Verizon Media Platform

Smooth Streaming Administration Guide

verizon media

Disclaimer

Care was taken in the creation of this guide. However, Verizon 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

VERIZON is a registered trademark of Verizon.

FLASH is a registered trademark of Adobe Systems Incorporated.

SMOOTH STREAMING and WINDOWS are registered trademarks of Microsoft Corporation.

About This Guide

Smooth Streaming Administration Guide
Version 1.34
10/22/2019

© Verizon Media. All rights reserved.

Table of Contents

Smooth Streaming

Overview

Internet Information Services (IIS) 7.0 offers an HTTP-based adaptive streaming extension (Smooth Streaming), which allows the dynamic switching of bit rate streams based on your client's bandwidth and CPU usage. In other words, stream quality is improved to match a client's bandwidth, as long as that client's computer is sufficiently powerful to handle the feed. Likewise, if a client's bandwidth or CPU usage is not sufficient for the current stream, then the video stream will be seamlessly switched to a lower quality.

Smooth Streaming provides many benefits for the distribution of your content. Some of these benefits are:

- Smooth Streaming is based on the HTTP protocol. This makes it easier to implement, it is more widely supported through firewalls, and it has a more resilient infrastructure.
- By dynamically adapting to each user's environment, it avoids wasting bandwidth while delivering an optimal viewing experience.
- Dynamic bit rate switching takes the guesswork out of choosing the appropriate stream quality for each client.

Smooth Streaming also improves the viewing experience for your users in the following ways:

- Fast start-up and seek times.
- If your clients support the requirements for your minimum bit rate stream, then they will not suffer from the following issues: delays due to buffering, disconnects, or playback stutter.

Types of Smooth Streaming

As mentioned earlier, Smooth Streaming refers to the dynamic streaming extension provided by IIS 7.0. Although it is provided by IIS 7.0, you do not necessarily need your own copy of IIS 7.0 to take advantage of Smooth Streaming. Our CDN allows you to use our servers to publish your live and on-demand streams.

This guide will explain how your live and on-demand streams can take advantage of our network to ensure quicker and more reliable delivery of your streams to your clients. Our CDN allows you to generate a publishing point using either IIS 7.0 or a server in our network. In other words, you can make your streams available to your clients using any of the following Smooth Streaming configurations:

- Ingest and publish your live stream through our CDN.
- Publish your on-demand stream through our CDN.
- Ingest and publish your live stream through your own IIS server.
- Publish your on-demand stream through your own IIS server.

Note: The purpose of this guide is to assist you in the integration of Smooth Streaming with our CDN. It is not meant to replace the documentation provided with your encoder. Additionally, if you are planning on publishing streams using your own IIS server, then you should refer to Microsoft's documentation for information on how to implement a Smooth Stream solution.

Note: Although you can use your own IIS server to publish your streams, they can still be routed and cached throughout our network. This type of configuration can be achieved by pointing a Smooth Stream compatible player to the stream using a CDN or an edge CNAME URL.

How Does Smooth Streaming Work?

An outline for the process through which an event can be streamed to a Smooth Stream compatible player is provided below.

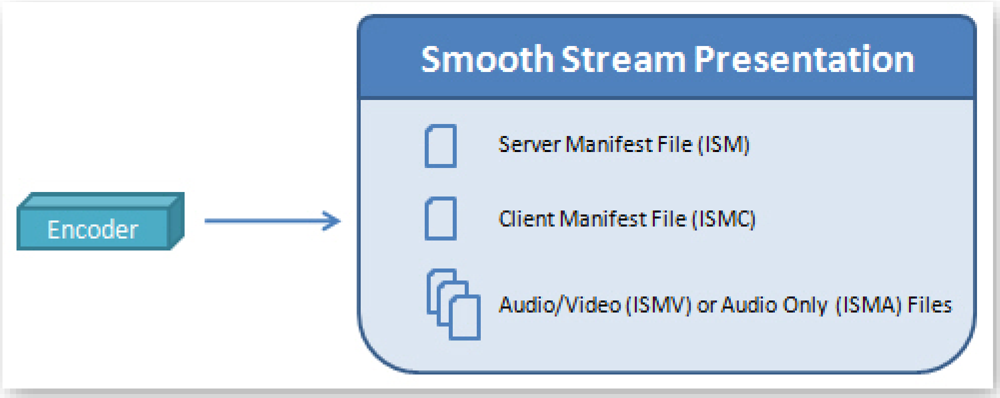
1. An event is encoded.
2. A client requests a live or on-demand smooth stream via a Smooth Stream compatible player.
3. A client manifest file is served to the client's player.
4. The player requests a file chunk and it is served to the client's player.
5. The player then requests the next file chunk based on a reassessment of the client's bandwidth and CPU usage.

Each step involved in serving a smooth stream to your clients is described below.

An Event is Encoded

Before an event can be streamed to your clients, the video associated with it will need to be encoded. The timing for this process is determined by whether it is a live or on-demand event. Live Smooth Streaming requires that an event be encoded as it occurs, while On-Demand Smooth Streaming requires that the entire event be encoded before it is made available to your clients.

Detailed Information


Encoder
Smooth Stream Presentation
Server Manifest File (ISM)
Client Manifest File (ISMC)
Audio/Video (ISMV) or Audio Only (ISMA) Files

The encoding process will generate a single ISMV or ISMA file for each desired bit rate stream. Both of these file types are encoded to match the MP4 codec specifications. Internally, each of these files is organized into virtual file segments or file chunks. The encoding process will also generate a server manifest file (ISM file) and a client manifest file (ISMC file). These files provide stream information to IIS and the Smooth Stream compatible player, respectively.

Note: Although these files are created for both live and on-demand streams, they may not be permanently stored for live streams. Please refer to the documentation provided with your encoder from more information.

Client Requests a Live or On-Demand Smooth Stream

When a client requests a stream, our edge servers will serve a client manifest file to it. This client manifest file allows the video player to request new video segments that match the client's available bandwidth and CPU.

Detailed Information

- If no other clients have requested the stream from the same POP, then an edge server on that POP will serve the client manifest file from an ingest point or an origin server. The server from which the file will be served depends on whether the client is requesting live or on-demand content, and whether you are using our Smooth Streaming services or your own instance of IIS 7.0 to publish your streams. Once the client manifest file is served to the client, it will then be cached on the edge server that served it.

- If another client has requested the stream from the same POP, then the client manifest file may be served from cache. This depends on whether the client manifest file is still fresh. If this file is stale, then the edge server will query the ingest point or an origin server to check whether it should still serve the cached version of the file.

File Chunk is Requested and Served to the Client

Once the client receives the client manifest file, it will request an audio/video segment from the ingest server. This request will identify the desired bit rate and a time segment in the stream. Once a client makes a request, one of the following will occur:

- If the desired bit rate and time segment combination has not been previously requested, then the following steps will occur:

  i. The edge server will forward the request to the ingest server.

  ii. The ingest server will use the requested bit rate to identify an audio/video (ISMV) or an audio-only (ISMA) file.

  iii. The ingest server will then use the requested time span to identify and extract an audio/video segment in that file.

  iv. It will then package the extracted file chunk as a file and send it back to the appropriate edge server.

  v. The edge server will then deliver this file to the client and cache it.

- If another client has previously requested the same bit rate and time segment and the file chunk is still fresh, then the edge server will serve the requested file chunk directly from cache.

- If another client has previously requested the same bit rate and time segment and the file chunk is stale (i.e., a file chunk that has exceeded its TTL or max-age), then the edge server will forward the request to the ingest server.

  - **Live Smooth Streaming:** A request for a stale file chunk will typically be denied. By default, a file chunk's max-age is tied to the DVR duration assigned to that publishing point. File chunks that are older than the specified time will be discarded from the ingest server. Once a file chunk has been discarded, the ingest server will deny all requests for it. In turn, the default behavior of an edge server is to deny these types of client requests.

  - **On-Demand Smooth Streaming:** A request for a stale file chunk will cause the ingest server to inform an edge server that the file has not been modified. The edge server will then proceed to update the header information for the cached version of that file and then serve the cached file directly to the client.

Reassessment of Client's Bandwidth and CPU Usage

If a Smooth Stream compatible player deems that the quality of the stream should be adjusted due to bandwidth or CPU usage, then it will request a virtual file chunk from a lower quality bit rate stream. This alternate virtual file chunk picks up at the exact point where the previous one ended. This allows the client to seamlessly switch between different quality streams. The new request will undergo the process described in the previous step. This process repeats until the stream has stopped or it has been completely downloaded.

Requirements

Before you can stream live and on-demand streams, your Smooth Streaming configuration must meet a minimum set of requirements. Please make sure that your configuration meets all applicable requirements.

General Requirements

The following requirements apply to both live and on-demand Smooth Streaming configurations:

- **Player:** Microsoft Silverlight 2 or above, or Open Video Player (OVP) Silverlight
- **CDN:** The Smooth Stream presentation (i.e., manifest and audio/video files) must be accessible through our CDN. This means that you must use a CDN URL or an edge CNAME URL when configuring the location of the stream in your video player.

Note: If you are using an IIS ingest point, then you will need to ensure that our CDN has been configured to recognize that customer origin.

Note: On-Demand Smooth Streaming can only publish Smooth Stream presentations that are located on a CDN origin server. If you would like to use a customer origin server, then you will need to make sure that it is stored and published from your own IIS origin server.

Live Smooth Streaming Requirements

The following requirements apply to all live Smooth Streaming configurations:

- **Media Encoder:** Microsoft Expression Encoder 2 Service Pack 1 (SP1) or higher

Live Smooth Streaming (IIS Ingest Point) Requirements

The following requirements apply to all live Smooth Streaming configurations that will be published using a web server outside of our network:

- **Web Server:** Internet Information Services (IIS) 7.0
    - Smooth Streaming IIS extension; Available with IIS Media Services 3.0 or higher

Interaction with the HTTP Large Platform

Smooth Streaming is served by the network of servers used by the HTTP Large platform. This means that Smooth Streaming can take advantage of the services and features that are available on the HTTP Large platform. The majority of these services and features do not require additional configuration to work properly with Smooth Streaming. The exception is Token-Based Authentication. For additional information on how Token-Based Authentication can impact your streams, please refer to the **Appendix A: Interaction with Token-Based Authentication**.

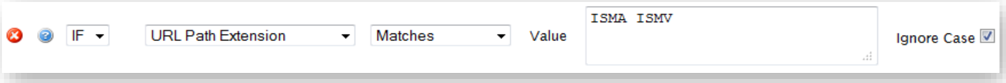
If you would like to use Token-Based Authentication on the HTTP Large platform, then you will need to perform one of the following:

- Make sure that streams are only published to locations that have not been secured by Token-Based Authentication.
- Us e HTTP Rules Engine to create a custom rule that prevents Token-Based Authentication security from being applied to ISMV/ISMA assets. A stream would still be protected by requiring authentication on the manifest file. However, it is conceivable under this type of configuration that a malicious entity with sufficient technical and logistical information could bypass this security precaution.

Note: For additional information on HTTP Rules Engine, which must be purchased separately, please contact your CDN account manager.

To prevent Token-Based Authentication security from being applied to ISMV/ISMA assets

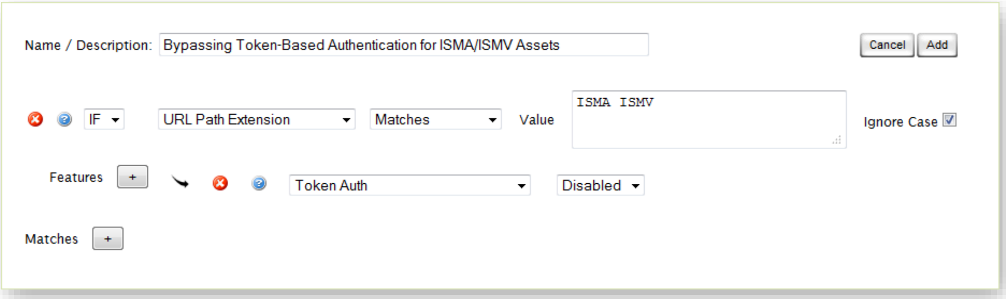
1. From the main menu, find the **HTTP Large** menu and then select **Rules Engine**.
2. Specify the name for the new rule in the **Name / Description** option.
3. For the match option, select "URL Path Extension."
4. In the **Value** option, type "ISMA ISMV."
5. Mark the **Ignore Case** option.



IF | URL Path Extension | Matches | Value: ISMA ISMV | Ignore Case ☑

6. Add a feature by clicking the plus symbol that appears next to it.

7. For the feature option, select "Token Auth" and set its status to "Disabled." Your new rule should look like the following illustration:



Name / Description: Bypassing Token-Based Authentication for ISMA/ISMV Assets

IF URL Path Extension Matches Value ISMA ISMV Ignore Case ☑

Features Token Auth Disabled

8. Click **Add**.

Note: A rule will not immediately take effect. It must first undergo an approval process. Once it has been approved, it will take an hour to take effect.

Reports and Analytics

Smooth Streaming is a service that is provided for the HTTP Large platform. As such, data for Smooth Streaming is logged under the HTTP Large platform. If you would like to distinguish between the traffic/bandwidth usage for regular HTTP Large content and Smooth Streaming, then you will need to create an edge CNAME dedicated to Smooth Streaming and enable its **Custom Reports** option. This allows our edge servers to keep traffic of the number of hits and the amount of data transferred for that edge CNAME.

Generate a custom report on Smooth Streaming traffic by:

1. From the main menu, navigate to the **Reports** menu and then select **Custom Reports**.
2. Select **HTTP Large** from the side navigation bar.
3. Statistics on the "dedicated" Smooth Streaming edge CNAME can be viewed in the table displayed below the bar graph.

Live Smooth Streaming

Overview

Live Smooth Streaming allows adaptive streaming of a live event. The encoded video for your live event can be pushed to either your own IIS server or to an ingest point on our network. From there, it will be ingested and your video feed will be published as a stream. Your clients will then request this stream through our CDN. Combining Live Smooth Streaming with the power of our CDN ensures that your customers will enjoy uninterrupted live feeds that automatically adjust to changing network conditions.

This chapter will explain how to configure Live Smooth Streaming from start to finish. It covers all of the following topics:

- How to Prepare an Ingest Point for Smooth Stream Publishing
- Encoding Audio/Video
- Setting up a CDN Delivery Configuration
- Client Implementation

Publishing Your Stream

In order to publish a stream, you will need to perform the following tasks:

- Verify that Live Smooth Streaming has been enabled for your account.
- Create a publishing point.
- Encode your video feed and broadcast it to the ingest server.

This section explains how to perform these tasks.

Enabling Live Smooth Streaming

Before you can publish your live video feed using Smooth Streaming, you will need to enable it. Once you have enabled this feature, it may take up to an hour for it to take effect. During this time period, you will be allowed to create publishing points. However, it is recommended that you do not serve the feed to your clients until the activation of this feature on your account has been fully propagated throughout our network.

To activate Live Smooth Streaming

1. Load the **Live Smooth Streaming** page.
2. If the **Enable Live Smooth Streaming playback** option is cleared, then you should mark it and then click **Update**.

Reminder: It may take up to an hour before the Live Smooth Streaming feature is fully activated.

Publishing Point

The ingest point for Smooth Streaming can be located either on our CDN or on your own IIS server. Regardless of where your ingest point is located, you will need to create a publishing point. A publishing point defines the properties of the stream that will be published. The creation of a publishing point will also provide you with publishing point URLs and a player URL. A publishing point URL is used to point an encoder to the location where the encoded video will be ingested. The player URL is used by a Silverlight or OVP player to request a published stream.

Tip: It is recommended that you create a new publishing point for each new live stream. This will ensure that old content is never served to your clients.

Note: This section only explains how to set up a publishing point when the ingest point is located on our CDN. If you are publishing your stream using your own IIS server, then you should refer to Microsoft's documentation for more information on how to configure a publishing point.

Note: It may take up to 10 minutes before the creation or modification of a publishing point takes effect.

Note: Both the publishing point URLs and the player URL are case-sensitive. It is recommended that you copy and paste these URLs when configuring your encoder and your video player.

Creating a Publishing Point

A publishing point can be created from the **Live Smooth Streaming** page. When creating a publishing point, you can set the following properties:

Properties	Description
Name	The name assigned to a publishing point is a required field. It allows a stream to be identified by an encoder and our ingest server. This name is a required component of the publishing point URL. The name assigned to a publishing point can only consist of alphanumeric characters (i.e., 0-9, A-Z, a-z), dashes, and underscores. All other characters (e.g., !, @, #, etc.), including spaces, are not allowed.
Title	The title of the video that will be streamed to your clients. A video player can be configured to display this title to your users.

Properties	Description
Duration	The estimated duration (in seconds) of the live stream. This field allows a video player's seek bar to be properly scaled to the length of the video.
DVR Duration	The desired duration (in minutes) for DVR playback. DVR playback allows a user to pause/resume, rewind, or fast forward your live event. The value assigned to this field indicates the length of the sliding window for which video will be available. This setting must be set to a value between 1 and 240. This means that the DVR playback window will be between 1 and 240 minutes.
Auth Key	An authentication key is a required field that prevents an unauthorized user from publishing content through your account. Although a default value is assigned to each new publishing point, you can set the value that will be assigned to the key. If you decide to modify the authentication key, make sure to only use whole numbers. A stream will not be published when the publishing point URL specified in the encoder contains an authentication key that does not correspond to the specified publishing point. Therefore, it is very important to paste the updated publishing point URL in your encoder's publishing point location whenever you change the publishing point name or authentication key. **Important:** Access to this key should be restricted. For more information, please refer to the **Best Practices: Live Streaming Security** section.
Expires on	Indicates the date on which the publishing point expires. A publishing point is eligible for deletion at midnight (00:00:00 GMT) on the expiration date. The default value for this option is 30 days from which it was created. This feature is useful for cleaning up old publishing points that are no longer in use. If you plan on reusing the same publishing point, then you should set a longer expiration time.
Enable iOS streaming (HLS transmuxing)	Determines whether iOS streaming capability will be added to the publishing point. If this feature is enabled, then an additional stream will be created that wraps your encoded video in a container that is supported by an HTTP Live Streaming (HLS) player. **Important:** This feature requires the use of H.264 video and AAC audio codecs. If your encoder has not been configured to use these codecs, then your live stream will not function properly. **Note:** This feature does not affect the live stream that is served to your Silverlight-compatible players. It simply adds supports for iOS devices. **Note:** A different player URL is required to stream content on an iOS device. For additional information, please refer to the **Configuring a Video Player for a CDN Ingest Point** section in this chapter.

To create a publishing point

1. Load the **Live Smooth Streaming** page.
2. Under the **Publishing Points** section, find the **Name** option. This option determines the name that will be assigned to the publishing point that will be created. Type the desired name.
3. Optional. In the **Title** option, provide a descriptive title for the video that will be streamed.
4. Optional. In the **Duration** option, indicate the estimated duration of the live stream in seconds.
5. In the **DVR Duration** option, indicate the number of minutes that will be available for DVR playback.
6. In the **Auth Key** option, review the live authentication key that will be assigned to your publishing point. If you decide to modify this key, make sure that you assign it a numeric value.
7. In the **Expires On** option, verify the date on which your new publishing point will expire.
8. Optional. If you would also like to stream to iOS devices, then mark the **Enable iOS streaming (HLS transmuxing)** option.
9. Click **Add** to create your publishing point. Your new publishing point will indicate the available publishing point and player URLs.

Reminder: It may take up to 10 minutes before all clients can connect to your newly created publishing point.

Modifying a Publishing Point

Any setting assigned to a publishing point can be modified. However, modifying certain options (i.e., publishing point name and live authentication key) will affect your publishing point and/or player URLs. These types of changes will require that you update your encoder and/or video players to reflect the new URLs.

To modify a publishing point

1. Load the **Live Smooth Streaming** page.
2. Under the **Publishing Points** section, click the pencil (✏) that appears to the left of the publishing point that you would like to modify. The settings associated with that publishing point will appear directly below the **Publishing Points** section.
3. Modify the desired settings.
4. When finished, click **Update** to save your changes. The publishing point and player URLs will be updated to reflect the changes made to your publishing point.

Reminder: It may take up to 10 minutes before the changes to your publishing point take effect.

Deleting a Publishing Point

A publishing point can be deleted by clicking ✖, which appears to the left of the desired publishing point on the **HTTP Streaming** page. When prompted, you will need to confirm the deletion of the publishing point.

Warning: A publishing point can be deleted even if there are clients connected to it. Once a publishing point is deleted, all connections to that stream will be dropped.

Reusing a Publishing Point

A single publishing point configuration may be used on multiple occasions to stream various live events. Before pointing an encoder to a previously used publishing point configuration, make sure that the following conditions have been met:

- The publishing point is not currently in use by another encoder.
- The publishing point has been shut down.
    - **MCC:** If the publishing point is shut down from within the MCC, then no further action is required.
    - **IIS:** If the publishing point is shut down from an IIS web server, then a recursive purge of the directory corresponding to the publishing point location is highly recommended. This procedure prevents old content from being accidentally served to customers connecting to the new live stream.
- **HTTP Rules Engine:** If a custom cache policy has been applied to the manifest files or the file chunks generated for a publishing point, 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 a publishing point that does not meet the above conditions may result in playback issues.

DVR Playback

DVR playback allows your clients to pause/resume, rewind, and fast forward a live stream, regardless of when that player connected to the stream. In other words, the starting point for DVR playback is determined by when your encoder starts pushing encoded video to the ingest point. The length of time that a video can be paused or to which it can be rewound depends on the value assigned to the **DVR Duration** option for the desired publishing point. In other words, the assigned value determines the number of minutes for the sliding window that will be available for DVR playback. Once the set amount of time has elapsed, content older than the specified time span will no longer be available for viewing.

Note: The maximum amount of time for DVR playback is 240 minutes. By extension, a file chunk's max-age is also limited to 240 minutes.

Note: The **DVR Duration** option determines the default max-age that will be assigned to file chunks generated for client requests. However, a rule can be created through HTTP Rules Engine that overrides this setting. This type of configuration is not recommended.

Note: It may take up to 10 minutes before changes to the DVR duration of a publishing point takes effect.

Best Practices: Live Streaming Security

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 authentication keys and then experienced unauthorized streaming on their account.

Although 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 authentication keys are never shared outside of your organization. For example, they should not be inadvertently posted on an online support form.
- Periodically change your authentication keys.
- Periodically check for publishing points that are no longer needed and delete them.
- 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.

Broadcasting Encoded Video

Once you have created a publishing point, you will need to set up your encoder. Basic encoder configuration consists of:

- Defining the location where the encoder can find the desired video feed (i.e., video source).
- Defining the video output (e.g., output format, audio/video format, bit rate levels, etc.).
- Defining the location of your publishing point.

For detailed information on how to configure your encoder, please consult the documentation provided with it.

Note: For information on encoding video, please refer to **Appendix B: Encoding Audio/Video**.

Publishing Point Configuration

An encoder's publishing point URL should always point to the Live Smooth Streaming publishing point definition file. The location of this file depends on whether our CDN or your own IIS server is ingesting the encoded video.

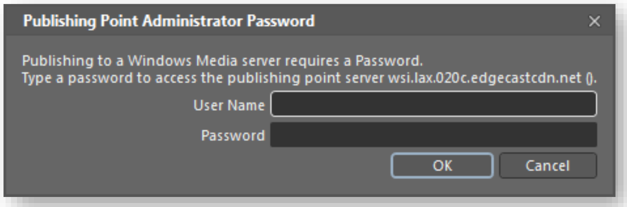
- **CDN ingest point:** A list of the available publishing point URLs can be viewed from the **Live Smooth Streaming** page. On this page, each publishing point will display three URLs that correspond to different locations around the world. You should choose the URL corresponding to the location closest to your encoder. This ensures quicker data delivery between your encoder and the CDN ingest point. Once you have selected a publishing point URL, you should copy and paste it into your encoder's publishing point location.
- **IIS ingest point:** This file was created when you added a publishing point from Internet Information Services (IIS) Manager. It should be located in the IIS virtual directory from which you added the publishing point. You are not required to use a CDN or edge CNAME URL when configuring your encoder's publishing point location. However, the use of one of these URLs is required when configuring your Smooth Stream compatible player.

Note: If the encoder reports that the publishing point is currently in use, then this typically means that your encoder was broadcasting encoded video to that location when it unexpectedly experienced an issue that interrupted the connection. This type of situation requires that you shutdown the publishing point. For more information, please refer to the **Restarting an Interrupted Stream** section.

Microsoft Expression Encoder 4

Upon starting an encoding session, Microsoft Expression Encoder 4 Service Pack 2 will prompt for credentials through which it can authenticate to our stream ingest servers. These credentials are unnecessary when using our service. Our service authenticates your encoder session through the use of an authentication key which is embedded into the publishing point URL. Therefore, it is important that you specify a valid publishing point URL when configuring your encoder. A list of current publishing point URLs can be viewed from the **Live Smooth Streaming** page in the MCC.

If Expression Encoder 4 prompts for credentials, please click **Cancel**. It will still be able to connect and publish your stream to our servers.



Publishing Point Administrator Password

Publishing to a Windows Media server requires a Password.
Type a password to access the publishing point server wsi.lax.020c.edgecastcdn.net ().

User Name

Password

Expression Encoder 4 SP2 Prompt for Credentials

Restarting an Interrupted Stream

If at any point an encoder stops pushing your live stream, then the viewing experience for the clients currently connected to the live stream may be affected. In such a case, you may not be allowed to restart your encoder until the publishing point has been shut down. The proper procedure for shutting down a publishing point depends on whether the encoded feed is being ingested by your own IIS server or directly by our CDN. The stream will be restarted once the encoder resumes pushing data to the ingest point.

Key information:

- Shutting down a publishing point will disconnect all client connections to your live stream.
- It is important to perform a recursive purge on the publishing point location prior to reusing it. If this location is not purged, then audio/video fragments from the previous event may be served to your customers. Keep in mind that purging will discard the video and audio assets stored for the purposes of DVR playback.
  - This step is automatically performed when shutting down a publishing point from within the MCC.
  - For more information on how to purge publishing point content, please refer to the **Purging Cached Manifest Files and File Chunks** topic which can be found in the **File Caching** section.
- It may take up to two minutes for a request to shut down a publishing point to take effect. You can estimate when the shutdown process has been finalized by adding 2 minutes to the time reported by the **Last Shut Down Request** option for that publishing point.
- Prevent clients from being disconnected from a stream when an ingest point experiences a point of failure by creating multiple ingest points for a single stream. This type of configuration is only recommended when using an encoder that supports pushing encoded video to multiple ingest points.
- Clients that were connected to a stream prior to the shutdown of a publishing point won't be able to continue viewing a stream until the media player sends a new request to the edge server.
  - **Example:** Refreshing a web page will cause an embedded video player to send a new request.

Shutting Down a Publishing Point (CDN Ingest Point)

A publishing point can be shut down from the **Live Smooth Streaming** page. Once you have shut down the publishing point, the date and time on which the publishing point was last shutdown will be displayed. After shutting it down, you should be able to restart encoding your live stream.

To restart an interrupted stream

1. If your encoder is still encoding video, you should disconnect it from our publishing point. One way to ensure that it is disconnected from our publishing point is to close the encoder application.
2. Load the **Live Smooth Streaming** page.
3. Under the **Publishing Points** section, click the pencil (✎) that appears to the left of the desired publishing point. The settings associated with that publishing point will appear directly below the **Publishing Points** section.
4. Click the **Shut Down** link. A message will indicate whether it was successfully shutdown.
5. Using the desired encoder, resume encoding your video feed to our publishing point.

Shutting Down a Publishing Point (IIS Ingest Point)

A publishing point can be shut down through Internet Information Services (IIS) Manager. The **Actions** pane of the **Live Smooth Streaming Publishing Points** page provides a **Shut Down Publishing Point** option. For detailed instructions on how to shut down your publishing point, please consult IIS documentation.

Reminder: After shutting down a publishing point, a recursive purge of the directory corresponding to the publishing point location (e.g., wpc.0001.edgecastcdn.net/800001/*Path*/*PublishingPoint*.isml/*) is highly recommended. This procedure prevents old content from being accidentally served to customers connecting to the new live stream.

Serving Streams through our CDN

A CDN plays a crucial process in improving the speed through which data is delivered to your clients. The first way in which a CDN helps data delivery speeds is by bypassing traditional Internet communication routes to deliver data from the ingest point to your client.

The second way is by caching files. In the case of Smooth Streaming, there are two types of file that will be delivered to your client and therefore cached. Each type of file is explained below.

The first type of file that can be cached by our edge servers is the client manifest file. A client manifest file provides the information that a video player uses to request video segments from an ingest server. The second type of file that can be cached by our edge servers is the file generated for each requested file chunk. If a file chunk has already been cached on the POP requesting it, then that edge server can serve the file directly without having to download it from an origin server. By increasing data delivery speed, your clients can enjoy higher quality uninterrupted video.

Serving Streams from a CDN Ingest Point

A CDN ingest point will serve your live stream using a server in our network. This means that it will automatically take advantage of our CDN to cache and route files to your clients.

Live Smooth Streaming is available under the HTTP Large platform, which means that it can take advantage of most of the features provided for that platform. One such feature is an edge CNAME. This feature allows you to mask the player URL. If you would like to take advantage of an edge CNAME to mask the player URL, then you will need to select the **EdgeCast Origin** option when creating it. Additionally, you will need to select the "/21*xxxx* (Live Smooth Streaming Origin)" item from the **Points to** option. For detailed information on how to create an edge CNAME, please refer to the **HTTP Large Administration Guide**.

Serving Streams from an IIS Ingest Point

In order to take advantage of our content delivery network, you will need to create a customer origin that points to the IIS server that will publish your live stream. After setting up your customer origin, you would then use a CDN URL when pointing your Smooth Stream compatible player to the Live Smooth Streaming publishing point definition file (ISML) stored on your IIS server.

Note: The customer origin pointing to your IIS server must be created for the HTTP Large platform. For more information on how to create a customer origin, please refer to the **HTTP Large Administration Guide**.

Tip: Once you have created a customer origin, you may choose to create an edge CNAME to mask the CDN URL that points to your IIS server. If you decide to create an edge CNAME, make sure to set it to the customer origin that corresponds to the desired virtual directory on your IIS server. For detailed information on how to create an edge CNAME, please refer to the **HTTP Large Administration Guide**.

Client Implementation

The final step involves creating a Smooth Stream compatible player and then implementing an instance of that player that points to your live stream. In order to help you test your Smooth Streaming configuration, a sample Smooth Streaming player is provided on the Smooth Streaming Player page of the MCC. This Smooth Streaming player allows you to visualize how the media player is able to dynamically switch between bit rates to adjust for changing network conditions and CPU usage. Additionally, a tool is provided to limit the maximum bit rate at which the player will download the stream. This provides the means through which you can simulate network congestion or excessive CPU usage.

Note: Please refer to the documentation provided for your Smooth Stream compatible player for precise instructions on how to create/configure it.

Once you have created your Smooth Streaming compatible player, you will need to generate an instance of it that points to the player URL. One way of accomplishing this is to create a web page that references it. A sample web page (i.e., SmoothStreamingStatisticsTracker_Live.html) is provided from the **Smooth Streaming Player** page. This sample web relies on the previously downloaded sample Smooth Streaming player.

Important: The sample web page (referenced above) will need to be modified before it can stream your live event. You will need to verify the location of the sample Smooth Streaming player and set the **mediaSource** parameter to the player URL. Information on how to set the **mediaSource** parameter is provided below.

Configuring a Video Player for a CDN Ingest Point

If you are taking advantage of the Live Smooth Streaming service that we provide, then you will need to point your Smooth Stream compatible player to the player URL associated with the desired publishing point. This player URL can be found directly below the available publishing point URLs for the desired publishing point on the **Live Smooth Streaming** page. Once you have copied the player URL from the desired publishing point, you will need to paste it into your Smooth Streaming compatible player.

Tip: The player URL is case-sensitive. In order to ensure that you are using the proper case, it is recommended that you copy and paste the player URL when configuring your video player.

A sample HTML code fragment for a Silverlight object is provided below.

```
mediaSource="http://wsp.xxxx.edgecastcdn.net/21xxxx/Path/FileName.isml/Manifest"
```

In the above HTML code fragment, you should replace the entire URL with the previously copied player URL. A definition is provided below for each italicized term in the above URL.

- **xxxx:** This term represent your CDN account number. Your CDN account number can be found in the upper-right hand corner of the MCC.
- **Path:** This term represents the path to the folder that contains the Live Smooth Streaming definition file.
- **FileName:** This term represents the file name of the Live Smooth Streaming publishing point definition file. The naming convention used for this file is: *PublishingPointName*.isml.

Tip: An alternative to the above player URL is to use an edge CNAME to mask it. This type of configuration requires that you replace the CDN domain and content access point (e.g., http://wsp.0001.edgecastcdn.net/210001/) with the edge CNAME domain (e.g., http://www.mydomain.com/). Before using an edge CNAME, verify that it points to "/21*xxxx* (Live Smooth Streaming Origin)."

Setting up a Video Player for iOS Streaming

In order to stream content to iOS devices, you will need to first make sure of the following:

- **Publishing Point:** The **Enable iOS Streaming (HLS Transmuxing)** option has been enabled on the desired publishing point.
- **Encoder:** An encoder is generating H.264 video and AAC audio codecs from your live event.
- **Video Player:** A video player must point to the manifest file using an iOS-specific player URL instead of the player URL provided on the **Live Smooth Streaming** page.

As mentioned above, an iOS-specific player URL is required when streaming content to an iOS device. The syntax for this URL is provided below.

> http://wsp.*xxxx*.edgecastcdn.net/21*xxxx*/*PublishingPoint*.isml/Manifest(format=m3u8-aapl).m3u8

In the above URL, you should replace the following terms:

- **xxxx:** This term represent your CDN account number. Your CDN account number can be found in the upper-right hand corner of the MCC.
- **PublishingPoint:** This term represents the name assigned to the desired Smooth Streaming publishing point. This publishing point must have the **Enable iOS Streaming (HLS Transmuxing)** option enabled.

The above link will automatically launch the stream corresponding to the live event in full screen mode. If you would like to define additional video options, then you should use an HTML 5 video tag. A sample HTML 5 tag is provided below.

```
<video width="320" height="240" controls="controls">

     <source
src="http://wsp.0001.edgecastcdn.net/210001/MyPublishingPoint.isml/Manifest(format=m
3u8-aapl).m3u8" />

</video>
```

Tip: An alternative to the above player URL is to use an edge CNAME to mask it. This type of configuration requires that you replace the CDN domain and content access point (e.g., http://wsp.*xxxx*.edgecastcdn.net/21*xxxx*/) with the edge CNAME domain (e.g., http://www.mydomain.com/). During configuration, verify that the edge CNAME points to "/21*xxxx* (Live Smooth Streaming Origin)."

Configuring a Video Player for an IIS Ingest Point

Once you have created your various bit rate streams, hosted it on IIS, and properly configured our CDN to recognize the IIS virtual directory, you are ready to make it available to your clients.

Note: The player URL is case-sensitive. This means that you should take special care to ensure that the proper case is used.

A sample HTML code fragment for the Silverlight object is provided below.

```
mediaSource="http://wpc.xxxx.edgecastcdn.net/80xxxx/Path/FileName.isml/Manifest"
```

In the above HTML code fragment, 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.
- **Path:** This term should be replaced by the path to the folder where the Live Smooth Streaming publishing point definition file is stored. This path should start with the name of the customer origin corresponding to the IIS server. The path to the desired folder should then be specified directly after it.
- **FileName:** This term should be replaced by the file name of the Live Smooth Streaming publishing point definition file.

Tip: An alternative to the above player URL is to use an edge CNAME to mask it. This type of configuration requires that you replace the CDN domain and content access point (e.g., http://wpc.0001.edgecastcdn.net/800001/CustomerOrigin) with the edge CNAME domain (e.g., http://www.mydomain.com/). Make sure to append the full path to the Live Smooth Streaming publishing point definition file.

File Caching

There are two different types of files that our POPs will cache, which are the client manifest file and file chunks. The default amount of time for which these files will be cached may depend on whether you are publishing your stream using your own IIS server or you using our Live Smooth Streaming service. The amount of time that a file will be cached is determined by the header information assigned to it. However, these cache headers can be automatically overwritten by HTTP Rules Engine. Finally, you can shorten the amount of time a stream will be cached by purging it at any time from our CDN. Please refer to the appropriate section below.

Cache Headers for Files Generated from a CDN Ingest Point

This section explains how our CDN caches the client manifest file and the file chunks requested by your video players. Both of these files are assigned a default max-age, which determines how long they will remain cached on an edge server after a client request.

A client manifest file has a default max-age of 2 seconds. This short TTL for the client manifest file is due to the fact that it is constantly being updated with references to new video segments created for your live stream. This short lifespan allows a new client to retrieve an updated version of the file from which it can generate file chunk requests.

Our CDN also caches each file chunk extracted from your audio/video files (i.e., ISMV and ISMA). The default max-age that will be assigned to each file chunk is determined by the **DVR Duration** option set for the corresponding publishing point.

Cache Headers for Files Generated from an IIS Ingest Point

Internet Information Services (IIS) Manager allows you to configure how server and client caching will be handled for all live smooth streams. Server caching determines the length of time for which the metadata for a live smooth stream will be cached, while client caching determines the header information that will be assigned to the file chunks generated for live smooth streams. Among the header information that can be assigned to file chunks is max-age. Please refer to IIS documentation for a complete list of supported headers.

Note: By default, our CDN respects the header information assigned to files generated from your IIS server. However, you can create rules that will override the header information generated by IIS.

Purging Cached Manifest Files and File Chunks

The manifest file and the file chunks associated with a particular live smooth stream can be purged from all of our POPs. This means that the cached version of those files will be permanently deleted from our edge servers. All future requests for those files will cause our edge servers to forward the request to the origin server, which will either be an ingest point on our CDN or your own IIS origin server. Since all file chunks will have been purged from our CDN, DVR playback will not be available for the time frame covered by those files.

You can purge an entire live smooth stream from our CDN. This means that the client manifest file and the requested file chunks will be purged from all of our POPs worldwide. The proper purge URL that you should use depends on whether you are using a CDN or an IIS ingest point.

Please select the purge URL that applies to your configuration.

- **Base Purge URL for a CDN Ingest Point:**
  http://wpc.*xxxx*.edgecastcdn.net/21*xxxx*
- **Base Purge URL for an IIS Ingest Point:**
  http://wpc.*xxxx*.edgecastcdn.net/80*xxxx*/*CustomerOrigin*

The CDN URL path that should be appended to the above path is noted below.

- **Purge path for a CDN Ingest Point:**
  /*StreamName*.isml/*
- **Purge path for an IIS Ingest Point:**
  /*Path*/*StreamName*.isml/*
  The term *Path* represents the path to the folder that contains your Live Smooth Stream publishing point definition file. For example, if an edge CNAME points to the IIS virtual directory and the Live Smooth Streaming publishing point definition file is located in a subfolder called "My_Streams," then you should use the following CDN URL path: /My_Streams/StreamName.isml/*.

Note: For detailed information on how to purge files, please refer to the **HTTP Large Administration Guide**.

How HTTP Rules Engine Can Affect Live Smooth Streaming

HTTP Rules Engine is a very useful tool that allows you to customize how our CDN handles file caching. Due to the inherent power of this tool, you may create a configuration that prevents your live smooth streams from being properly cached. For example, if you create a global rule that overrides the default internal and/or external max-age to 2 minutes, then this may potentially cause various issues with Live Smooth Streaming. The first consequence of this type of configuration is that the client manifest file retrieved by new clients may be out of date. This may prevent those clients from knowing which time segment to request. Additionally, the requested file chunks will all have a max-age of 2 minutes. This means that DVR playback will only be available for 2 minutes, regardless of the DVR duration that was specified for your publishing points.

Reminder: The HTTP Rules Engine feature can also be used to make Smooth Streaming compatible with Token-Based Authentication. Please refer to the **Interaction with the HTTP Large Platform** section above.

On-Demand Smooth Streaming

Overview

The On-Demand Smooth Streaming feature provides adaptive streaming of a previously recorded event. In order for clients to take advantage of adaptive streaming, they will need to request the server manifest file from a server that is publishing the archived video as a stream. The requested file chunks will be cached on that POP creating an environment where future similar requests are served at an even quicker rate.

In order to provide smooth streaming capabilities to your clients, you should perform the following steps:

1. Encode your video into a Smooth Stream compatible video format. For information on encoding video, please refer to **Appendix B: Encoding Audio/Video**.
2. Decide whether you would like to use our on-demand smooth streaming service or your own IIS server. This publishing location, which is either a server on our CDN or your own IIS server, is the ingestion point for your encoded video. The location where you can store your video depends on the selected ingestion point.
    - **CDN:** The desired smooth stream presentation must be located on a CDN origin server.
    - **IIS Server:** Make sure that the presentation is located in an IIS virtual directory. Additionally, you will need a customer origin that points to an IIS virtual directory. If that customer origin does not exist, then you should create one.
3. If you would like to mask the CDN URL that points to the origin server, then you should create an edge CNAME for it. You will also need to define the specified host name on a DNS server.
4. Point your Smooth Stream compatible player to the server manifest file for the desired smooth stream using a CDN or an edge CNAME URL.

Each of the above steps is explained in this chapter.

Note: On-Demand Smooth Streaming does not transmux video into an M3U8 format. As a result, an iOS device cannot stream content through On-Demand Smooth Streaming. However, if you have properly segmented/packaged the desired content and uploaded it to CDN storage or a customer origin server, you can stream it to iOS devices using a standard CDN URL or edge CNAME URL (e.g., http://wpc.0001.edgecastcdn.net/800001/2012/playlist.m3u8).

Setting up an Ingest Point

Although On-Demand Smooth Streaming takes advantage of existing audio/video files (i.e., ISMV and ISMA files), it still requires an ingest point to translate client requests into a file chunk that corresponds to the correct bit rate and time span. This ingest point can either be your own IIS server or it can be located on our network. Both types of configurations are explained below.

Note: Regardless of the ingest point used to publish your on-demand content, the server manifest file will need to be served to your clients using a CDN or an edge CNAME URL. For more information, please refer to the **Serving On-Demand Content through our CDN** section below.

Publishing On-Demand Content through our CDN

Before you can stream on-demand content from our network, you will need to enable the On-Demand Smooth Streaming feature. This can be performed from the **On-Demand Smooth Streaming** page. Once you have enabled this feature, it may take up to an hour for it to take effect. It is strongly recommended that you wait until this feature has been fully propagated throughout our network before making on-demand content available to your clients.

Publishing On-Demand Content through your own IIS Server

An alternative to taking advantage of our Smooth Streaming services is to publish your on-demand content through your own IIS server. This type of configuration requires that your Smooth Streaming presentation be located in an IIS virtual directory.

Note: For information on how to create or configure an IIS virtual directory, please refer to IIS documentation.

Serving On-Demand Content through our CDN

A CDN plays a crucial process in improving the speed through which data is delivered to your clients. The first way in which a CDN helps data delivery speeds is by bypassing traditional Internet communication routes to deliver data from the ingest point to your client.

The second way is by caching files. In the case of Smooth Streaming, there are two types of file that will be delivered to your client and therefore cached. The first type of file is the client manifest file. The second type of file is the file generated for each requested file chunk. If a requested file chunk has already been cached on the POP requesting it, then that edge server can serve the file directly without having to download it from an origin server. By increasing data delivery speed, your clients can enjoy higher quality uninterrupted video.

Serving Streams from a CDN Ingest Point

Before you can serve a stream, you will need to store your Smooth Stream presentation on a CDN origin server. Once you have uploaded the required files, a CDN ingest point can serve your on-demand content using a server in our network. This allows you to take advantage of our CDN to cache and route Smooth Stream files to your clients. The CDN URLs that you can use to publish on-demand content can be viewed from the **On-Demand Smooth Streaming** page. This list consists of a CDN URL to the base on-demand ingest point and an edge CNAME URL for each edge CNAME that points to that URL.

Reminder: On-Demand Smooth Streaming can only publish Smooth Stream presentations that are located on a CDN origin server. If you would like to use a customer origin server, then you will need to make sure that it is stored and published from your own IIS origin server.

An edge CNAME allows you to mask the player URL. If you decide that you would like to mask a CDN URL, then you will need to make sure to select the **EdgeCast Origin** option when creating the edge CNAME. Additionally, you will need to select the "/01*xxxx* (On-Demand Smooth Streaming Origin)" item from the **Points to** option. For detailed information on how to create an edge CNAME, please refer to the **HTTP Large Administration Guide**.

Each player URL consists of a base URL that can be used to provide adaptive streaming to your clients. Before you can provide on-demand smooth streaming to your clients, you will need to append the full path to the desired server manifest (ISM) file. An example for each type of player URL is provided below.

- **CDN URL:**
  http://wpc.*xxxx*.edgecastcdn.net/01*xxxx*/*Path*/*FileName*.ism/Manifest
- **Edge CNAME URL:**
  http://*EdgeCNAME*/*Path*/*FileName*.ism/Manifest

A definition is provided below for each italicized term in the above URLs.

- **EdgeCNAME:** This term represents the host name associated with the edge CNAME created for the CDN ingest point. This host name must also be defined on a DNS server.
- **xxxx:** This term represent your CDN account number. Your CDN account number can be found in the upper-right hand corner of the MCC.
- **Path:** This term represents the path to the folder that contains the desired Smooth Stream presentation. If your files are located in the root folder, then you should omit this term.
- **FileName:** This term represents the file name of the server manifest file. The naming convention used for this file is: *PublishingPointName*.ism.

Serving Streams from an IIS Ingest Point

Once you have stored your Smooth Stream presentation in an IIS virtual directory, you will need to create a customer origin for the HTTP Large platform. This customer origin should point to an IIS virtual directory. A customer origin is what allows your files to be routed throughout or network and cached on our edge servers. For detailed information on how to create a customer origin, please refer to the **HTTP Large Administration Guide**.

Once you have created a customer origin for your IIS virtual directory, a CDN URL to that location will appear on the **HTTP Large** page. You can point your Smooth Stream compatible players to your presentation using this CDN URL. Alternatively, you can mask this CDN URL as a way to provide a friendlier player URL. In order to mask a CDN URL, you will need to create an edge CNAME to the corresponding customer origin. When creating an edge CNAME, you will need to make sure to select the **Customer Origin** option when creating the edge CNAME. Additionally, you will need to select the content access point that corresponds to the desired customer origin from the **Points to** option. For detailed information on how to create an edge CNAME, please refer to the **HTTP Large Administration Guide**.

Note: If you decide to create an edge CNAME, make sure to define the host name specified by that edge CNAME on a DNS server.

Client Implementation

The final step involves creating a Smooth Stream compatible player and then implementing an instance of that player that points to your on-demand content. In order to help you test your Smooth Streaming configuration, a sample Smooth Streaming player is provided on the Smooth Streaming Player page of the MCC. This Smooth Streaming player allows you to visualize how the media player is able to dynamically switch between bit rates to adjust for changing network conditions and CPU usage. Additionally, a tool is provided to limit the maximum bit rate at which the player will download the stream. This provides the means through which you can simulate network congestion or excessive CPU usage.

Note: Please refer to the documentation provided for your Smooth Stream compatible player for precise instructions on how to create/configure it.

Note: On-Demand Smooth Streaming is not compatible with iOS devices. An alternative solution for streaming on-demand content to an iOS device is to convert your content to an iOS-compatible format and then link to your content over the HTTP Large platform.

Once you have created your Smooth Streaming compatible player, you will need to generate an instance of it that points to the player URL. One way of accomplishing this is to create a web page that references it. A sample web page (i.e., SmoothStreamingStatisticsTracker_OnDemand.html) is also provided from the **Smooth**

Streaming Player page. This sample web relies on the previously downloaded sample Smooth Streaming player.

Important: The sample web page (referenced above) will need to be modified before it can stream your on-demand content. You will need to verify the location of the sample Smooth Streaming player and set the **mediaSource** parameter to the player URL. Information on how to set the **mediaSource** parameter is provided below.

Configuring a Video Player for a CDN Ingest Point

If you are taking advantage of the On-Demand Smooth Streaming service that we provide, then you will need to point your Smooth Stream compatible player to the server manifest file for the desired smooth stream. The base player URL can be found on the **On-Demand Smooth Streaming** page. Once you have copied the player URL, you will need to paste it into your Smooth Streaming compatible player. After which, you will need to append "/*Path*/*ManifestName*.ism/Manifest" to that URL. A sample HTML code fragment for a Silverlight object is provided below.

```
mediaSource="http://wpc.xxxx.edgecastcdn.net/01xxxx/Path/ManifestName.ism/Manifest"
```

In the above HTML code fragment, you should replace the entire URL with the previously copied player URL. A definition is provided below for each italicized term in the above URL.

- **xxxx:** This term represent your CDN account number. Your CDN account number can be found in the upper-right hand corner of the MCC.
- **Path:** This term represents the path to the folder that contains the server manifest file.
- **ManifestName:** This term represents the file name of the Smooth Streaming server manifest file.

Tip: An alternative to the above player URL is to use an edge CNAME to mask it. This type of configuration requires that you replace the CDN domain and content access point (e.g., http://wpc.0001.edgecastcdn.net/010001/) with the edge CNAME domain (e.g., http://www.mydomain.com/).

Configuring a Video Player for an IIS Ingest Point

Once you have created your various bit rate streams, hosted it on IIS, and properly configured our CDN to recognize the IIS virtual directory, you are ready to make it available to your clients. Sample HTML code fragments for the Silverlight object are provided below.

```
mediaSource="http://wpc.xxxx.edgecastcdn.net/80xxxx/Path/ManifestName.ism/Manifest"
```

In the above HTML code fragment, 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.
- **Path:** This term should be replaced by the path to the folder where the Smooth Streaming server manifest file is stored. This path should start with the customer origin corresponding to the IIS server. The path to the desired folder should then be specified directly after it.
- **ManifestName:** This term should be replaced by the file name of the Smooth Streaming server manifest file.

Tip: An alternative to the above player URL is to use an edge CNAME to mask it. This type of configuration requires that you replace the CDN domain and content access point (e.g., http://wpc.0001.edgecastcdn.net/800001/CustomerOrigin) with the edge CNAME domain (e.g., http://www.mydomain.com/). Make sure to append the full path to the desired Smooth Streaming server manifest file.

File Caching

There are two different types of files that our POPs will cache, which are the client manifest file and file chunks. The default amount of time for which these files will be cached may depend on whether you are publishing your stream using your own IIS server or you using our On-Demand Smooth Streaming service. The amount of time that a file will be cached is determined by the header information assigned to it. However, these cache headers can be automatically overwritten by HTTP Rules Engine. Finally, you can shorten the amount of time a stream will be cached by purging it at any time from our CDN. Please refer to the appropriate section below.

Tip: If you have updated your on-demand content or if you suspect that the cached version of your on-demand content has been corrupted, then we highly recommend that you purge your on-demand content as indicated in the **Purging Cached Manifest Files and File Chunks** section below.

Cache Headers for Files Generated from a CDN Ingest Point

This section explains how our CDN caches the client manifest file and the file chunks requested by your video players. Both of these files are assigned a default max-age, which determines how long they will remain cached on an edge server after a client request.

A client manifest file has a default max-age of 2 hours. However, since this is a static file, new requests for a stale client manifest file will cause the edge server to serve it from cache.

Our CDN also caches each file chunk extracted from your audio/video files (i.e., ISMV and ISMA). The default max-age that will be assigned to each file chunk is 2 hours.

Cache Headers for Files Generated from an IIS Ingest Point

Internet Information Services (IIS) Manager allows you to configure how server and client caching will be handled for all on-demand smooth streams. Server caching determines the length of time for which the metadata for an on-demand smooth stream will be cached, while client caching determines the header information that will be assigned to the file chunks generated for on-demand smooth streams. Among the header information that can be assigned to file chunks is max-age. Please refer to IIS documentation for a complete list of supported headers.

Note: By default, our CDN respects the header information assigned to files generated from your IIS server. However, you can create rules that will override the header information generated by IIS.

Purging Cached Manifest Files and File Chunks

The manifest file and the file chunks associated with a particular smooth stream can be purged from all of our POPs. This means that the cached version of those files will be permanently deleted from our edge servers. All future requests for those files will cause our edge servers to forward the request to the origin server, which will either be an ingest point on our CDN or your own IIS origin server. Since all file chunks will have been purged from our CDN, DVR playback will not be available for the time frame covered by those files.

You can purge an entire on-demand smooth stream from our CDN. This means that the client manifest file and the requested file chunks will be purged from all of our POPs worldwide. The proper purge URL that you should use depends on whether you are using a CDN or an IIS ingest point.

Please select the purge URL that applies to your configuration.

- **Base Purge URL for a CDN Ingest Point:**
  http://wpc.*xxxx*.edgecastcdn.net/01*xxxx*
- **Base Purge URL for an IIS Ingest Point:**
  http://wpc.*xxxx*.edgecastcdn.net/80*xxxx*/*CustomerOrigin*

The CDN URL path that should be appended to the above path is noted below.

- **Purge path for a CDN Ingest Point:**
  /*Path*/*StreamName*.ism/*
- **Purge path for an IIS Ingest Point:**
  /*Path*/*StreamName*.ism/*

Note: The term *Path* represents the path to the folder that contains your On-Demand Smooth Stream server manifest file.

Note: For detailed information on how to purge files, please refer to the **HTTP Large Administration Guide**.

Appendix A

Interaction with Token-Based Authentication

By default, a location secured by Token-Based Authentication is not compatible with Smooth Streaming. This compatibility issue arises due to the nature of how Smooth Streaming delivers content to your clients. For each stream, Smooth Streaming generates a manifest file and the ISMV/ISMA assets associated with each bit rate feed. The manifest file provides a list of ISMV/ISMA assets that the client will use when requesting the next video segment. If Token-Based Authentication has been enabled on the publishing location, then a valid token will be required for the manifest file and for each of those ISMV/ISMA assets. It is a relatively simple matter to specify a valid token for the manifest file, since you can define this value when implementing your Silverlight player. However, a default manifest file does not include such a token when it lists the available ISMV/ISMA assets. This creates a situation where a client can be authorized to access the manifest file but be denied access to the file chunks. As a result, the player would get a listing of the available streams but be unable to play any of them.

Note: On-demand content takes advantage of a static manifest file. As a result, you can easily add the proper token values to each file chunk either through a script or by manually appending them. However, the recommended practice is to bypass Token-Based Authentication security on file chunks, regardless of whether you are streaming live or on-demand content.

Appendix B

Encoding Audio/Video

The first step that you need to take to stream live or on-demand content is to encode your audio and/or video. Before you can encode your audio/video, you will need to choose the quality of each bit rate stream that will be made available to your clients. Some general guidelines on how to choose bit rate quality are provided below.

- **Low quality:** This stream should not require much in terms of bandwidth or CPU usage. All of your clients should be able to view this type of stream without experiencing delays due to buffering, disconnects, or playback stutter.
- **Balance between quality and bandwidth/CPU requirements:** This type of stream will cater to most of your users, since it will be the highest quality video that they can view with their limited bandwidth or hardware. Most of your streams should fall under this category.
- **High quality:** This stream should provide an optimal viewing experience to demanding clients that expect higher quality due to their available bandwidth and computer hardware.

Tip: Smooth Streaming excels when it can choose between various bit rate streams. This allows it to match video quality more closely to a user's environment. However, keep in mind that increasing the number of bit rate streams also increases CPU usage and the minimum amount of Internet bandwidth capacity for the computer hosting the encoder.

Note: Microsoft Expression Encoder automatically chooses default bit rates according to the selected output format. However, you may always add, modify, or delete the existing default bit rates.

Once you have chosen which bit rate streams will be created, the encoder will convert your audio/video into a MP4 file for each one of them. The type of MP4 file that will be created depends on whether the target media is audio & video or audio only. If you have an audio/video stream, then an ISMV file will be generated. If your stream is audio only, then an ISMA file will be generated. In addition to the MP4 file, the encoder will also generate a server manifest file (ISM) and a client manifest file (ISMC). The server manifest file provides information about the stream, such as the available bit rate streams and the files to which they are mapped, to IIS. The client manifest file provides the Silverlight player with the information it needs to dynamically switch between different bit rate streams.

Reminder: In order to stream a live event to iOS devices, it must be encoded as H.264 video and AAC audio codecs.