

Note: The terms prefix and suffix refer to the portions of the stream name that appear before and after, respectively, the %b parameter.

Filename Conventions:

- All streams must have a common prefix.
  Example: video100, video200, and video300
- The end of the prefix should be indicated with a comma. This should be followed by a comma-delimited list of the bit rate values used to identify each unique stream. Append a comma after the final bit rate level.
  Example: video,100,200,300,
- Each specified bit rate stream should only include the total bit rate quality in Kbps. Including any other data or using different units will generate a malformed playlist.
- A suffix identifies the portion of the stream name that extends beyond the comma-delimited list of bit rate values used to identify each unique stream. If a suffix has been specified, then it may be appended after the last comma.
  - **Example:** The suffix for the following sample streams is "kbps."
    video100kbps, video200kbps, and video300kbps

    Append the suffix after the last comma.
    video,100,200,300,kbps

Archiving a Live Event (Server-Side Archiving Preview)

Important: The Server-Side Archiving capability, which must be activated via your CDN account manager, is available as a preview. Our support policy for preview capabilities is to offer our best effort to resolve issues. Therefore, it is not recommended that critical business workflows rely on the Server-Side Archiving capability.

Important: Both Encrypted HLS and Encrypted Key Rotation are incompatible with Server-Side Archiving.

A live event can be recorded as it is being streamed and then archived to CDN storage. This process is known as Server-Side Archiving (SSA). This allows a live event to be streamed as on-demand content.

Server-Side Archiving Activation

The Server-Side Archiving capability must be activated on the desired instance before the live streams generated from it will be archived. This feature can be activated by marking the **Enable SSA** option when creating an instance.

Note: The **Enable SSA** option is only available when the Server-Side Archiving feature has been activated on your account. If this option is not available when creating an instance, please contact your CDN account manager to activate this feature.

Note: If SSA has been enabled on an instance, then all of its live streams will be automatically archived.

Archived Content

If a live stream is being published to an instance on which SSA has been enabled, then a new file will be generated whenever any of the following conditions are true:

- The live stream is interrupted.
- The size of the archived file reaches 500 MB.
- An encoder starts encoding a new live event.

Note: A live event is always archived as an MP4 file.

Note: It typically takes approximately 15 minutes to generate archived content. However, longer live events may experience a delay.

Storage Location

This file will be stored in CDN storage within a folder named after the live event's instance name. This folder is located in the root folder of your CDN storage account.

For example, if you set your encoder to output encoded media to:

```
rtmp://fso.dca.0001.edgecastcdn.net/200001/myinstance
```

Then the live stream would be archived to the following relative path on CDN storage:

```
/myinstance
```

Reminder: Manage content stored on CDN storage via a third-party FTP client.

Naming Convention

The naming convention for archived content is described below.

Syntax:

StreamName-UnixTime.mp4

Example:

```
mystream-1493145518.mp4
```

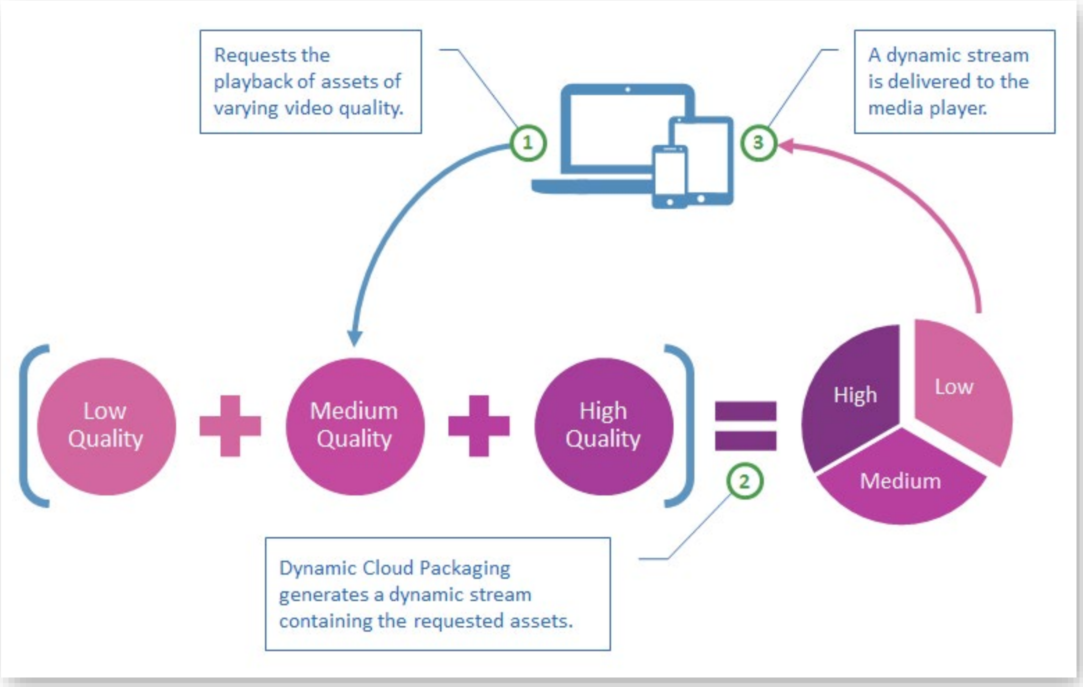
On-Demand Streaming

Introduction

Dynamic Cloud Packaging can stream on-demand content to any standard HLS or MPEG-DASH player.

How Does It Work?

Dynamic Cloud Packaging can dynamically repackage one or more assets into a HLS or MPEG-DASH-compatible stream. This process, which is known as transmuxing, allows a media player to leverage dynamic streaming on a set of assets as if they had been packaged together.


Requests the playback of assets of varying video quality.
1
3
A dynamic stream is delivered to the media player.
Low Quality
Medium Quality
High Quality
High
Low
Medium
2
Dynamic Cloud Packaging generates a dynamic stream containing the requested assets.

How Does On-Demand Streaming Work?

Delivery

Dynamic Cloud Packaging leverages the HTTP Large platform to stream on-demand content. This allows it to take advantage of our caching technology and our network to efficiently deliver on-demand content from a customer origin or CDN storage.

Workflow

The workflow through which Dynamic Cloud Packaging streams on-demand content is described below.

1. **Request:** A media player submits a request to stream one or more assets.
   This request identifies:
   - A streaming service (i.e., Dynamic Cloud Packaging).
   - A streaming technology (i.e., HLS or MPEG-DASH).
   - The set of assets that will be streamed.
   - The location where the assets can be found.
2. **CDN (HTTP Large):** This request is routed to the POP closest to the viewer.
   A check will be performed to see whether the requested content has been previously cached.
   - **Cached:** If the requested dynamic stream has been previously cached, then it will be streamed directly from the edge of our network to the viewer.
   - **Not Cached:** The request will be forwarded to Dynamic Cloud Packaging.
3. **Dynamic Cloud Packaging:** Dynamic Cloud Packaging will transmux the set of assets requested by the media player. This process generates a HLS or MPEG-DASH manifest file that allows the media player to dynamically switch streams based on the viewer's bandwidth and CPU usage. Once this manifest file has been generated, the requested content will be streamed via the HTTP Large platform to the viewer.

Setup

The following tasks must be performed to stream on-demand content:

Task	Component	Description
Data Upload	Origin Server	The desired H.264 media files must be uploaded to either: • CDN storage • An external server (i.e., customer origin server)
Media Player Implementation	Media Player	An HLS and/or MPEG-DASH media player that points to those media files must be implemented and made available to the desired users.

Note: Our CDN service will take care of packaging and segmenting the video into a format that can be understood by the HLS or MPEG-DASH-compatible media player.

Origin Setup

On-demand content (i.e., H.264 media files) may be streamed from either a customer origin server or CDN storage.

CDN Storage

Streaming on-demand content from CDN storage involves the following steps:

1. Upload the desired H.264 media files to CDN storage via SFTP, rsync, or FTP.
2. Optional. Set up an edge CNAME whose **Origin Directory** option is set to:
   /04AN (Dynamic Cloud Packaging CDN Origin)

   Note: Edge CNAME setup information varies according to whether it will serve HTTP or HTTPS traffic.

3. Leverage a CDN or edge CNAME URL to point a media player to the desired H.264 media file.

Customer Origin

Streaming on-demand content from a customer origin involves the following steps:

1. Create a customer origin configuration that points to the desired web servers.
2. Upload the desired H.264 media files to the web servers referenced by the above customer origin configuration.
3. Optional. Set up an edge CNAME whose **Origin Directory** option is set to:
   /84AN/*CustomerOrigin*

   Note: Edge CNAME setup information varies according to whether it will serve HTTP or HTTPS traffic.

4. Leverage a CDN or edge CNAME URL to point a media player to the desired H.264 media file.

HTTPS Streaming

On-demand content may be streamed over HTTPS. The two main benefits of streaming content over HTTPS are:

1. It allows end-to-end encryption of on-demand content. This ensures secure communication between the viewer and the origin server.
2. It improves the user experience by eliminating the web browser-generated security warning that pops up when a HTTPS web page contains content delivered over HTTP (e.g., audio/video).

An important caveat to HTTPS streaming is that only certain media players (e.g., iOS devices and QuickTime) support this capability. Refer to the media player's documentation to find out whether it supports streaming over HTTPS.

Tip: Use encrypted HLS to apply additional security to your on-demand streams.

Setup

Setting up HTTPS streaming involves the following steps:

1. Perform the above steps to set up CDN storage or a customer origin.
2. Contact your CDN account manager to request an SSL certificate.
3. Once your CDN account manager has informed you that the SSL certificate has been installed on our network, create or update an edge CNAME for the HTTP Large platform. This edge CNAME must be configured as indicated below.
    - **CDN Storage:** Set the **Points To** option to "CDN Origin." After which, set the **Origin Directory** option to:
      /04AN (Dynamic Cloud Packaging CDN Origin)
    - **Customer Origin:** Set the **Points To** option to "Customer Origin." After which, set the **Origin Directory** option to:
      /84AN/*CustomerOrigin*
4. From your DNS service provider, update a CNAME record to point the edge CNAME to the hostname provided by your CDN account manager.
5. Point the desired media player to a player URL that leverages the above edge CNAME.
   Player URL syntax for HLS:
   https://*EdgeCNAME*/*RelativePath*/*Filename*.m3u8
   Player URL syntax for MPEG-DASH:
   https://*EdgeCNAME*/*RelativePath*/*Filename*.mpd

Media Player

Setting up a HLS or MPEG-DASH-compatible media player requires pointing it to a valid playback URL. A playback URL may be constructed from the base CDN URLs provided on the **Dynamic Cloud Packaging - VOD** page.

Constructing a playback URL involves the following steps:

1. Copy a base CDN URL that corresponds to the desired origin type from the **Dynamic Cloud Packaging - VOD** page.
2. **Customer Origin Only:** Append a forward slash and the name of the desired customer origin configuration to the base playback URL.
3. Append the relative path to the directory where the desired on-demand content can be found.
4. Append the filename of the desired asset(s).
   A media player can point to one or more assets. The syntax for specifying a filename varies according to whether a single or multiple assets are defined in the URL.
   Single asset syntax: *Filename*.mp4
   Multiple assets syntax: *BaseFilename*,*BitRate1*,*BitRate2*,*BitRateN*,.mp4
5. Append either a HLS or MPEG-DASH filename extension.
   - **HLS:** .m3u8
   - **MPEG-DASH:** .mpd
6. Verify that the playback URL looks similar to the following URL:
   Sample playback URL:
   http://wpc.0001.edgecastcdn.net/040001/videos/fly,110,400,650,.mp4.m3u8

Tip: Alternatively, a friendlier player URL may be generated by creating an edge CNAME configuration.

Single Asset

A single H.264 asset may be streamed from CDN storage or a customer origin server.

Note: The bit rate at which this asset was encoded determines playback quality.

Syntax:

Filename.mp4.*FilenameExtension*

Multiple Assets

Specifying multiple H.264 assets within a playback URL allows the media player to vary the stream's bit rate quality to provide an optimal viewing experience. Specifically, the media player will analyze a user's environment at frequent intervals and dynamically choose the asset that provides the highest bit rate quality that the client can support without causing buffering or stuttering.

This capability requires the following:

- An H.264 asset for each desired bit rate stream. All assets referenced within a single playback URL must be stored in the same directory.
- The filename for each H.264 asset should indicate its corresponding bit rate quality in Kbps (e.g., 100, 200, or 400). The bit rate quality is the only difference allowed in the naming convention.
- Each bit rate stream should be indicated in the player URL. A media player's initial request will be for the first bit rate specified in the player URL.

Syntax

Follow these conventions when specifying multiple assets within a single playback URL:

Note: The terms "prefix" and "suffix" refer to the portions of the filename that appear before and after, respectively, the bit rate level.

- All filenames must have a common prefix.
  Example: video100.mp4, video200.mp4, and video300.mp4
- The end of the prefix should be indicated with a comma. This should be followed by the bit rate level for each desired stream as a comma-delimited list in the filename. Append a comma after the final bit rate level.
  Example: video,100,200,300,.mp4
- Each specified bit rate stream should only include the bit rate quality in Kbps. Including any other data or using different units will generate a malformed playlist.
- A suffix identifies the portion of the filename that extends beyond the bit rate level. The suffix for the following sample filenames is "kbps."
  video100kbps.mp4, video200kbps.mp4, and video300kbps.mp4
- Append the suffix after the last comma.
  Example: video,100,200,300,kbps.mp4

Preventing Unauthorized Viewing

Overview

Secure media against unauthorized viewing through the following methods:

- **Token-Based Authentication:** Use time, country, URL, IP address, and referrer information to determine whether users can view your stream.
- **Stream Encryption:** Apply AES-128 encryption to your stream.

Token-Based Authentication

The Token-Based Authentication feature may secure live and on-demand streams by requiring that a viewer satisfy a set of security requirements before being granted access to your content. This type of configuration is supported under the following circumstances:

- Content should not be streamed from a location defined in the **Directories to Authenticate** section of the **Token Auth** page.
- Token-Based Authentication should only be enabled on manifest files (i.e., m3u8 and mpd). This may be achieved via the following Rules Engine configuration:
  - **Match:** Match the m3u8 and/or mpd filename extensions via the URL Path Extension Literal match condition.
  - **Feature:** Enable the Token Auth feature.

To create a rule that secures manifest files with Token-Based Authentication (Rules Engine v4)

1. If a policy has been deployed to the Production environment, then create a copy of it. Otherwise, create a draft.
2. Modify the draft to include the URL Path Extension Literal match condition.
3. Configure this match condition as indicated below.
4. Set the **Result** option to "match."
5. Set the **Value** option to "m3u8 mpd" to secure HLS and MPEG-DASH manifest files.
6. Set the **Ignore Case** option to "yes."
7. Directly below this match condition, add the Token Auth feature.
8. Click + .
9. Select "Feature." set the category to "Access," and then select "Token Auth."
10. Enable this feature by clicking **no** under the **Enabled** option. This option will now be set to "yes."
11. Click **Save**.
12. Convert the draft into a policy.
13. Deploy the policy to the Production environment.

Stream Encryption

Important: Both Encrypted HLS and Encrypted Key Rotation are incompatible with Server-Side Archiving.

Note: Stream encryption requires the activation of the Encrypted HLS feature. Please contact your CDN account manager to activate this feature.

AES-128 encryption can be applied to HLS streams generated for your live events and on-demand content. Encrypted streams can only be decrypted by players that support encrypted HLS (e.g., iOS devices, QuickTime, and Android devices). Players that do not support encrypted HLS will be unable to play back encrypted streams.

Key information:

- Enabling encrypted HLS will generate encrypted streams. This will result in encrypted content being cached on our network.
- Disabling encrypted HLS requires that all cached HLS content be purged to prevent encrypted cached content from impacting playback.
- The playback of an encrypted stream can be performed by any media player that supports encrypted HLS playback. It does not require media player customization or additional configuration.
- Only HLS streams can be encrypted at this time. This means that it may be possible to download or stream your content using a different streaming technology (e.g., MPEG-DASH).

Tip: Additional protection may be applied to sensitive streams by denying all non-HLS requests for H.264 assets. This type of setup may be achieved via Rules Engine. For more information, please contact your CDN account manager.

Live Streaming Configuration

An event's configuration determines whether its streams will be encrypted. Specifically, the **Encrypt HLS** option toggles whether AES-128 encryption will be applied to all streams associated with the instance.

Key Rotation

Note: Key rotation requires the activation of the Encrypted Key Rotation feature. Please contact your CDN account manager to activate this feature.

The encryption key generated for a live stream may be rotated at regular intervals to prevent unauthorized playback via a shared link. Upon enabling this capability, a media player will be required to fetch the latest version of the encryption key at the specified interval.

Key information:

- This interval may be defined on a per instance basis by setting the **Encrypt Key Rotation (seconds)** option to the desired interval in seconds.
- This interval must be specified in multiples of 5 (e.g., 10, 15, 20, etc.) seconds.
- The specified interval must be greater than or equal to 10 seconds.
- The specified interval must be less than or equal to 1440 seconds (i.e., 24 minutes).

On-Demand Streaming Configuration

The **Protected Directories for Encrypted HLS** section on the **Dynamic Cloud Packaging - VOD** page defines the set of locations that will generate encrypted streams from on-demand content.

Secure a directory by:

- Specifying the relative path to the desired directory.
- Indicating whether the above directory applies to all customer origins or CDN storage.

Key information:

- An encrypted directory only applies to the selected origin type (i.e., CDN storage or customer origin).

  Tip: The same relative path may be secured for both CDN storage and customer origins by creating an encrypted directory configuration for each origin type.

- The path to a protected folder always starts with a forward slash (/).
- The path to a protected folder is case-insensitive.
- Encrypted HLS cannot be applied to the root folder.
- It may take up to an hour before a new location is fully protected.
- Wildcard characters (e.g., *) are not supported when setting up protected directories.
- Only the on-demand content stored in the specified directory will be secured by encrypted HLS. Content that resides in a subfolder of that location must be secured separately.

- The starting point for an encrypted directory's relative path is indicated below.

URL Type	Relative Path (Starting Point)
CDN URL	Specify a relative path that starts directly after the content access point (e.g., /040001 and /840001). Use the following information to interpret the following sample URLs: • **Gray Font:** Indicates what should be excluded when securing a location. • **Blue Font:** Indicates the relative path that should be defined to secure the requested on-demand content. **Sample URL (CDN Storage):** http://wpc.0001.edgecastcdn.net/040001/mybusiness/videos/fly.mp4 **Sample URL (Customer Origin):** http://wpc.0001.edgecastcdn.net/840001/mycustomerorigin/mybusiness/videos/fly.mp4
Edge CNAME URL (CDN Origin)	Specify a relative path that starts directly after the hostname. **Sample URL:** http://www.domain.com/mybusiness/videos/fly.mp4 In the above sample URL, the gray text indicates what should be excluded when securing a location. This sample request can be secured by any of the following configurations: • /mybusiness • /mybusiness/videos

Purging Live and On-Demand Content

Content streamed over Dynamic Cloud Packaging may be purged from our network. Purging this type of content:

- Provides a quick and easy way to prevent users from viewing cached portions of a live event after it has completed.
- Allows an encoder to publish a previously used instance.

Purging an Instance

The following types of files are generated by Dynamic Cloud Packaging:

- General and bit rate-specific manifest files
- Bit-rate specific fragments

A player URL typically points to a master manifest file. This means that purging the player URL will not purge the majority of the content that was published by an encoder. In order to properly purge this content, the entire publishing location should be purged. This type of purge request is achieved through the use of a recursive purge (i.e., /*).

Purge syntax is provided below.

- **Live Streaming:**
  http://wpc.*AN*.edgecastcdn.net/24*AN*/*Instance*/*
- **On-Demand Streaming (CDN Storage):**
  http://wpc.*AN*.edgecastcdn.net/04*AN*/*Path*/*
- **On-Demand Streaming (Customer Origin):**
  http://wpc.*AN*.edgecastcdn.net/84*AN*/*CustomerOrigin*/*Path*/*

Key information:

- Submit a purge request using a CDN URL.
- It is important to specify a CDN URL that points to the correct origin identifier (i.e., 24, 04, or 84). Otherwise, content played back from Dynamic Cloud Packaging will not be purged.
- Purges are protocol-independent (i.e., HTTP or HTTPS).

Example

The following example demonstrates how to purge a live event.

Player URL:

http://wpc.0001.edgecastcdn.net/240001/myinstance/mystream.m3u8

Purge the instance:

http://wpc.0001.edgecastcdn.net/240001/myinstance/*