



Akamai® Streaming

(for Live Adobe® Flash® Video)

User Guide

Flash Media Server 3.5 (Akamai Version 7)

Akamai Confidential
For Customer Use Under NDA Only

November 2, 2010

Akamai Technologies, Inc.

Akamai Customer Care: **1-877-425-2832** or, for routine requests, e-mail **ccare@akamai.com**

The EdgeControl® portal, for customers and resellers: **<http://control.akamai.com>**

US Headquarters
8 Cambridge Center
Cambridge, MA 02142

Tel: 617.444.3000
Fax: 617.444.3001

US Toll free 877.4AKAMAI (877.425.2624)

For a list of offices around the world, see:
<http://www.akamai.com/en/html/about/locations.html>

Akamai® Streaming (for Live Adobe® Flash® Video) User Guide

Copyright © 2006–2008 Akamai Technologies, Inc. All Rights Reserved.

Reproduction in whole or in part in any form or medium without express written permission is prohibited. Akamai, the Akamai wave logo, and the names of Akamai services referenced herein are trademarks of Akamai Technologies, Inc. Other trademarks contained herein are the property of their respective owners and are not used to imply endorsement of Akamai or its services. While every precaution has been taken in the preparation of this document, Akamai Technologies, Inc. assumes no responsibility for errors, omissions, or for damages resulting from the use of the information herein. The information in these documents is believed to be accurate as of the date of this publication but is subject to change without notice. The information in this document is subject to the confidentiality provisions of the Terms & Conditions governing your use of Akamai services.

Adobe, ActionScript, Flash, and the Flash logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Adobe product screen shot(s) reprinted with permission from Adobe Systems Incorporated.

On2, VP6, and Flix are either registered trademarks or trademarks of On2 Technologies, Incorporated. On2 Flix Live 8 screen shots are reprinted by permission from On2 Technologies, Incorporated.

Microsoft, Windows, and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The QuickTime logo is a trademark of Apple Inc., registered in the U.S. and other countries.

The Real logo is a registered trademark of RealNetworks, Inc.

All other product and service names mentioned herein are the trademarks of their respective owners.

Contents

PREFACE • 1

About This Document	1
Other Resources	2

CHAPTER 1. INTRODUCING AKAMAI STREAMING (FOR LIVE ADOBE FLASH VIDEO) • 5

About Adobe Flash Video Streaming	5
How Akamai Streaming Works	6

CHAPTER 2. GUIDELINES AND BEST PRACTICES • 7

CHAPTER 3. PROVISIONING AKAMAI STREAMING (FOR LIVE ADOBE FLASH VIDEO) • 15

Accessing Your Akamai Streaming Account	15
About the Manage Streams Page	17
Creating Configurations	18
About Live Configuration Details	21
Creating Streams	22
Viewing Stream Details	30
Retrieving the E-Type Token Binary	31
Editing Configurations	32
Editing Streams	33
Modifying Secure Streaming	35

CHAPTER 4. SETTING UP THE VIDEO ENCODER APPLICATION • 39

Using Dynamic Streaming	40
Encoding Your Videos for Dynamic Streaming	40
Using Adobe Flash Media Live Encoder and On2 Flix Live	41
Using High-Bit-Rate Streams Over High-Latency Connections	41
Using Adobe Flash Media Live Encoder	42
Using On2 Flix Live	45
Using the AkamaiFCSPublish Class	47
Creating a Simple Application for Encoding Live Flash Video	48
Applying a Broadcast Time Limit	50
Using the Akamai Broadcaster Encoding Tool	51

CHAPTER 5. TESTING LIVE FLASH STREAMS • 55

Testing with the Flash Video Test Player	55
Troubleshooting with the Flash Video Test Player	57
Using Trace Data	57
Overriding the Server IP	57

CHAPTER 6. SETTING UP THE VIDEO PLAYBACK APPLICATION • 59

Akamai Streaming (for Live Adobe Flash Video) Communications	59
Creating a Flash Playback Application	60
Creating a Flash Playback Application for Dynamic Streaming	60
Building a Playback Application Using the AkamaiConnection Class	61
Building a Playback Application Without the AkamaiConnection Class	61
A Note on the NetStream.onStatus Event Handler	64

Expediting the Connection	65
Dealing with Problematic Client-Side Proxy Servers	67
Solving the Problem Using the AkamaiConnection Class	67
Solving the Problem Without the AkamaiConnection Class	68
CHAPTER 7. USING SECURE STREAMING • 71	
Secure Streaming Guidelines for Flash Video	71
Using an slist to Handle Secure Streaming/Dynamic Streaming	72
Implementing an slist.	72
Integrating Secure Streaming for Live Flash Video	73
Passing the Token to Akamai Streaming	73



Preface

Akamai® Streaming (for Live Adobe® Flash® Video) combines the technologies of both Akamai and Adobe Systems Incorporated to offer customers a live streaming Flash video solution that rounds out the Akamai Streaming suite.

About This Document

This guide provides an overview of Akamai Streaming (for Live Adobe Flash Video), as well as details regarding its setup and use. It is intended for customers who will be provisioning the service for use with their Web properties. It also provides information regarding the actual configuring of Flash video encoder applications and playback applications, and will be of use to developers of those applications.

This document is organized into chapters as follows:

Chapter 1. Introducing Akamai Streaming (for Live Adobe Flash Video) gives an overview of the Akamai Streaming service.

Chapter 2. Guidelines and Best Practices provides tips, guidance, and suggestions regarding the Akamai Streaming (for Live Adobe Flash Video) service.

Chapter 3. Provisioning Akamai Streaming (for Live Adobe Flash Video) provides procedures for using the EdgeControl® portal to prepare your Akamai Streaming account for operation.

Chapter 4. Setting Up the Video Encoder Application discusses options and methods for encoding your Flash video for live streaming on the Akamai Streaming service.

Chapter 5. Testing Live Flash Streams describes using Akamai's Flash Video Test Player to test and troubleshoot your live streams on the Akamai Streaming service.

Chapter 6. Setting Up the Video Playback Application discusses options and methods for setting up your playback Flash applications to receive live streaming video from the Akamai Streaming service.

Chapter 7. Using Secure Streaming outlines the Secure Streaming feature as it applies to live Flash video streams, providing you additional control over which end users access your content.

Other Resources

Additional information regarding the following Akamai products is available from the EdgeControl portal's **Documentation** area (<https://control.akamai.com>).

Akamai Streaming (for Adobe Flash Video) Tools ([Live Streams >> Documentation](#))

- AkamaiConnection Class (Adobe ActionScript® 2.0 class)
- AkamaiConnection Class (Adobe ActionScript 3.0 class)
- AkamaiFCSPublish Class (Adobe ActionScript 2.0 class)
- AkamaiBroadcaster Live Video Encoding Tool

Akamai Secure Streaming ([Live Streams >> Documentation](#))

- *Akamai Secure Streaming Integration Guide*

Akamai Traffic Reports ([Documentation >> Traffic Management](#))

- “Reports Help”
- *Akamai Data Reference*

Documents and Articles Regarding Adobe Flash Software and Video

Adobe Flash CS3 and Flash 8

- Flash CS3 (<http://www.adobe.com/support/documentation/en/flash/>)
- Flash 8 (<http://www.adobe.com/support/documentation/en/flash/documentation.html>)

Dynamic Streaming

- *Dynamic Streaming in Flash Media Server 3.5—Part 1: Overview of the New Capabilities* (http://www.adobe.com/devnet/flashmediaserver/articles/dynstream_advanced_pt1.html)
- *Dynamic Streaming in Flash Media Server 3.5—Part 2: ActionScript 3.0 Dynamic Stream API* (http://www.adobe.com/devnet/flashmediaserver/articles/dynstream_advanced_pt2.html)—This document is for those who plan to build their own client dynamic code.
- *Dynamic Streaming in Flash Media Server 3.5—Part 3: Integrating Dynamic Streaming with Existing Video Players* (http://www.adobe.com/devnet/flashmediaserver/articles/dynstream_advanced_pt3.html)

This document is for those who intend to incorporate their own code into an existing video player.

- *Live Dynamic Streaming with Flash Media Server 3.5* (http://www.adobe.com/devnet/flashmediaserver/articles/dynstream_live.html)

This document includes best encoding practices.

Supported Media Formats

- *List of Codecs Supported by Adobe Flash Player* (<http://kb.adobe.com/selfservice/viewContent.do?externalId=kb402866&sliceId=2>)
- *Flash Video Primer* (http://www.adobe.com/devnet/flash/articles/flash_flv.pdf)
- *Encoding Best Practices for Live Flash Video* (http://www.adobe.com/devnet/flash/articles/flv_live.html)
- *Flash Video Learning Guide* (http://www.adobe.com/devnet/flash/articles/video_guide.html)
- *Exploring Flash Player Support for High-Definition H.264 Video and AAC Audio* (http://www.adobe.com/devnet/flashplayer/articles/hd_video_flash_player.html)
- *Using the FLVPlayback Component with Flash Player 9 Update 3* (http://www.adobe.com/devnet/flash/articles/flvplayback_fplayer9u3.html)
- *MIME Type Registration for MPEG-4* (<http://www.rfc-archive.org/getrfc.php?rfc=4337>)
- *New File Extensions and MIME Types* (<http://www.kaourantin.net/2007/10/new-file-extensions-and-mime-types.html>)

Chapter 1. Introducing Akamai Streaming (for Live Adobe Flash Video)

In This Chapter



About Adobe Flash Video Streaming • 5

How Akamai Streaming Works • 6

Akamai Streaming (for Live Adobe Flash Video) leverages Akamai's network of thousands of servers to deliver live Flash video streams. Flash videos originating from a video encoder are transferred to the distributed Akamai Streaming service to enable delivery of live streams to end users from the edge of the Internet.

About Adobe Flash Video Streaming

Flash video provides a high-quality video format for online Web delivery. Live streaming Flash video offers a number of advantages, including low-latency playback. In addition, Akamai Streaming (for Adobe Flash Video) supports MP3 audio, as well as the MPEG-4 standards H.264 video and HE-AAC audio.



*Note: Unless specified otherwise, the terms **Flash video** and **video** are used to collectively refer to all supported video and audio formats.*

In addition to the video itself, the following are required to stream live Flash video:

- **Video Encoder.** This tool encodes your audio and video sources into your format of choice and pushes the resulting video stream to Akamai Streaming's Entry Point servers. Unlike other streaming media formats, live Flash video does not necessarily require a standardized video encoder, though third-party encoders are available from Akamai, Adobe Systems Incorporated, and On2 Technologies, Inc. Instead, the encoder can be a Flash application (identified by the ".swf" file extension) that your Flash developer can create from scratch, allowing it to have as much functionality and as many features as you desire.
- **Flash Playback Application.** This file is also a Flash application (.swf) file that your Flash application developer creates. It resides on the Web server and is called by HTML in your Web page. This starts Adobe Flash Player, which reacts to script (ActionScript) written into the application, establishes a connection with the Akamai Streaming service, and requests the live video stream.
- **Adobe Flash Player (Version 6 or Higher (Version 9.0.115.0 or Higher for H.264 Video and HE-AAC Audio Streaming)).** This is a rich client that plays Flash applications and communicates with Akamai Streaming to enable Flash video streaming. It is downloaded by the end user and installed on his or her computer.

How Akamai Streaming Works

A Flash video encoder application encodes raw video from audio and video sources into Flash-formatted video and pushes the stream to an Akamai Entry Point. It is then sent to the Akamai Streaming servers, which accept the stream and wait for end-user requests.

When an end user's Flash Player requests the video stream, the request resolves, using Akamai's intelligent routing technology, to an Akamai Streaming server located optimally for that end user, which then streams the live video to the end user. This sequence of events distributes your video around the world on Akamai's global distributed network.

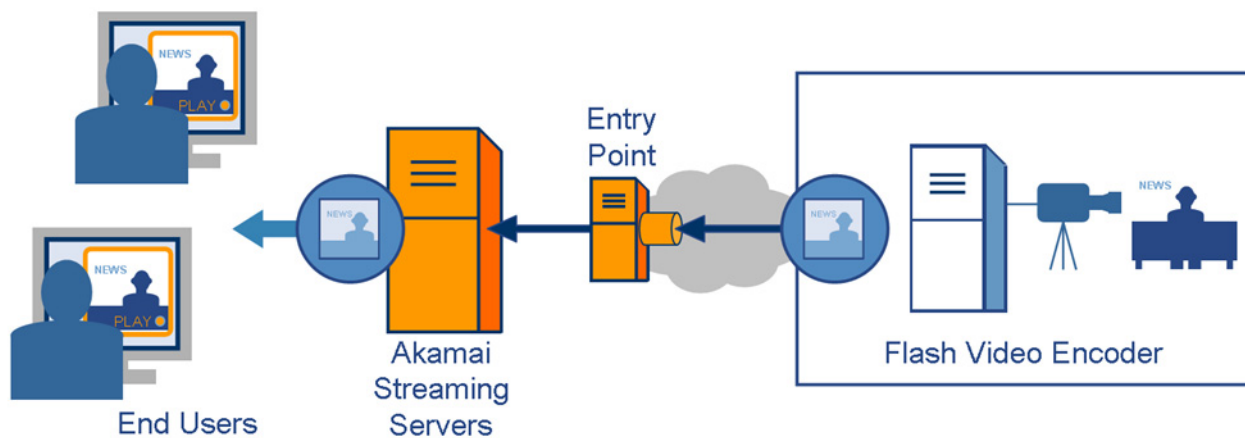



Figure 1-1. Akamai Streaming Service (for Live Adobe Flash Video)

Chapter 2. Guidelines and Best Practices

There are several important considerations you should take into account when using the Akamai Streaming (for Live Adobe Flash Video) service.

 *Note: The following is not intended to be a complete list of all precautions you should take or considerations you should evaluate in using the service. If you have questions, please consult with an Akamai Customer Care representative.*

Use Akamai's AkamaiConnection class

The AkamaiConnection class is available as both an ActionScript 2.0 and ActionScript 3.0 class that you can download from the EdgeControl portal. It is designed to assist Flash application developers in establishing a robust connection with the Akamai Streaming (for Adobe Flash Video) service using current best practices for connection expedition, client-proxy penetration, and buffer management. Its use is strongly recommended and is discussed in Chapter 6.

If not using the AkamaiConnection class, include ActionScript to automate the connection expedition and client-side proxy solutions

If you decide not to use Akamai's AkamaiConnection class to build your client-side Flash application, it is strongly recommended you include ActionScript in your application to expedite Flash Player's connection to Akamai Streaming and to help surmount problematic client-side proxies. Chapter 6 discusses both of these solutions.

HTTP headers for ident requests must not be larger than four (4) kilobytes in size


Your client Flash applications can direct Flash Player to send ident requests to Akamai Streaming as a means of circumventing problematic proxy servers (see “Dealing with Problematic Client-Side Proxy Servers” on page 67). Some firewalls and proxy servers, however, can introduce large cookies into these request's HTTP headers, adding considerable bulk. To accommodate these, Akamai Streaming supports HTTP headers up to four (4) kilobytes in size. For the sake of efficiency, headers larger than this are denied.

Akamai Streaming only supports streaming media

Adobe's Flash Media Server can enable many different types of communication applications. Akamai Streaming, however, only supports streaming video.

Akamai Streaming (for Live Adobe Flash Video) supports several video and audio streaming media formats

Akamai Streaming supports Flash video and MP3 audio, as well as MPEG-4 standard H.264 video and HE-AAC (High Efficiency Advanced Audio Coding).

 *Note: Support of the H.264 video and HE-AAC audio codecs was introduced with Adobe Flash Player 9 Update 3 (version 9.0.115.0).*

Be aware of potential high-bit-rate issues for end users when encoding video with the H.264 video codec

The H.264 video codec is designed for high-quality, high-bit-rate video, and many encoders, by default, create videos with bit rates that may be difficult for end users' bandwidth capacities to handle. You should, therefore, use care when encoding your videos to ensure you are not overpowering your audience's capabilities.

Akamai Streaming automatically closes idle connections

By default, if an end-user connection remains idle (i.e., no data has streamed) for 60 minutes, Akamai Streaming automatically closes the connection. Instances in which this might occur include end users pausing playback on their client applications, or the stream terminating for some reason (e.g., end of stream).

If the 60-minute default is impractical for your situation (e.g., you are serving advertisement streams), your Akamai representative can adjust it to your specification.

The AkamaiConnection class's "Fast Start" feature does not work with live Flash streaming

The AkamaiConnection class's "Fast Start" feature is a dual-buffer feature for on-demand Flash streaming. It works by buffering a short period of video to begin playback as quickly as possible. As playback begins it continues buffering video for a longer period. This is possible because the video is pre-recorded and can be retrieved at will by the class.

With live Flash streams, however, once the buffer begins playback, it is refilled in real time making additional buffering impossible. Doing so would cause the video to freeze.

Akamai Streaming supports only Akamai-furnished server-side scripts and applications

Akamai does not support customer-authored scripts and applications for Flash Media Server.

Akamai Streaming does not support custom server configurations

This includes such things as log formats and server-side protocol/port rollover instructions.

Strive for a "perfect" first-mile link

Flash video streams' tolerance for first-mile issues is low. If the upload bandwidth is insufficient, for example, end users may see stuttering and/or an additional latency introduced in the stream.


To help address this, be certain to have a tested upload throughput for both primary and backup streams of at least two times greater than their bit rates, and reduce the bit rate if you encounter too many problems. For example, if your stream's bit rate is 500 kpbs, you will need 1 Mbps of throughput each for primary and backup, or 2 Mbps total.

In addition, a "perfect" first mile link will have no intervening firewall. This is important because some routers and firewalls force the use of a small MTU (Maximum Transmission Unit), which adversely affects Flash streaming quality. Worse still, some

routers may not communicate this properly if ICMP (ping) traffic is blocked. (Refer to <http://support.microsoft.com/kb/314825> for a full description of the issue.)

Use the backup Entry Point

Using both primary and backup Entry Points for your stream is important for redundancy. Should your encoder lose connectivity with the primary Entry Point, Akamai Streaming can switch to the backup stream to ensure a continuous stream to your end users. For example, Akamai performs required maintenance periodically on its Entry Point servers, and if your primary encoder is scheduled for this during the course of your streaming event, having the backup set up ensures uninterrupted service.

 *Note: Be aware, the failover from primary to backup streams occurs at the server level, not at the client level as with other streaming formats. There is no “backup URL” for live Flash streaming.*

If you are streaming media encoded with the H.264 or HE-AAC codecs, force end users to upgrade to a supported version of Adobe Flash Player

It is important to avoid streaming media encoded with either the H.264 video or HE-AAC audio codecs to end users having Adobe Flash Player versions older than 9.0.115.0. Due to a bug in the current version of Flash Media Server used on the Akamai Streaming platform, attempts to do so will cause the server to crash, thereby causing a denial of service.

As a precaution, if you are using these codecs you should force end users with older Flash Players to upgrade to the current version. There are methods available for detecting an end user's Flash Player version and producing a particular behavior based on the result, including forcing end users to upgrade their Flash Player or redirecting them to the Adobe Flash Player download Web page if no player is detected at all. These methods are discussed at http://kb.adobe.com/selfservice/viewContent.do?externalId=tn_14526&sliceId=2. The Flash Player Detection Kit is available for assistance in setting this up at http://www.adobe.com/products/flashplayer/download/detection_kit/.

Lastly, a useful online tool is available that tells users which version of Flash Player is currently installed on their computers. (<http://www.adobe.com/products/flash/about/>).

Consider publishing Flash SWF files to accommodate older versions of Flash Player

Streaming Flash video was first enabled in Adobe Flash Player version 6. This version has since been superseded, and to ensure your files are playable on all generations of streaming-video-capable players, you may wish to change your Flash SWF file publishing settings to Flash Player 6. As mentioned in the previous guideline, a major exception to this is if the video you plan to stream in the SWF is encoded with the H.264 video or HE-AAC audio codecs, which are only supported with the release of Adobe Flash Player 9 Update 3 (version 9.0.115.0).

Alternatively, there are methods available for detecting an end user's Flash Player version and producing a particular behavior based on the result, including forcing end users to upgrade their Flash Player or redirecting them to the Adobe Flash Player

download Web page if no player is detected at all. These methods are discussed at http://kb.adobe.com/selfservice/viewContent.do?externalId=tn_14526&sliceId=2. The Flash Player Detection Kit is available for assistance in setting this up at http://www.adobe.com/products/flashplayer/download/detection_kit/.

Lastly, Adobe has a useful online tool that tells users which version of Flash Player is currently installed on their computers. (<http://www.adobe.com/products/flash/about/>).

For live Flash streaming, Akamai Secure Streaming is applied on a per-play basis.

Per-CP-code/per-hostname application of Akamai Secure Streaming is not yet available for live Flash streaming as it is with on-demand Flash streaming. Rather, it is currently applied to individual streams.

For Security, use Adobe's best practice recommendations.

Following are steps you can take to help improve the security of your content (Details are available at:

<http://kb.adobe.com/selfservice/viewContent.do?externalId=kb405456>)

- **Ensure end users are using Flash Player version 10.0.22 or newer.** Use of this Flash Player version or newer with Akamai Streaming adds improved authentication.

Refer to the best practice above entitled “If you are streaming media encoded with the H.264 or HE-AAC codecs, ensure end users are using a supported version of Adobe Flash Player” for information on how to do this.

- **Use the RTMPE protocol and SWF Verification options together.** Using these two features in tandem provides a much more secure solution than using either by itself.

► *Note: Use of RTMPE and SWF Verification are currently considered secure only when used with the combination of Adobe Flash Media Server 3.0.3, which is currently used in the Akamai Streaming servers, and Flash Player 10.0.22. Be aware, known vulnerabilities exist in other server/client version combinations, and the potential always exists for new ones to be introduced in future versions of either.*


- **RTMPE protocol.** RTMPE encrypts data packets to prevent packet sniffing. Its implementation simply requires using the `rtmpe://` or `rtmpte://` protocol in the connection request instead of `rtmp://` or `rtmpt://`.
- **SWF Verification.** This feature authorizes the Flash playback application requesting the stream. Copies of all authorized SWFs are uploaded to a specified NetStorage location and are used to validate stream-requesting SWFs at connect time.

► *Note: SWF Verification is an additional service you must include in your Akamai contract. Its use also requires an Akamai NetStorage account.*

- **Enable RTMPE Enforcement.** RTMPE Enforcement can be enabled by your Akamai representative at your request and is designed to ensure that any connec-

tion attempts using a non-encrypted protocol (RTMP and RTMPT) will be denied. Be advised, you must ensure your streams are using RTMPE and RTMPTE exclusively before enabling this option; otherwise your end users could experience a denial of service (DoS).

- **Use Secure Streaming.** Akamai's Secure Streaming feature uses tokens to add an additional layer of security. This can be used to prevent your Flash playback application from being posted on a third-party site, something that SWF Verification and RTMPE do not directly address.

 *Note: Secure Streaming is an additional service you must purchase separately from Akamai. It also requires you to integrate token generation software on the origin server for your content.*

When using SWF Verification, upload your SWF file to your SWF Path on Akamai NetStorage before making it available to end users to avoid denials of service

If your SWF file is available to your end users and they use it to request your video before you have uploaded the SWF to your SWF Path on NetStorage, the SWF Verification will fail, resulting in denials of service lasting 15 minutes (the Akamai Streaming server cache TTL).

When replacing SWF Verification files on Akamai NetStorage, temporarily maintain both the old and new SWFs to prevent denials of service to end users

If you are using the SWF Verification feature, use care when replacing existing SWF files with updated versions, as a misstep could result in denials of service to your end users. Specifically, it is important to temporarily keep both versions available on NetStorage for SWF Verification until such time as you are reasonably certain all end users are using your new SWF. The following scenario illustrates why this is important:

1. A customer uploads a SWF file, **player.swf**, to their SWF Path on NetStorage, followed by an upload of the same file to their HTTP origin Web site.
2. An end user requests **player.swf** from your Web origin, and it is subsequently delivered to his or her Web browser.
3. The Flash Player client creates a hash of the SWF file that is sent with the video request to an Akamai Streaming server.
4. The Streaming server looks in its cache to determine if it has a matching SWF hash. In this case it does not, so it requests a list of hashes kept in the appropriate SWF Path from NetStorage.
5. A matching hash is found and is cached at the Streaming server for 15 minutes.
6. With a valid hash in its cache, the Streaming server begins streaming the video to the end user.
7. After some time, the customer updates the SWF file, saving it to their NetStorage SWF Path with the same name, thus overwriting the original.
8. Before the updated SWF is uploaded to the HTTP origin Web site, an end user again requests **player.swf**, which is delivered from the origin. This is, however,

the original SWF, not the updated version that is now on NetStorage. At the same time, the existing SWF hash in the Streaming server's cache expires.

9. Once again, the SWF sends a stream request (and the Flash Player-created hash) to the Streaming server.
10. Because the previous SWF hash has expired from the server's cache, the server again requests a list of hashes from NetStorage. This time, however, there is no match.
11. With no matching hash, the Streaming server caches a “no-match” for 15 minutes, prohibiting the video from streaming to any end user using the original SWF and resulting in a denial of service.

To avoid this situation, it is best to upload your updated SWF file to NetStorage using a name different from the original (the files on the SWF Path and on your Web site origin do not need to have the same name, only the same hash) to avoid overwriting and, thus, leaving the original available to any end users who may request your video before you are able to upload the updated SWF to your origin. You do not need to do the same with your origin's file; you can overwrite the original SWF file there with the same name. In fact, this is recommended so you do not need to update your Web page, as well.

Once you feel reasonably sure that enough time has passed to where your end users are no longer using the original SWF, you can delete it from your SWF path.

Do not upload Adobe AIR™ (Adobe Integrated Runtime) files to the SWF Verification path

Adobe AIR files are essentially archive files (similar to .zip files) containing Flash application (SWF) files, which are extracted at their runtime. If you use the SWF Verification feature with Akamai Streaming, the server will attempt to authenticate the requesting SWF by comparing its hash with the hash of your authorized archived version in your SWF Path on Akamai NetStorage.

You should not, therefore, upload your AIR file to your NetStorage SWF Path. Rather, you should extract and upload the SWF files contained within its contents, which you can do by renaming the AIR file with the .zip file extension and opening it with the unzipping application of your choice.

Do not use the native Adobe Flash media components—MediaController, MediaDisplay, MediaPlayback, and FLVPlayback—with live Akamai Streaming

Client-side Flash playback applications prompt Akamai Streaming to begin serving a live stream by sending an `FCSubscribe` call to the live server-side application. As long as the video is available and a subscribed client is connected, the stream continues; when the last subscribed client unsubscribes/disconnects from Akamai Streaming, the stream stops.

The native Adobe Flash media components—MediaController, MediaDisplay, MediaPlayback, and FLVPlayback—do not send this `FCSubscribe` call and so can only view an Akamai stream if another client elsewhere has subscribed. This means that playback applications built with these components have no control over the starting and stopping of the live stream: if no other client has subscribed before them,

the stream will not begin, and if the last subscribed client disconnects before they are finished viewing, the stream will stop.

Akamai, therefore, strongly recommends you not use these components to build your playback applications for use with Akamai Streaming.

Akamai Streaming does not support SSL streaming

If you are serving your Web content and SWFs using Secure Sockets Layer (SSL), be aware that any videos streamed into them from Akamai Streaming will be unencrypted. You can, however, stream your videos using Adobe's enhanced RTMP protocol (RTMPE and RTMPTE), which uses 128-bit encryption.

Use of the RTMPE/RTMPTE protocol is enforceable on a per-configuration basis

If you use the RTMPE and RTMPTE protocols exclusively to stream videos associated with a particular Flash configuration, it is recommended you ask your Akamai representative to enable RTMPE enforcement for that configuration to prevent unauthorized access of your video via regular RTMP. Since enforcement is applied on a per-configuration basis, there are situations in which you may not wish to do this. For example, if your Flash configuration has multiple RTMP streams that you are in the process of migrating to RTMPE, you should disable enforcement until the migration is complete else end users attempting to access unmigrated streams via RTMP will be denied.

Also, be aware that Flash player versions previous to 9.0.115 do not support RTMPE, so when creating your client Flash applications you may need to consider either forcing end users with older versions to upgrade or support them through the RTMP protocol.

Chapter 3. Provisioning Akamai Streaming (for Live Adobe Flash Video)

In This Chapter

Accessing Your Akamai Streaming Account • 15

Creating Configurations • 18

Creating Streams • 22

Editing Configurations • 32

Editing Streams • 33

Modifying Secure Streaming • 35

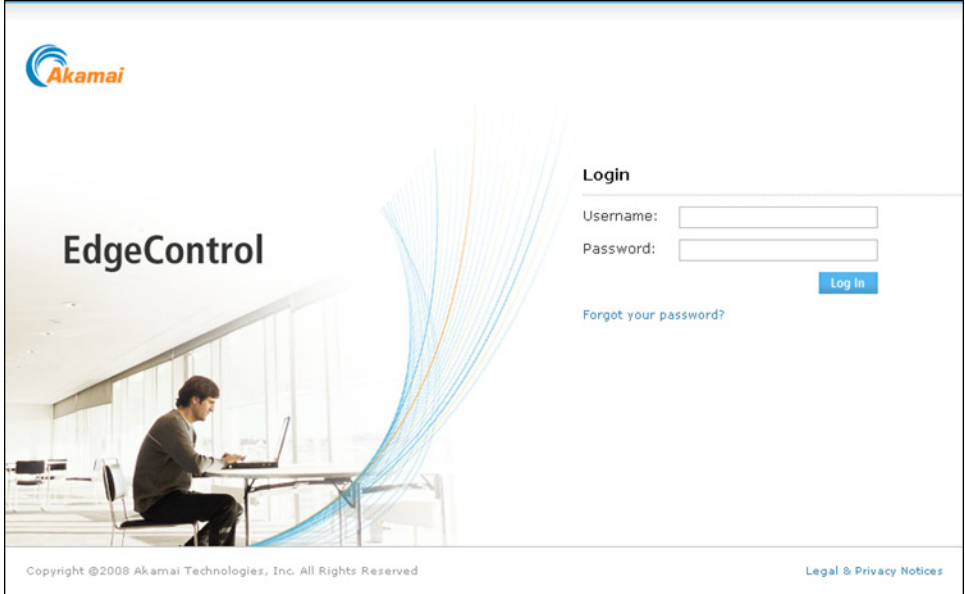
Akamai Streaming (for Live Adobe Flash Video) setup begins with initial activation of your account by Akamai. When completed, you access your Akamai Streaming account through the EdgeControl portal and set all parameters necessary to serve your live Flash video stream. You then configure your video encoder and playback Flash applications as discussed in chapters 4 and 6, respectively.

Accessing Your Akamai Streaming Account

Access your Akamai Streaming account as follows:

1. Log in to the EdgeControl portal.
 - a. Start your Web browser and open <https://control.akamai.com>.

The **Login to Akamai EdgeControl** page appears.



The screenshot displays the Akamai EdgeControl login interface. On the left, the Akamai logo is positioned above the 'EdgeControl' text. The background image shows a person at a desk with a laptop, with abstract blue and orange lines flowing across the scene. On the right side, there is a 'Login' section with two input fields: 'Username:' and 'Password:'. Below these fields is a blue 'Log In' button. A link for 'Forgot your password?' is located below the password field. At the bottom of the page, there is a copyright notice: 'Copyright ©2008 Akamai Technologies, Inc. All Rights Reserved' and a link for 'Legal & Privacy Notices'.

Figure 3-1. The Login to Akamai EdgeControl Page

- b. Enter your username and password, and click [Log In](#).

The EdgeControl Welcome page appears.

2. Navigate to the **Manage Streams** page.

- a. In the left-hand navigation menu, click [Live Streams](#).

The **Live Streams** page appears.

- b. In the left-hand navigation menu under [Live Streams](#), click [Manage Streams](#).

The **Manage Streams** page appears.

Akamai EdgeControl

[Support](#) [Logout](#)

MY SERVICES

- All Services
- HTTP Content Delivery
- HTTP Downloads
- Live Streams
 - Live Streams
 - Traffic
 - Visitors
 - URLs
 - Manage Streams**
 - Tools
 - Log Delivery
 - Alerts
 - Recurring Reports
 - Documentation
- On Demand Streams
- NetStorage
- Site Accelerator
- Web Application Accelerator
- Performance Analytics
- Enhanced DNS

ADMINISTRATION

- Manage CP Codes
- Edit your Profile
- Manage Users

SUPPORT

- Support Home
- Documentation
- Training Resources
- Open/View Support Cases
- Support Contacts
- Feedback

Manage Live Streams [? Help](#)

This page displays a list of your streams configured for live streaming and also allows you to create, edit and delete streams. For background information on the technology and tips on using this tool, click [Live Streams Training](#).

Start/Stop WM Pull Streams | Create New Stream | Stream Creation History | [Configure format...](#)

Search: in [Go](#)

4 Result(s) | [Show Backup Streams](#) Page 1 of 1

Format	Stream Name	Port	CP Code	Encoding Bit Rate	Entry Point	Encoder
<input type="checkbox"/>	a123Stream_Flash@s9 (dynamic)	N/A	9389	N/A	p.ep9.i.akamaientrypoint.net b.ep9.i.akamaientrypoint.net	192.169.0.1 192.169.0.2
<input type="checkbox"/>	a123Stream_qt_128_1	33691	9391	128 kb/s	123.456.789.1	192.168.0.3
<input type="checkbox"/>	a123Stream_real_128_1	21285	9391	128 kb/s	example.g.r.akamaientrypoint.net	192.168.0.4
<input type="checkbox"/>	a123Stream_vms_128_1	39126	9391	128 kb/s	123.456.789.2	http://192.168.0.5

[Delete](#) Page 1 of 1

Show: per Page | Show All

Contact Us | How Can We Serve You Better? | Legal & Privacy | Copyright ©2008 Akamai Technologies, Inc. All Rights Reserved

Figure 3-2. The Manage Streams Page

About the Manage Streams Page



The **Manage Streams** page is the starting point for live Flash Streaming. It lists all your configured streams and includes the following information:

- **Format.** All Streaming formats are available on this page, but Flash streams are identified by the Flash logo.


Additionally, if the stream has Akamai Secure Streaming enabled, a padlock icon () reflecting this state follows the format logo.

- **Stream Name.** The name you chose for the stream, followed by a unique stream ID (e.g., **event@9**), the latter of which is prepended with a letter *s* (e.g., **event@s9**) if Secure Streaming is enabled. This is what you will call your stream when you push it from your Flash video encoder. If you created multiple streams, they will share the same stream name, but are made distinctive by an underscore (_) and a unique integer appended to the end of the stream name (e.g., **event_1@9**, **event_2@10**, **event_3@11**).

Additionally, if the stream has the Dynamic Streaming feature enabled, the stream name is followed by a (**dynamic**) indicator (see “Using Dynamic Streaming” on page 40 for more information).


- **CP Code.** The Content Provider code that will be used for reporting and billing the stream.
- **Entry Point.** The hostnames for the primary () and backup () Akamai Entry Points to which you will push your Flash video encoder’s output. (Use of the backup Entry Point is optional.)
- **Encoder.** The IP addresses of the computers on which your primary and backup Flash video encoder applications are running (or your ping proxy if the encoders are unpingable).

The page provides access to the **Flash Configuration** page from which you will create your live Flash configurations (see “Creating Configurations” below), the first step in provisioning Akamai’s live Flash streaming. In addition, several other functions are available on the **Manage Streams** page:

- **Search for specific streams by stream name or Entry Point port number, and CP code.** Enter either the name of the stream you would like to view or its Entry Point port number in the **Search:** text box, select the stream’s CP code (or “All CP Codes”) from the **CP Code** dropdown menu, and click .


Including the “@” symbol in your stream name search term (e.g., **keynote@3211**, **keynote@**, or **@32**) restricts the search to Flash streams (see “Viewing Stream Details” on page 30 for a description of Flash stream names).


- **Modify how the page’s information is displayed:**
 - Sort the table by a different column header by clicking a column header’s name. For example, clicking **Encoder** sorts the list numerically by encoder IP address.

- Choose the number of rows per page you want to display by selecting either a number (e.g., 10, 20, 30, etc.) or [Show All](#) from the **Show** list.
- **Request a new live Flash stream.** You can set up new live Flash streams by clicking [Create New Stream](#) (see “Creating Streams” on page 22).
- **Delete a stream.** To delete a Flash stream, select its check box and click .
- **Display a stream's details.** You may display additional information by clicking the stream's name (see “Viewing Stream Details” on page 30).

Creating Configurations

Before you can create new Flash streams, you must create at least one Flash configuration to establish parameters for the streams such as the CP code and hostname. You may create as many live Flash configurations as you have CP codes available, depending on how you would like to have Akamai report on and bill your Flash video streams.

 *Note: Each CP code may have both live and on-demand configurations assigned to it. A single CP code may not, however, have two configurations of the same type (e.g., two live configurations).*

1. Navigate to the **Flash Configuration** page.
 - a. Log in to the EdgeControl portal.
The EdgeControl portal home page appears.
 - a. In the left-hand navigation menu, click [Live Streams](#) tab.
The **Live Streams** page appears.
 - b. In the left-hand navigation menu under [Live Streams](#), click [Manage Streams](#).
The **Manage Streams** page appears.
 - c. From the **Other links...** dropdown menu, select **Flash Configuration**.
The **Flash Configuration** page appears.
2. Create the Flash configuration.
 - a. From the **Reporting Code** dropdown menu, select a CP code for which you have not yet created a configuration and click .

The page displays the new, unconfigured CP code

Manage Streams ? Help

[All Streams](#) | **Flash Configuration**

Reporting Code: 9391 - Example.com Switch

In order to create new streams, Reporting code "9391 - Example.com" must be configured for Live Flash streaming. [Configure now.](#)

(1) Encoder IP or optional ping proxy if encoder is not pingable.

Stream Name	Encoder IP(1)	Entry Point
No Available Streams		

Figure 3-3. The Flash Configuration Page with an Unconfigured CP Code

- a. Click the [Configure now](#) link.

The **Add Flash Configuration** page appears.

Add Flash Configuration

[Live Configurations](#) | **Add Configuration**

To add a new Flash configuration, complete the form below and click **Save**.

Configuration Name:

Billing CP Code: 9391 - Example.com

Default Hostname: cp9391.live.edgefcs.net

Default Application Name: live

Host Aliases (optional):

Separate multiple aliases with a new line. To stream content on these hostnames, you must create them on your DNS server as CNAME records pointing to the "default hostname" above.

SWF Verification: ☐ Enable SWF Verification
Please note that SWF Verification feature is only for customers on the Flash Media Server 3 network.

Save Cancel

Figure 3-4. The Add Flash Configuration Page


The page displays some prepopulated configuration information, including **Billing CP code** and **Default Hostname**. Your Flash application developers will use the latter, which is an Akamai-generated hostname consisting of the CP code plus the “live.edgefcs.net” domain (e.g., cp9391.live.edgefcs.net), in their playback applications to access Akamai Streaming.

In addition, the page displays the **Default Application Name**. This is the name—**live**—of the server-side application your Flash developers will use in

their applications (unlike Akamai On Demand Streaming, you may not create aliases for the live server-side application name).


- b. In the **Configuration Name** text box, enter a unique identifier.
- c. In the **Host Aliases** text box, enter any domain names you plan to CNAME to the default hostname, entering each alias on a new line (**optional**).


If you have unique, registered domain names you would like to resolve to the Akamai-generated default hostname, enter them here. Your Flash application developers may use the aliases in their applications in lieu of the default.

 *Note: You must set these aliases up as CNAME registers yourself; Akamai does not provide this service. The information you enter here is required by Akamai so that the appropriate server-side configurations can be made. If you plan to use host aliases and you do not populate this field, your streams will not work.*

- d. Enable SWF Verification, if desired.

To use the SWF Verification feature, you must have an Akamai NetStorage account set up, and you must create your SWF path structure on that account before designating it here. Once complete, you must upload to this location all Flash playback application (SWF) files you intend to use to stream this Flash configuration's videos. Akamai Streaming will use these files to create hashes with which it can verify the authenticity of any SWF files requesting your streams. You may also use this location as the origin for the SWF files on your Web site, but if you choose not to, you must still upload copies of them here.

 *Note: SWF Verification is only available with Flash Player version 9.0.115.0 and higher (use the online tool at <http://www.adobe.com/products/flash/about/> to check your version).*

 *Note: It takes up to 15 minutes for new or replacement file uploads to NetStorage to become available for SWF Verification on the Akamai Streaming network. During this period, attempts at SWF Verification with these files may fail. Likewise, file deletions from NetStorage take up to 15 minutes to take effect.*

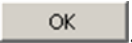
- i. If you wish to enable SWF Verification for all the configuration's videos, click the **Enable SWF Verification** check box.

The **SWF Path** dropdown menu and text box appear.

- ii. From the **SWF Path** dropdown menu, select the Akamai NetStorage domain housing your authorized client-side Flash application SWF files.
- iii. In the **SWF Path** text box, enter any subdirectories (if any) along the NetStorage domain path in which your authorized SWF files are kept.

All your SWF files must reside on this exact path. Unspecified subdirectories are not checked. For example, if you enter a SWF Path of **example.download.akamai.com/swfauth/**, a SWF file in **example.download.akamai.com/swfauth/sports/** will not be checked.

- e. Click .

A dialog box appears asking you to verify the correctness of the configuration information. If all is correct, click . A message appears confirming the configuration's creation. From here, clicking [Live Configurations](#) returns you to the **Flash Configuration** page, which displays the details of your newly-created configuration.



Reporting Code:  [Edit](#) | [Delete](#)

Configuration Name: Product Demonstrations

Username: 9391

Hostname(s): cp9391.live.edgefcs.net (default)

SWF Verification Enabled: Enabled

SWF Path: example.download.akamai.com/9389/swfauth/

Figure 3-5. Live Configuration Details

About Live Configuration Details

Live configuration details include:

- **Reporting Code.** The CP code used by Akamai for reporting and billing of your Flash streams.
- **Configuration Name.** The unique name you gave the configuration.
- **Username.** The user name your Flash video encoder will use to access the Akamai Entry Point. This value is the same as the configuration's CP code.
- **Hostname(s).** The default hostname assigned by Akamai Streaming, as well as any hostname aliases you created to use in lieu of the default.
- **SWF Verification Enabled.** The status of the SWF Verification feature, which helps ensure only authorized client-side Flash applications (SWFs) can stream your video. If enabled, when your SWF file attempts to access a video associated with the Flash configuration, Akamai Streaming compares a hash of the SWF file with the hash of your authorized archived version, and if the two match it commences the stream. If the two do not match, a **NetConnection.Connect.Close** event is sent.
- **SWF Path.** This is only present if SWF Verification is enabled. It is the Akamai NetStorage location to which you will upload your authorized SWF files. Akamai Streaming uses these to verify the validity of the SWF files requesting videos associated with the Flash configuration.

After creating the configuration, you may change it by clicking its [Edit](#) link (see “Editing Configurations” on page 32), delete it by clicking its [Delete](#) link, or create streams for it by returning to the **Manage Streams** page and clicking [Request Stream](#).

Creating Streams

1. Navigate to the **Manage Streams** page.
 - a. Log in to the EdgeControl portal.
The EdgeControl portal home page appears.
 - a. In the left-hand navigation menu, click [Live Streams](#).
The **Live Streams** page appears.
 - b. In the left-hand navigation menu under [Live Streams](#), click [Manage Streams](#).
The **Manage Streams** page appears.

Manage Live Streams ? Help

This page displays a list of your streams configured for live streaming and also allows you to create, edit and delete streams. For background information on the technology and tips on using this tool, click [Live Streams Training](#).

[Start/Stop WM Pull Streams](#) |
 [Create New Stream](#) |
 [Stream Creation History](#) |
 [Configure format...](#)

Search: in

4 Result(s) | [Show Backup Streams](#) Page 1 of 1

Format	Stream Name	Port	CP Code	Encoding		Entry Point	Encoder
				Bit Rate			
<input type="checkbox"/>	a1235stream_Flash@s9 (dynamic)	N/A	9389	N/A		p.ep9.i.akamaiendpoint.net b.ep9.i.akamaiendpoint.net	192.169.0.1 192.169.0.2
<input type="checkbox"/>	a1235stream_qt_128_1	33691	9391	128 kb/s		123.456.789.1	192.168.0.3
<input type="checkbox"/>	a1235stream_real_128_1	21285	9391	128 kb/s		example.g.r.akamaiendpoint.net	192.168.0.4
<input type="checkbox"/>	a1235stream_wms_128_1	39126	9391	128 kb/s		123.456.789.2	http://192.168.0.5

Page 1 of 1

Show: per Page | Show All

Figure 3-6. The Manage Streams Page

2. Complete Step 1.
 - a. Click the [Create New Stream](#) link.
The **Create New Stream: Step 1** page appears.
 - b. From the **Stream Format** dropdown menu, select **Flash**.

The page displays the Flash stream parameters.

Create New Stream: Step 1 [Help](#)

For background information on the technology and tips on using this tool, click [Live Streams Training](#).

[All Streams](#) | **Create New Stream**

Stream Format:

Stream Name: ☐ Allow any name (dynamic)

Enter a name to identify your stream. Valid stream names can only use the following characters "a-z" "A-Z" "0-9" " _ " - ".
If "Allow any name" check box above is checked, the stream name input above is only for reference purpose within EdgeControl and you can use any stream name during the live event.

CP Code:

Per Play Secure Streaming: ☐ Enable Per Play Secure Streaming

Primary Encoder IP / Ping Proxy:

If your encoder is pingable, please enter your encoder IP address. Otherwise, please enter another pingable IP address near your encoder. Akamai will briefly ping this IP address from multiple locations to determine the best available Entry Point server for your stream. Typically your firewall IP could be used here. This will enable us to ping and find the optimal Entry Point server on the Akamai streaming network.

Backup Encoder IP / Ping Proxy:

If your encoder is pingable, please enter your encoder IP address. Otherwise, please enter another pingable IP address near your encoder. Akamai will briefly ping this IP address from multiple locations to determine the best available Entry Point server for your stream. Typically your firewall IP could be used here. This will enable us to ping and find the optimal Entry Point server on the Akamai streaming network.

Number of Streams (Optional): # of bulk streams to create.

Stream End Date (Optional): (mm/dd/yyyy)

Password (Optional): *

* For Live Flash streams, you will need a user account to be able to connect to the Flash servers on the Akamai streaming network and push your live streams. Your "Username" will be your CP Code and the password you can specify above. If left blank, Akamai will automatically generate one for you.

Primary Contact:

These contacts will be notified via email in the event of Entry Point datacenter unavailability.

Secondary Contact:

Figure 3-7. The Create New Stream: Step 1 Page

- c. In the **Stream Name** text box, type a name for your stream.

Your stream name may be up to 90 characters in length and may include the characters “a–z”, “A–Z”, “0–9”, underscores (_), and hyphens (-).

- d. If you wish to use the Dynamic Streaming feature (see “Using Dynamic Streaming” on page 40), click the **Allow any name (dynamic)** check box.

Checking this box means you can use any stream name you like in your Flash encoder. The name you entered in the **Stream Name** text box will be used only for reference within the EdgeControl portal.

► *Note: If this check box is not present, Dynamic Streaming is not enabled for your Flash configuration. Contact your Akamai representative to enable this feature.*

- e. From the **CP Code** dropdown menu, select the CP code you would like to associate with the stream.
- f. Click the **Per Play Secure Streaming** check box if you wish to enable this per-play feature.

► *Note: Secure Streaming helps prevent theft or deep linking of your Flash video links. It is an additional service you must purchase separately from Akamai. Otherwise it is unavailable, and the check box is not present.*

- g. In the **Primary Encoder IP/Ping Proxy** text box, enter the IP address of the primary computer on which your Flash video encoder application is running and click **Test**.

The EdgeControl portal tests your primary IP address by pinging it. This allows Akamai to determine the optimal Akamai Streaming Entry Point for your stream.

► *Note: If your encoder is behind a firewall or other such barrier, you may use the firewall's IP address here. This is called a ping proxy.*

- h. In the **Backup Encoder IP/Ping Proxy** text box, enter the IP address of the backup computer on which your Flash video encoder application is running and click **Test**.

The EdgeControl portal tests your backup IP address by pinging it. If the backup IP address is the same as the primary, you may click

Populate same as Primary to populate the text box with the primary IP address.

- i. In the **Number of Streams (optional)** text box, enter the number of actual streams you would like to associate with this particular stream configuration (up to a maximum of 50).

If you only want one stream, you may leave this blank.

- j. In the **Stream End Date (optional)** text box, enter the date on which you expect your live streaming event to end, if applicable.

Alternatively, you may click **Show Calendar** to display a calendar from which you can select a date.

- k. In the **Password (optional)** text box, type a password for your encoder to use to gain access to the Akamai Entry Points (if you would like Akamai to create a random password for you, leave this entry blank).

Passwords can be made up of any character type, but must be no more than ten (10) characters in length.

- l. From the **Primary Contact** and **Secondary Contact** dropdown menus, select the persons within your organization to contact regarding Akamai Entry Point availability issues (these contacts are required).
- m. Click **Next >**.

The **Create New Stream: Step 2** page appears.

Create New Stream: Step 2

[All Streams](#) | [Create New Stream](#)

Stream Name : event (dynamic)

CP Code : 9391 - Example.com

Primary Encoder IP: 192.168.0.1

Backup Encoder IP: 192.168.0.2

Per Play Secure Streaming : True

Format : Flash

Primary Contact : John Doe

Secondary Contact : Jane Roe

Stream

Stream Name: (dynamic)

Primary Encoder IP / Ping Proxy:

Backup Encoder IP / Ping Proxy:

Stream End Date (Optional): (mm/dd/yyyy)

Password (Optional):

Next > **Cancel**

Figure 3-8. The Create New Stream: Step 2 Page

3. Complete Step 2.
 - a. Verify the information on the page, making any necessary changes, and click **Next >**.

The **Create New Stream: Step 3** page is displayed, the appearance of which depends on whether you enabled Per Play Secure Streaming for the stream.

4. Complete Step 3.

- If Per Play Secure Streaming is not enabled, Step 3 allows you to review and submit the stream request.

Create New Stream: Step 3

[All Streams](#) | [Create New Stream](#)

Please confirm that the information below is correct. This request will be submitted to The Akamai network for provisioning. If you enter your email address in the space provided, you would be notified only if an exception arises during provisioning of this stream.

When you are done reviewing this information, click **Submit Request** to submit this request.

Stream Summary

Stream Name: event (dynamic)
Format: Flash
CP Code : 9391 - Example.com
Per Play Secure Streaming : False
Primary Contact : John Doe
Secondary Contact : Jane Roe

Individual Streams

Stream Name	Primary Encoder IP	Backup Encoder IP	End Data	Password
event (dynamic)	192.168.0.1	192.168.0.2		

Send email notification to:

[Submit Request](#) [Cancel](#)

Figure 3-9. The Create New Stream: Step 3 Page for a Stream Without Secure Streaming

- Confirm the stream parameters, and, if you desire to receive a notification once the the stream has been provisioned, enter your e-mail address in the **Send email notification to** text box, click [Submit Request](#), and proceed to step 2.n. below.

The **Stream Request Submitted** page appears listing the full names of the streams you have requested (see “Viewing Stream Details” below for a description of Flash stream names).

- If Per Play Secure Streaming is enabled, Step 3 lists your stream’s Per Play Secure Streaming profiles and allows you to configure new ones. The page’s table will initially be empty, and you must create at least one profile before proceeding.
 - On the **Create New Stream: Step 4** page, click [Add additional profile](#).

The Add Secure Streaming Profile page appears.

Create New Stream: Step 3

[All Streams](#) | [Per Play Secure Streaming Profiles](#) | **Add Per Play Secure Streaming Profile**

Authentication Profile for CP Code 9391
[Hide Advanced Options](#)

Profile name:

Password:

Advanced Options

Additional advanced options to configure per username.

Use E-type token:

No

Enabling E-type token requires inclusion of a Rijndael key in your token.

IP required in all tokens? :

No

Path required in all tokens?:

Yes

The path value in the token ties the token to a particular stream or piece of content. By removing the path from the token you are reducing the security of the token. By selecting "No" in this field you acknowledge that you are rejecting Akamai's strong recommendation against treating this parameter as "optional" and that you are introducing vulnerabilities to the security of your service by choosing to make this parameter optional. You are hereby on notice that, if you nevertheless choose to lower the security restriction, you bear all of the associated risks if any vulnerability occurs and release Akamai from any responsibility for any losses of any sort arising from taking this action.

Payload required in all tokens?:

No

Setting this value to Yes will reject any token that does not include a payload field

CIDR Block Restriction:

Use this option if you would like to restrict viewing of this stream to end users in a certain IP range or CIDR block. Entries are as follows: IP range restriction: a.b.c.d-w.x.y.z
CIDR block restriction: a.b.c.d/e
Single IP restriction: a.b.c.d-a.b.c.d

WARNING: The authentication profile added here applies to all the streams provisioned in this session.

Add

Cancel

Figure 3-10. The Add Per Play Secure Streaming Profile Page

- ii. In the **Profile name** text box, enter a name for the profile.
- iii. In the **Password** text box, type a password to use with the profile.

- iv. If you want to add advanced options to the profile, click [Show Advanced Options](#) to display these.
 - aa. From the **Use E-type token** dropdown menu, select whether you want to use an E-type token. E-type token generation requires a special binary file, which you can download from the **Per Play Secure Streaming Profiles** page after your stream request has been provisioned (see “Retrieving the E-Type Token Binary” on page 31).
 - bb. From the **IP required in all tokens?** dropdown menu, select whether you want the end user’s IP address included in the token. This is used in conjunction with the **CIDR block restriction** text box in which you enter the range of IP addresses and/or CIDR block or blocks to which you wish to restrict content viewing. The restriction is entered as in the following examples:
 - IP range restriction: **a.b.c.d-w.x.y.z**
 - CIDR block restriction: **a.b.c.d/e**
 - Single IP restriction: **a.b.c.d-a.b.c.d**
 - cc. Select whether you would like to required the content path be included in the token.



***CAUTION:** Akamai enables the path requirement by default, which ties the token to a specific stream or piece of content. Be aware, if you disable it you reduce the token’s security, and you acknowledge that you are rejecting Akamai’s strong recommendation against doing so, that you are introducing vulnerabilities to the security of your Secure Streaming service, and that you bear all associated risks should any vulnerability occur. Accordingly, you release Akamai from any responsibility for any losses of any sort arising from taking this action.*

- dd. Select whether you would like to require a payload field be included in the token.
- v. Click [Add](#).

The **Create New Stream: Step 3** page reappears, listing the new profile. If you chose to use an E-type token in the profile, you can download it now by clicking [Download E Token](#).
- vi. To create another profile, click [Add additional profile](#), or click [Next >](#) to continue creating your stream.

The Create New Stream.: Step 4 page appears.

Create New Stream: Step 4

[All Streams](#) | [Create New Stream](#)

Please confirm that the information below is correct. This request will be submitted to The Akamai network for provisioning. If you enter your email address in the space provided, you would be notified only if an exception arises during provisioning of this stream.

When you are done reviewing this information, click **Submit Request** to submit this request.

Stream Summary

Stream Name: event (dynamic)
Format: Flash
CP Code : 9391- Example.com
Per Play Secure Streaming : True
Primary Contact : John Doe
Secondary Contact : Jane Roe

Individual Streams

Stream Name	Primary Encoder IP	Backup Encoder IP	End Data	Password
event (dynamic)	192.169.0.1	192.169.0.2		

Authorization profiles

Username	Password
Example	abcdefgh

Send email notification to:

Figure 3-11. The Create New Stream: Step 4 Page

5. Complete Step 4 (Secure Streaming-enabled streams only).
 - a. Confirm the stream parameters, and, if you desire to receive a notification once the the stream has been provisioned, enter your e-mail address in the **Send email notification to** text box and click .

The **Stream Request Submitted** page appears listing the full names of the streams you have requested (see “Viewing Stream Details” below for a description of Flash stream names).

 - b. Click [All Streams](#) to return to the **Manage Streams** page or [Request New Stream](#) to create another live stream.

Viewing Stream Details

After creating a stream, you can view many of its parameters on the **Manage Streams** page. You can also view those and additional parameters by clicking the stream's name on that same page, which displays the **Stream Details** page.

Stream Details

[All Streams](#) | [Stream Details](#) | [Edit Stream](#)

Reporting Code: 9391 - Example.com

Stream Name: event@s9 (dynamic)

Per Play Secure Streaming: Enabled [Configure Profiles](#)

Encoder IP / Ping Proxy: 192.168.0.1

Stream End Date (optional): Unspecified

Username: 9391

Password: x6je2V

Primary Contact: John Doe

Secondary Contact: Jane Roe

Primary Encoder IP / Ping Proxy: 192.169.0.1

Backup Encoder IP / Ping Proxy: 192.169.0.2

☒ Primary Entrypoint: rtmp://p.ep9.i.akamaiendpoint.net/EntryPoint

☒ Backup Entrypoint: rtmp://b.ep9.i.akamaiendpoint.net/EntryPoint

ARL(s):

CONNECT rtmp://cp9391.live.edgefcs.net/live/

PLAY event@s9?auth=[stream_token]&aifp=[stream_fp]

[Back](#)

Figure 3-12. The Stream Details Page

In addition to the information presented on the **Manage Streams** page, the **Stream Details** page provides:

- **Per Play Secure Streaming.** Indicates whether the stream has Akamai Per Play Secure Streaming enabled. If it does, a [Configure Profiles](#) link appears that you can click to access the **Per Play Secure Streaming Profiles** page where you can download your E-type token binary, if enabled, and create, edit, and delete Per Play Secure Streaming profiles.
- **Stream End Date.** The date on which you expect your live streaming event to end.
- **Username.** The username your Flash video encoder will use to gain access to your Entry Points.
- **Password.** The password your Flash video encoder will use to gain access to your Entry Points (if you left the **Password** text box blank when creating the stream, this is populated by an Akamai-generated password).

- **Primary Contact** and **Secondary Contact**. The persons within your organization to contact regarding Akamai Entry Point availability issues.
- **ARL(s)**. The values to use in your client-side Flash application's **connect()** and **play()** methods for stream viewing by your end users.

Retrieving the E-Type Token Binary

If your stream has Per Play Secure Streaming enabled and you configured your profile to use an E-type token, you must download the binary to use with your token generator. This is done on the **Per Play Secure Streaming Profiles** page.

1. Open the **Per Play Secure Streaming Profiles** page.
 - a. On the **Manage Streams** page, click the name of the stream for which you would like to retrieve an E-type token binary.
The **Stream Details** page appears.
 - b. Click [Configure Profiles](#).

The **Per Play Secure Streaming Profiles** page appears.

Stream Details

[All Streams](#) | [Stream Detail](#) | [Per Play Secure Streaming Profiles](#)

Per Play Secure Streaming Profile for event@s9 [Add additional profile](#)

Profile Name	Token Type	IP Check
Example	D and E Token (Download E Token)	No Delete Modify

Please make sure to commit changes by clicking the "Commit Changes" button after adding/modifying or deleting secure streaming profile.

Note: You must modify the AIFP (Authentication Information Finger Print) parameter when you update any secure streaming profiles. Please refer to the streaming documentation to learn more about this.

[Commit Changes](#)

Figure 3-13. The Secure Streaming Profiles Page

2. Download the token binary
 - a. Click ([Download E Token](#)).

A dialog box appears prompting you to save the file to your computer. Click **OK** to proceed.

Editing Configurations

Should you need to, you may make changes to your Flash configurations' names and aliases.

1. Open the **Edit Configuration** page.
 - a. On the **Manage Streams** page, select **Flash Configuration** from the **Other links...** dropdown menu.

The **Flash Configuration** page appears.

- b. On the **Flash Configuration** page, select the CP code to which you would like to make a configuration change from the **Reporting Code** dropdown menu and click **Switch**.

The selected CP code's configuration parameters and associated streams appear.

- c. Click **Edit**.

The **Edit Configuration** page appears.

Flash Streaming

[Live Configurations](#) | **Edit Configuration**

To edit the current Flash configuration, complete the form below and click **Save**.

Configuration Name:

Billing CP Code: 9391 - Example.com

Default Hostname: cp9391.live.edgefcs.net

Default Application Name: live

Host Aliases (optional):

Separate multiple aliases with a new line. To stream content on these hostnames, you must create them on your DNS server as CNAME records pointing to the "default hostname" above.

SWF Verification: ☒ Enable SWF Verification

SWF Path:

Please note that SWF Path is required for enabling SWF Verification feature and it needs to be a NetStorage location.

Example : To use "storage.download.akamai.com/1234/swfauth" as SWF Path, choose "storage.download.akamai.com/1234/" in the drop down list and enter "swfauth" in the text box.

Note: Subdirectories entered here must be first created at your origin site.

Save **Cancel**


Figure 3-14. The Edit Configuration Page

2. Change the configuration name, if desired.
 - a. Type a new identifier in the **Configuration Name** text box.
3. Add or delete host aliases, if desired.
 - a. In the **Host Aliases (optional)** text box, type a new alias or aliases (entering each on a new line) or delete unwanted aliases.

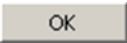
 *Note: Do not forget to create CNAME registers for any new aliases.*

4. Enable or disable SWF Verification, as desired.

If enabling SWF Verification, enter the Akamai NetStorage location to which you will upload your authorized SWF files. Akamai Streaming uses these to verify the validity of the SWF files requesting videos associated with the Flash configuration.

 *Note: SWF Verification is only available with Flash Player version 9.0.115.0 and higher (use the online tool at <http://www.adobe.com/products/flash/about/> to check your version).*

5. Click .

A dialog box appears asking you to verify the correctness of the configuration information. If all is correct, click . A message appears confirming the configuration's modification. Clicking [Live Configurations](#) returns you to the **Flash Configuration** page displaying your new parameters.

Editing Streams

If desired, you may make changes to your streams' parameters (excluding CP code).

1. Open the **Edit Stream** page.
 - a. On the **Manage Streams** page, click the name of the stream to which you would like to make changes.
The **Stream Details** page appears.
 - b. Click [Edit Stream](#).

The **Edit Stream** page appears.

Flash Streaming

[All Streams](#) | **Edit Stream**

Edit your stream details below and click **Save** to update your changes.

CP Code: **9391 - Example.com**

Stream Name: ☒ Allow any name (dynamic)

Enter a name to identify your stream. Valid stream names can only use the following characters "a-z" "A-Z" "0-9" "_" "-".

If "Allow any name" check box above is checked, the stream name input above is only for reference purpose within EdgeControl and you can use any stream name during the live event.

Per Play Secure Streaming: ☒ Enable Per Play Secure Streaming

Primary Encoder IP / Ping Proxy:

If your encoder is pingable, please enter your encoder IP address. Otherwise, please enter another pingable IP address near your encoder. Akamai will briefly ping this IP address from multiple locations to determine the best available Entry Point server for your stream. Typically your firewall IP could be used here. This will enable us to ping and find the optimal Entry Point server on the Akamai streaming network.

Backup Encoder IP / Ping Proxy:

If your encoder is pingable, please enter your encoder IP address. Otherwise, please enter another pingable IP address near your encoder. Akamai will briefly ping this IP address from multiple locations to determine the best available Entry Point server for your stream. Typically your firewall IP could be used here. This will enable us to ping and find the optimal Entry Point server on the Akamai streaming network.

Stream End Date (Optional):

Password: *

* For Live Flash streams, you will need a user account to be able to connect to the Flash servers on the Akamai streaming network and push your live streams. Your "Username" will be your CP Code (173031) and the password you can specify above.

Primary Contact:


These contacts will be notified via email in the event of Entry Point datacenter unavailability.

Secondary Contact:


Figure 3-15. The Edit Stream Page


2. Make any necessary changes.
 - a. Type a new identifier in the **Stream Name** text box, if desired.

- b. Click the **Allow any name (dynamic)** check box to enable or disable the Dynamic Streaming feature, if desired.

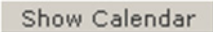
 *Note: If this check box is not present, Dynamic Streaming is not enabled for your Flash configuration. Contact your Akamai representative to enable this feature.*

- c. Enable or disable **Per Play Secure Streaming**, as desired.
- d. In the **Encoder IP / Ping Proxy** text box, enter a new IP address, if desired.

 *Note: If your encoder is behind a firewall or other such barrier, you may use the firewall's IP address here. This is called a ping proxy.*

- e. Click the  buttons following the **Primary** and **Backup Encoder IP / Ping Proxy** text boxes.

Before proceeding, you should successfully test the encoder IP address entries regardless of whether you changed them or not. This allows Akamai to determine the optimal Akamai Streaming Entry Point for your stream.

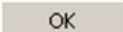
- f. In the **Stream End Date (Optional)** text box, enter the date on which you expect your live streaming event to end, if desired (alternatively, you may click  to display a calendar from which you can select a date).

- g. In the **Password** text box, type a new password for your encoder to use to gain access to the Akamai Entry Points, if desired

Passwords can be made up of any character type, but must be no more than 10 characters in length.

- h. From the **Primary Contact** and **Secondary Contact** dropdown menus, select new persons within your organization to contact regarding Akamai Entry Point availability issues, if desired (contacts are required).

- i. Click .

A dialog box appears asking you to verify the correctness of the configuration information. If all is correct, click . A message appears confirming the stream's modification. Clicking [All Streams](#) returns you to the **Manage Streams** page.

Modifying Secure Streaming

You can also add, delete, and edit your Secure Streaming profiles, as necessary.

1. Access the stream's **Flash Stream Secure Profiles Configuration** page.
 - a. On the **Manage Streams** page, click the name of the stream for which you would like to make Secure Streaming profile changes.

The **Stream Details** page appears.

- b. Click [Configure Profiles](#).

The **Per Play Secure Streaming Profiles** page appears.

Stream Details

[All Streams](#) | [Stream Detail](#) | **Per Play Secure Streaming Profiles**

Per Play Secure Streaming Profile for event@s9
[Add additional profile](#)

Profile Name	Token Type	IP Check	
Example	D and E Token (Download E Token)	No	Delete Modify

Please make sure to commit changes by clicking the "Commit Changes" button after adding/modifying or deleting secure streaming profile.

Note: You must modify the AIFP (Authentication Information Finger Print) parameter when you update any secure streaming profiles. Please refer to the streaming documentation to learn more about this.

Commit Changes

Figure 3-16. The Per Play Secure Streaming Profiles Page

2. Add, delete, or modify your profile(s), as desired.

- Add a profile.
 - i. Click [Add additional profile](#).

The **Add Per Play Secure Streaming Profile** page is displayed, which is identical in appearance and function as the page presented in Step 3 of the stream creation process. See page 26 for procedures on its use.

- Delete a profile.
 - i. Click [Delete](#) next to the profile you wish to remove.

The profile is deleted.
- Modify a profile.
 - i. Click [Modify](#) next to the profile you wish to edit.

The **Modify Secure Streaming Profile** page appears.

Stream Details

[All Streams](#) | [Stream Detail](#) | [Per Play Secure Streaming Profiles](#) | **Modify Per Play Secure Streaming Profile**

Profile name : Example

Use E-type token: Create a new E Token

Client IP Check: No

Path required in all tokens?: Yes

The path value in the token ties the token to a particular stream or piece of content. By removing the path from the token you are reducing the security of the token. By selecting "No" in this field you acknowledge that you are rejecting Akamai's strong recommendation against treating this parameter as "optional" and that you are introducing vulnerabilities to the security of your service by choosing to make this parameter optional. You are hereby on notice that, if you nevertheless choose to lower the security restriction, you bear all of the associated risks if any vulnerability occurs and release Akamai from any responsibility for any losses of any sort arising from taking this action.

Payload required in all tokens?: No

Setting this value to Yes will reject any token that does not include a payload field

CIDR Block Restriction:

Use this option if you would like to restrict viewing of this stream to end users in a certain IP range or CIDR block. Entries are as follows: IP range restriction: a.b.c.d-w.x.y.z
CIDR block restriction: a.b.c.d/e
Single IP restriction: a.b.c.d-a.b.c.d

Update Secure Streaming Profile Cancel

Figure 3-17. The Modify Per Play Secure Streaming Profile Page

ii. From the **Use E-type Token** dropdown menu, select an action.

- **Create a new E Token.** Enables the E-type token and generates a new token binary for use with your token generator (if the token was already enabled, a new binary is generated).

You must download this new token from the **Per Play Secure Streaming Profiles** page by clicking the profile's [\(Download E Token\)](#) link.

- **Keep the same token** (only present if the E-type token was enabled already). Make no changes to the E-type token (remain enabled with the same token binary).
- **Disable E Token.** Disable use of the E-type token.

- iii. From the **Client IP Check** dropdown menu, select whether you would like to require the end user's IP address be included in the token.
- iv. From the **Path required in all tokens?** dropdown menu, select whether you would like to require the content path be included in the token.



CAUTION: Akamai enables the path requirement by default, which ties the token to a specific stream or piece of content. Be aware, if you disable it you reduce the token's security, and you acknowledge that you are rejecting Akamai's strong recommendation against doing so, that you are introducing vulnerabilities to the security of your Secure Streaming service, and that you bear all associated risks should any vulnerability occur. Accordingly, you release Akamai from any responsibility for any losses of any sort arising from taking this action.

- v. From the **Payload required in all tokens?** dropdown menu, select whether you would like to require a payload field be included in the token.
- vi. If you wish to restrict content access to end users within a particular range of IP addresses, type the parameters in the **CIDR Block Restriction** text box as follows:
 - IP range restriction: **a.b.c.d-w.x.y.z**
 - CIDR block restriction: **a.b.c.d/e**
 - Single IP restriction: **a.b.c.d-a.b.c.d**

3. Submit the profile changes.

- a. Click **Update Secure Streaming Profile**.

The **Per Play Secure Streaming Profiles** page reappears.

4. Download the new E-type token binary, if applicable.

- a. On the **Per Play Secure Streaming Profiles** page, click the profile's [\(Download E Token\)](#) link and save the binary to your desired location.

This step can be performed at a later time, if desired.

5. Commit the profile modifications.

- a. Click **Commit Secure Streaming Profile**.

The profile is committed and ready for use.

The next chapter provides information and examples your Flash developers will find useful in integrating the service into their Flash applications.

Chapter 4. Setting Up the Video Encoder Application

In This Chapter



Using Dynamic Streaming • 40

Using Adobe Flash Media Live Encoder and On2 Flix Live • 4

Using the AkamaiFCSPublish Class • 47

Using the Akamai Broadcaster Encoding Tool • 51

On completing your live Flash configuration, you will have the information you need to set up your Flash video encoder to capture and push your stream to the Akamai Entry Points.

Two third-party live Flash video encoders fully support Akamai Streaming:

- Adobe Systems Incorporated's **Adobe Flash Media Live Encoder—Supports Dynamic Streaming.**
- On2 Technologies, Inc.'s **On2® Flix® Live.**

Five other companies have worked closely with Akamai to make their live video encoders compatible with Akamai Streaming (for Live Adobe Flash Video). If you encounter difficulties with these, please contact them directly:

- Anystream, Incorporated (www.anystream.com).
- Digital Rapids Corporation (www.digital-rapids.com).
- Inlet Technologies (www.inlethd.com)—**Supports Dynamic Streaming.**
- Kula Media Group, Inc. (www.kulabyte.com).
- ViewCast Corporation (www.viewcast.com).

In addition, Akamai offers two encoder solutions:

- You may use the **AkamaiFCSPublish** class to build your own encoder.
- You may use the prebuilt **Akamai Broadcaster** encoding tool.



Note: Be aware, the AkamaiFCS Publish class and the Akamai Broadcaster are built around the Adobe Flash Player and its embedded codec, which encodes video at a lower quality than the On2 VP6® codec available in other third-party encoders.

Using Dynamic Streaming

Dynamic Streaming enables seamless, on-the-fly switching—based on the end user's bandwidth (quality of service)—between multiple versions of a live video stream encoded in different bit rates. It is enabled at the Flash configuration level in the EdgeControl portal (contact your Akamai representative to enable this feature) and applied to the streams of your choosing that are associated with that configuration.


This feature is only supported as of the release of Adobe Flash Player 10, so to use it you must either require your end users to upgrade their Flash Players or provide an alternate experience for those with older players. While taking no action will not interfere with the video stream, it may result in an undesirable experience, as the video will not make the aforementioned quality of service adjustments. For example, if an end user is initially playing a high-bit-rate version of the video and his or her available bandwidth subsequently decreases, an inordinate amount of rebuffering might occur.

The application of Dynamic Streaming is based on interactions between your Flash playback application and the Akamai Streaming server. As the video plays, your application continuously monitors quality of service, including the actual bandwidth (based on incoming data and the buffer state) and the end user's device rendering capability (by monitoring the number of dropped frames). As conditions change, ActionScript in your playback application notifies the Akamai Streaming server when it becomes necessary to shift the bit rate up or down. The server then, on the same connection, seamlessly changes to the appropriate bit rate video at the next keyframe. This feature is advantageous in that it delivers a better end-user experience by accommodating wide variations in end-user bandwidth and device capabilities, and by gracefully handle dynamically changing conditions (e.g., in a wi-fi environment in which bandwidth can change mid-stream).

Encoding Your Videos for Dynamic Streaming

Dynamic Streaming supports the ON2 VP6, H.264, HE-AAC, and MP3 codecs, allowing the full range of Flash-compatible video and audio options. To use the feature, encode multiple streams of your video using different bit rates.

When a mid-stream switch occurs between different versions of a video, it takes place at keyframe locations. To avoid video and/or audio artifacts during a switch each version of your video should be encoded with a constant (versus variable) keyframe interval (two (2) seconds is recommended), as well as an identical timeframe, which will help to avoid jumps in the video during a switch. Also, audio for the videos should be encoded at the same bit rate and sample rate.

 *Note: For lower bit rate streams, encoding the audio in mono at the same bit rate (i.e., half the stereo bit rate) is acceptable.*


The maximum number of streams allowed for a single stream ID is 30. These can be any combination of angles and bitrates (e.g., 5 angles with 6 bit rates each or 6 angles with 5 bit rates each).

Using Adobe Flash Media Live Encoder and On2 Flix Live

Adobe Flash Media Live Encoder and On2 Flix Live are Flash video encoder applications available from Adobe Systems Incorporated and On2 Technologies, Inc., respectively.

Using High-Bit-Rate Streams Over High-Latency Connections

If you plan to use On2 Flix Live to broadcast high-bit-rate streams greater than 300 kbps over high-latency connections (greater than 50 milliseconds), be aware this encoder currently has burstiness and other quality issues at these levels.

 *Note: Be aware, Adobe Flash Media Encoder, version 1.x also experiences these issues, but subsequent versions do not.*

Making the following change to your encoding machine's Microsoft® Windows® registry will overcome these issues.

1. Determine if you have the **DefaultSendWindow** key defined.
 - Using the Microsoft Registry Editor (regedit) interactive tool.
 - i. In the left-hand pane, navigate to **HKEY_LOCAL_MACHINE >> SYSTEM >> CurrentControlSet >> Services >> AFD >> Parameters**.
The **Parameters** subdirectory's contents appear in the right-hand pane. If defined, the **DefaultSendWindow** key will be present. If it is not, proceed to step 2.
 - Using a command line.
 - i. From a Windows command prompt, run:

```
reg query "HKLM\SYSTEM\CurrentControlSet\Services\AFD\Parameters"
```

If **DefaultSendWindow** is defined, it should appear in the output. If it does not, proceed to step 2.

If you do not have **DefaultSendWindow** defined, the Windows operating system will default to an 8-kilobyte socket send buffer, which is insufficient for high-latency, high-bandwidth connections.

2. Define the **DefaultSendWindow** key with a DWORD value of 256 kilobytes, if it is not already.
 - Using the Registry Editor.
 - i. With the **Parameters** directory selected in the left-hand pane, select **Edit >> New >> DWORD Value**.
A new **REG_DWORD** key appears in the right-hand pane.
 - ii. Type **DefaultSendWindow** as the key name and press <Enter>.

- iii. Right click the key's name and select **Modify** from the pop-up menu.

The **Edit DWORD Value** dialog box appears.

- iv. In the **Base** area, select the **D**ecimal radio button.
- v. In the **Value data** text box, type **262144** and click **OK**.

The value is set in the right-hand pane.

- Using a command line.
 - i. From a Windows command prompt, run:

```
reg add "HKLM\SYSTEM\CurrentControlSet\Services\AFD\Parameters" /v
DefaultSendWindow /t REG_DWORD /d 262144 /f
```

The output should state that “The operation completed successfully”.

- ii. Repeat step 1 to verify **DefaultSendWindow** is set properly.

The output should display:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AFD\Parameters
DefaultSendWindow REG_DWORD 0x40000
```

- 3. Reboot the encoding machine.

Using Adobe Flash Media Live Encoder

This section limits itself to describing how to use Adobe Flash Media Live Encoder specifically with Akamai Streaming. You should refer to the encoder’s “Help” documentation (**Help >> Flash Media Live Encoder Help**) for additional information on its use. While Akamai Streaming supports earlier versions of the encoder, it is recommended you use version 2.0.1 or later.

After starting the encoder, use the procedures below to set it up to encode and broadcast your Flash video to the Akamai Entry Point.

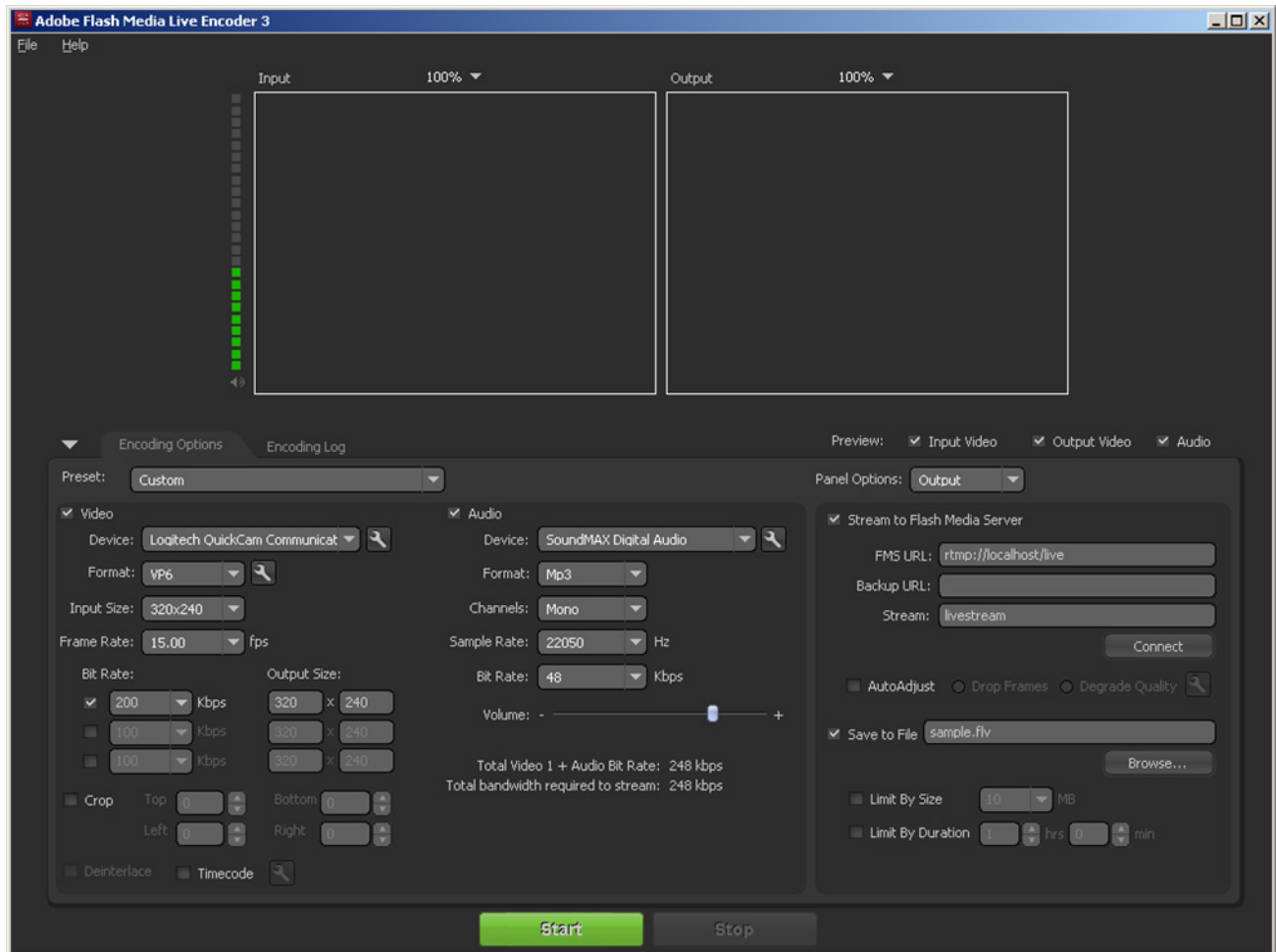



Figure 4-1. The Adobe Flash Media Live Encoder 3

1. Ensure that audio and video input devices are connected and working correctly.
2. In the left-hand panel, select your **Video** and **Audio** settings.

 *Note: If you are using the Dynamic Streaming feature, you may configure up to three individual streams by selecting two or three of the **Bit Rate** check boxes, selecting a bit rate (in kbps) for each from the associated dropdown menu and entering the **Output Size** of each in the associated text boxes.*

3. Connect to the Akamai Entry Points.
 - a. In the right-hand panel, select the **Stream to Flash Media Server** check box.

- b. In the **FMS URL** text box, enter your primary Entry Point's hostname, followed by the application name, **EntryPoint** (e.g., `rtmp://p.ep9.i.akamaientrypoint.net/EntryPoint`).

Obtain the hostname from the **Manage Streams**, **Stream Details**, or **Flash Configuration** pages in the EdgeControl portal.

► *Note: The "EntryPoint" application name is always required when broadcasting to an Akamai Entry Point. (Its inclusion is automated when using the AkamaiFCSPublish class or the Akamai Broadcaster tool.)*

- c. In the **Backup URL** text box, enter your backup Entry Point's hostname, followed by the application name **EntryPoint** (e.g., `rtmp://b.ep9.i.akamaientrypoint.net/EntryPoint`).

► *Note: While using a single encoder to broadcast to both primary and backup Entry Points is a valid setup, it lacks redundancy; should the encoder fail, both primary and backup broadcasts will cease. To avoid this occurring, consider using two encoders, with one broadcasting to the primary Entry Point and the other to the backup. Do not, however, use both encoders to broadcast to both primary and backup Entry Points, as the two broadcasts will conflict.*


- d. In the **Stream** text box, enter the name of your stream.
 - If you *are not* using Dynamic Streaming—, this is the stream name plus the stream ID (e.g., `event@9`). You can obtain this from the **Manage Streams**, **Stream Details**, or **Flash Configuration** pages in the EdgeControl portal.
 - If you *are* using Dynamic Streaming, this is the stream name plus the stream ID (e.g., `event@9`), but with one or more parameters of your choosing appended to the stream name. Upon beginning your streams, these parameters will be replaced with unique identifiers, creating multiple distinct streams at different bit rates on a single connection. For example, if you plan to stream three live videos with the stream name `event`, and you append it here with `%i` (e.g., `event_%i@9`), each stream's name will be appended with its respective file index (e.g., `event_1@9`, `event_2@9`, and `event_3@9`).

► *Note: Be aware, the stream name is not required, but its inclusion is recommended to help distinguish the streams from each other.*

Following are valid tags you may use:

- `%i`—File index. This is the numeric order of the streams' bit rates as they appear, from top to bottom, in the Flash Media Live Encoder. It is the parameter Adobe® recommends using, as it is always unique.
- `%v`—Video bit rate.
- `%f`—Video output size.

- %a—Audio bit rate.
- %s—Audio sample rate.
- %b—Total bit rate (value replacing %V + value replacing %A).

 *Note: Alternatively, you can manually enter unique stream names delimited with semicolons.*

- e. Click the **Connect** button.

The **Connect to FMS** dialog box appears for the primary Entry Point connection.

- f. In the **Username** text box, enter your stream's user name.

Obtain this from your **Stream Details** or **Flash Configuration** pages in the EdgeControl portal.

- g. In the **Password** text box, enter your stream's password.

Obtain this from your **Stream Details** page in the EdgeControl portal.

- h. Click **OK**.

The **Connect to FMS** dialog box reappears for the backup Entry Point connection.

- i. Enter your **Username** and **Password**, and click **OK**.

If all entries are valid, the encoder connects to the two Entry Points, and the **Connect** button is replaced by a **Disconnect** button.

4. Begin your broadcast.

- a. If you wish to create an archive file of your stream, select the **Save to File** check box, and enter the path and file name in the accompanying text box.

If you are using dynamic streaming, the encoder saves a file for each bit rate you are streaming. You must, therefore, include a valid parameter in the file name to produce differently-named archive files (e.g., sample_%i.f4v).

- b. Click the **Start** button to begin broadcasting.

The broadcast begins, and the encoder's view changes to the **Encoding Log** tab.

5. When your event has finished, click the **Stop** button to terminate the broadcast, and click the **Disconnect** button to drop the Entry Point connections.

Using On2 Flix Live

On2 Flix Live is a Flash video encoder available from On2 Technologies, Inc. This section is limited to describing how to use the encoder specifically with Akamai Streaming. You should refer to the *On2 Flix Live User Guide* included with the encoder for additional information on its use.

After starting the encoder, use the following procedures to set it up to encode and broadcast your Flash video to the Akamai Entry Point.

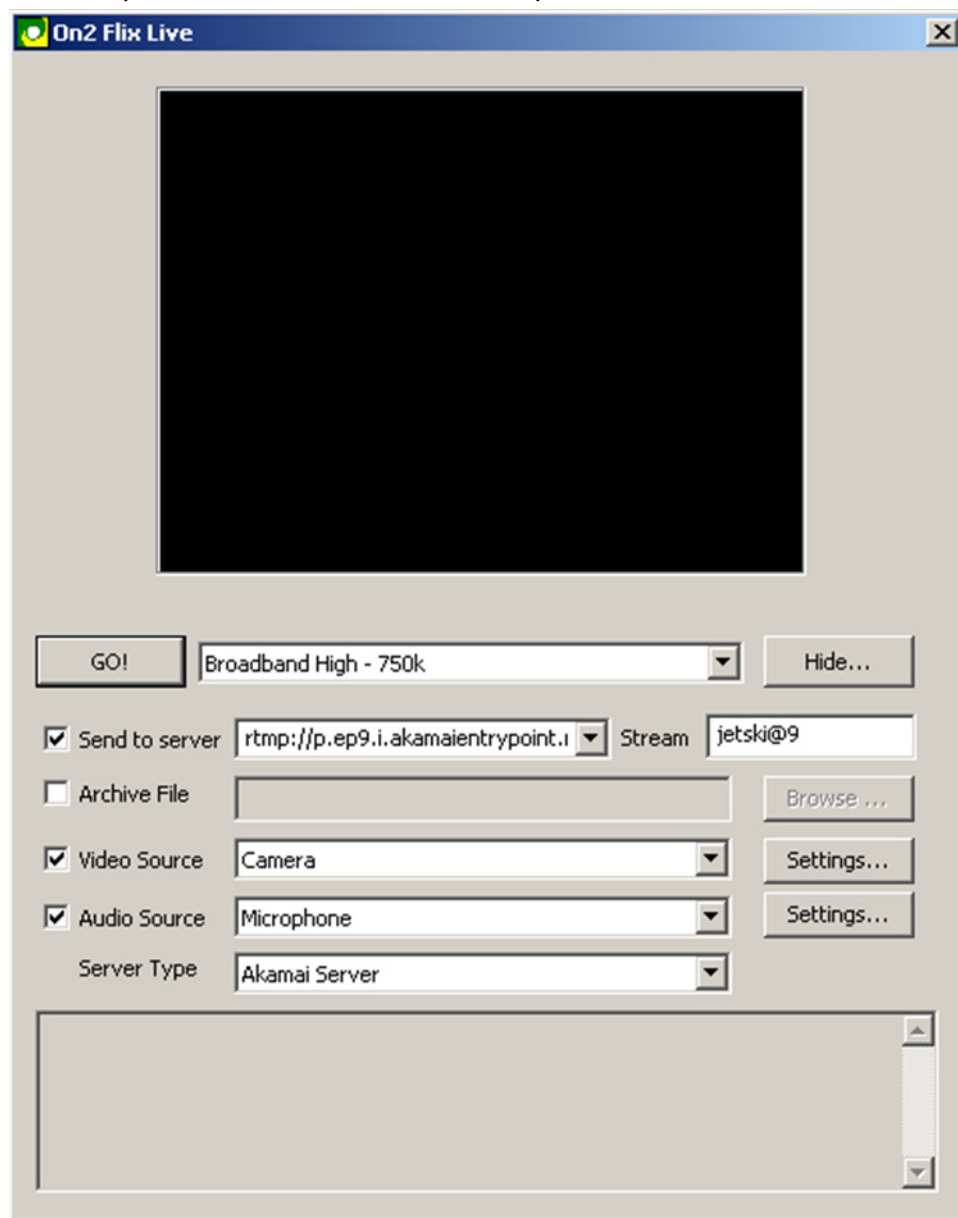



Figure 4-1. The On2 Flix Live 8 Encoder

1. Ensure that audio and video input devices are connected and working correctly.
2. Enter your live streaming parameters.
 - a. Select an encoding preset for your video stream from the dropdown menu and click the **Advanced...** button to display the additional configuration parameters.
 - b. Select the **Send to server** check box, and enter your primary Entry Point's hostname in the accompanying text box, followed by the application name, **EntryPoint** (e.g., **rtmp://p.ep9.i.akamaientrypoint.net/EntryPoint**).

Obtain the hostname from the **Manage Streams**, **Stream Details**, or **Flash Configuration** pages in the EdgeControl portal.

 *Note: The **EntryPoint** application name is always required when broadcasting to an Akamai Entry Point. (Its inclusion is automated when using the AkamaiFCSPublish class or the Akamai Broadcaster tool.)*

If you also desire to stream to the backup Entry Point, you must feed your audio and video sources to a second instance of On2 Flix Live running on a separate computer.

- c. In the **Stream** text box, enter the name of your stream (e.g., **event@9**).

Obtain this from the **Manage Streams**, **Stream Details**, or **Flash Configuration** pages in the EdgeControl portal.

- d. If you wish to create an FLV file of your stream, select the **Archive File** check box, and enter the path and file name in the accompanying text box.
- e. Select the **Video Source** and **Audio Source** check boxes as appropriate, and select those sources from their respective dropdown menus.
- f. From the **Server Type** dropdown menu, select **Akamai Server**.

3. Begin your broadcast.

- a. Click the **GO!** button to begin broadcasting.

The **Publish to Server** dialog box appears.

- b. In the **Username** text box, enter your stream's user name.

Obtain this from your **Stream Details** or **Flash Configuration** pages in the EdgeControl portal.

- c. In the **Password** text box, enter your stream's password.

Obtain this from your **Stream Details** page in the EdgeControl portal.


- d. Click the **OK** button.

If all entries are valid, the broadcast begins.

4. When your event has finished, click the **Stop** button to terminate the broadcast and drop the Entry Point connection.

Using the AkamaiFCSPublish Class

Flash video is unlike other streaming media formats in that it does not require a standardized video encoder application. Your Flash developer can create a Flash encoder application specifically for this purpose that can be tailored with as much flexibility in function as you need.


 *Note: Be aware, the AkamaiFCS Publish class is built around the Adobe Flash Player and its embedded codec, which encodes video at a lower quality than the On2 VP6 codec available in other third-party encoders.*

The ActionScript scripting language is the key to streaming Flash video from Akamai Streaming. Akamai provides an ActionScript class called **AkamaiFCSPublish** that is custom-built for use with Akamai Streaming (for Live Adobe Flash Video). This class is required if you are building your own Flash video encoder, as it contains many key elements that facilitate communications and authentication between the encoder and the Entry Point. You can obtain the class's file from the EdgeControl portal ([Live Streams >> Documentation](#)).

To use the AkamaiFCSPublish class, simply copy the file to the same directory as your FLA file. When you publish the FLA, ActionScript you include in it imports the class and compiles it into the resulting SWF file. If you store your class files in a different location, you can still pull the class into your SWF by specifying a class path in your ActionScript.

Creating a Simple Application for Encoding Live Flash Video

As with other types of Flash applications, your Flash video encoder can be as simple or as complicated as you like. You can build the most basic encoder application with Adobe Flash software simply by inserting an embedded video instance onto a blank stage, naming the instance, and adding an ActionScript.

 *Note: SWF files are cached in the browser. If your browser does not play your latest version, clear its cache and retry the application. Simply refreshing the page will not work.*

1. In Flash CS3 or Flash 8, open a new Flash document.
 - a. From the **File** menu, select **New**.
The **New Document** dialog box appears.
 - b. Select **Flash Document** (Flash 8) or **Flash File (ActionScript 2.0)** (Flash CS3) and click **OK**.
A new Flash document appears.
2. Create a new video instance.
 - a. If the Library pane is not already present, select **Library** from the **Window** menu.
The pane appears.
 - b. From the pulldown menu on the right-hand side of the **Library** pane's title bar select **New Video**.
The **Video Properties** dialog box appears.
 - c. Enter a name in the **Symbol** text box, if desired, select the **Video (ActionScript-controlled)** radio button and click **OK**.
A new **Video** instance appears in the **Library** pane.
 - d. Drag the **Video** instance on to the stage.
A video instance appears on the stage.

3. Name the video instance.
 - a. If it is not already expanded, click the **Properties** pane's title bar.
The pane expands.
 - b. In the <Instance Name> text box, type a name for the video instance.
4. Add the ActionScript.
 - a. In the **Timeline** pane, click the first frame of **Layer 1**.
 - b. If it is not already expanded, click the **Actions** pane's title bar.
The pane expands.
 - c. Select the first frame of the timeline pane.
 - d. In the **Actions** pane, type your ActionScript (see below).
5. Test the Flash application.
 - a. From the **Control** menu, select **Test Movie**.
A new pop-up window appears displaying video input from your video source, and video capture begins.

Following is a breakdown of ActionScript you insert in the Flash application:

```


1  // Create new Camera and Microphone objects
2  var camera_object:Camera = Camera.get();
3  var microphone_object:Microphone = Microphone.get();
4  // Display the live video in the embedded video instance
5  _root.video_instance_name.attachVideo(camera_object);
6  // Import the AkamaiFCSPublish class
7  var publisher = new AkamaiFCSPublish();
8  // Set the streaming parameters
9  publisher.setCpCode("cp_code");
10 publisher.setPassWord("encoder_password");
11 publisher.setPrimaryServer("primary_entry_point_hostname");
12 publisher.setBackup1Server("backup_entry_point_hostname");
13 publisher.setStreamName("stream_name");
14 publisher.setCamera(camera_object);
15 publisher.setMicrophone(microphone_object);
16 // Connect to the Entry Point and begin pushing the stream
17 publisher.start();

```

Some of the items in this script are drawn from the parameters you entered when you set up your live Flash configuration and stream in the EdgeControl portal:

- **cp_code**. This is the CP code associated with your live Flash configuration as provided on the **Manage Streams** and **Flash Configuration** pages.

- **encoder_password.** This is the password your encoder will use to gain access to the Akamai Entry Points (available on the **Stream Details** page).
- **primary_entry_point_hostname.** This is the Akamai-assigned hostname of your primary Entry Point as provided on the **Manage Streams**, **Stream Details**, and **Flash Configuration** pages.
- **backup_entry_point_hostname.** This is the Akamai-assigned hostname of your secondary Entry Point as provided on the **Manage Streams**, **Stream Details**, and **Flash Configuration** pages.

 *Note: All parameters are case sensitive.*

Using the previous example, an actual ActionScript might appear as follows:

```
1  var cam:Camera = Camera.get();
2  var mic:Microphone = Microphone.get();
3  _root.video.attachVideo(cam);
4
5  var publisher = new AkamaiFCSPublish();
6
7  publisher.setCpCode("9391");
8  publisher.setPassWord("x6je2V");
9  publisher.setPrimaryServer("p.ep9.i.akamaientrypoint.net");
10 publisher.setBackup1Server("b.ep9.i.akamaientrypoint.net");
11 publisher.setStreamName("event@9");
12 publisher.setCamera(cam);
13 publisher.setMicrophone(mic);
14 publisher.start();
```

One thing missing from this ActionScript is a means of stopping the encoder. As it is here, the only way to do so is to close the Web page containing the encoder application. This is easily remedied, however, by adding a control of some sort—say a **Stop** button—and using it to call the AkamaiFCSPublish class's **publisher.stop** method. Alternatively, the next section describes how to apply an automatic timer to the encoder application.

Applying a Broadcast Time Limit

If your event has a set time limit, and you would rather have the encoder stop broadcasting automatically, you can add additional ActionScript in your encoder that will stop pushing the encoded Flash video after a specified time period. To take advantage of this feature, add the following to the end of the aforementioned ActionScript:

```
1  var setInterval_object = setInterval(timerCallBack, time_period);
2  function timerCallBack()
3  {
4      clearInterval(setInterval_object);
5      setInterval_object = null;
6      publisher.stop();
7  }
```

In line 1, `time_period` is given in milliseconds, so if your event will last 15 minutes, the final version will look like this:

```
1 var intervalId = setInterval(timerCallBack, 900000);
2 function timerCallBack()
3 {
4     clearInterval(intervalId);
5     intervalId = null;
6     publisher.stop();
7 }
```

Using the Akamai Broadcaster Encoding Tool

Akamai also makes available a prebuilt Flash video encoder called **Akamai Broadcaster**, consisting of two files, **AkamaiBroadcaster.html** and **AkamaiBroadcaster.swf**. To use the encoder, simply copy the files to the same directory on your encoding computer and open **AkamaiBroadcaster.html** in a Web browser. The encoder appears.

The screenshot displays the Akamai Broadcaster web interface. It features a top section with 'Capture Input' and 'Playback' video preview windows. Below these are three main configuration panels: 'Camera Settings', 'Microphone Settings', and 'Broadcast Settings'. The 'Camera Settings' panel includes fields for Source, Bandwidth (128), Capture Width (160), Capture Height (120), FPS (15), and Quality (0), with an 'Apply' button. The 'Microphone Settings' panel includes fields for Source, Gain (50), Sample Rate (8 kHz), Silence Level (10), and Silence Timeout (2000), with an 'Apply' button and a 'Use Echo Suppression' checkbox. The 'Broadcast Settings' panel includes fields for CP Code, Password, Stream Name, Primary Entry Point, Backup Entry Point, and Playback URL, along with a 'Start Broadcast' button. The Akamai logo is visible in the bottom right corner.

Figure 4-1. The Akamai Broadcaster

Note: Be aware, the Akamai Broadcaster is built around the Adobe Flash Player and its embedded codec, which encodes video at a lower quality than the On2 VP6 codec available in other third-party encoders.

The encoder automatically starts with default parameters you may change as required. Contextual help is available for all parameters and controls; clicking one displays its help text in the text box at the bottom of the encoder. For example, if you click the **Capture Height** text box, the online help message, **The requested capture height, in pixels. The default value is 120.**, is displayed.

Following are the main panels of encoder window along with their respective fields:

- **Capture Input.** This panel displays the output from your video capture device. The display may be enabled or disabled using the **On/Off** button. **Current FPS** indicates the frames per second, and **Audio Level** indicates the relative audio level.
- **Playback.** This panel displays the output of the stream from the Akamai Streaming server. The stream may be played from an Edge server (primary or backup) by clicking the **Playback** button. The display may be enabled or disabled using the **On/Off** button.
- **Camera Settings.** Specifies the desired video capture device and video settings.
- **Microphone Settings.** Specifies the desired audio device and audio settings.
- **Broadcast Settings.** Many of these settings are taken from your Flash configuration.
 - **CP Code.** The Content Provider code associated with your live Flash configuration. Obtain this from your **Manage Streams** or **Flash Configuration** pages in the EdgeControl portal.
 - **Password.** Password required for authenticating the encoder when it connects to the Entry Point. Obtain this from the **Stream Details** page in the EdgeControl portal.
 - **Stream Name.** This is the name that identifies your stream to the Flash playback application. Obtain this from your **Manage Streams**, **Stream Details**, or **Flash Configuration** pages in the EdgeControl portal.
 - **Primary Entry Point.** This is the hostname of the primary Akamai-assigned Entry Point. Obtain this from your **Manage Streams**, **Stream Details**, or **Flash Configuration** pages in the EdgeControl portal.
 - **Backup Entry Point.** This is the hostname of the backup Akamai-assigned Entry Point. Obtain this from your **Manage Streams**, **Stream Details**, or **Flash Configuration** pages in the EdgeControl portal.
 - **Playback URL.** This is generated when the **Start Broadcast** button is clicked and is based on your broadcast settings. You will use this information in your playback application; it includes the Akamai Streaming hostname, server-side application name (“live”), and the name of the encoded stream.

The **Start Broadcast** button establishes the connection to the Entry Points and begins the video broadcast. Once clicked, the button changes to a **Stop Broadcast** button that terminates the broadcast and the connection.

The following steps explain how to use the Akamai Broadcaster.

1. Ensure that audio and video input devices are connected and working correctly.
2. Enter your live streaming parameters.
 - a. Select the camera and microphone settings.
 - To use the default settings, click the **Use Default Settings** check boxes
 - If you wish to use settings of your own choosing, enter them in the appropriate fields and click the **Apply** buttons.
 - b. Enter your broadcast settings: CP code, encoder password, stream name, and primary and secondary Entry Point hostnames.
3. Begin your broadcast.
 - c. Click the **Start Broadcast** button to begin broadcasting. The **Publish URL** field populates.

Depending upon the browser settings, a dialog box may appear requesting access to the camera and microphone.
 - d. If required, click the **Allow** button.

The broadcast begins and the **Capture Input** panel displays the audio and video input.
4. When the event has finished, click the **Stop Broadcast** button to terminate the broadcast and drop the connections to the Entry Points.

Chapter 5. Testing Live Flash Streams

In This Chapter



Testing with the Flash Video Test Player • 55

Troubleshooting with the Flash Video Test Player • 57

After creating your stream in the EdgeControl portal (Chapter 3) and setting up your encoder (Chapter 4), you can test your live stream to ensure it is working correctly. Akamai provides a Web-based Flash video test player tool with which you can do this and more.

Testing with the Flash Video Test Player

Before using Akamai's Flash video test player, be certain your encoder is actively broadcasting video to your Akamai Entry Points, and then use the following procedures.

1. Access the Flash video test player.
 - a. Open your Web browser and navigate to <http://support.akamai.com/flash/>.
The test player appears.

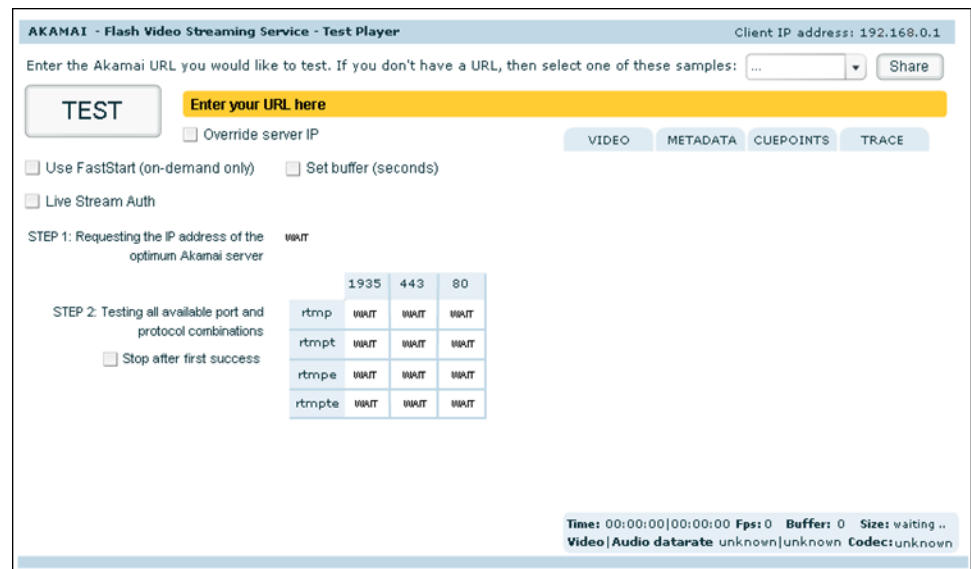


Figure 5-1. Flash Video Test Player

2. Test the stream.
 - a. In the yellow **Enter your URL here** field, enter your stream ARL as provided on the **Stream Details** page in the EdgeControl portal. (See “Viewing Stream Details” on page 30.)

If you are using Dynamic Streaming, the URL's stream name must include the resulting values of the stream name parameter(s) you are using in your encoder. For example, if you are streaming three different bit rates and you entered the stream name `event_%i@9`, here you must use either `event_1@9`, `event_2@9`, or `event_3@9`, depending on which bit rate stream you would like to test.

► *Note: At present, the test player only determines if your link is functional and the stream is broadcasting. It does not test for bit rate changes related to Dynamic Streaming.*

- b. If you wish to test against a specific Akamai Streaming server rather than using Akamai's normal mapping feature, select the **Override server IP** check box and enter the server's IP address in the resulting **Enter IP here** text box. (Refer to "Overriding the Server IP" on page 57 for more information.)
- c. If you wish to override the player's default 3-second video buffer, select the **Set Buffer (seconds)** check box and enter or select a value in the resulting spin box (values to one decimal point (e.g., 5.2) are acceptable).
- d. If you are testing a stream to which Per Play Secure Streaming is applied, select the **Live Stream Auth** check box and enter your per-play token in the resulting yellow text box (e.g., `auth=[stream_token]&caifp=[stream_fp]`).
- e. If you wish the player's **STEP 2** to limit itself to the first successful connection, foregoing subsequent connection attempts on remaining port and protocol combinations, select the **Stop after first success** check box.
- f. Click the **TEST** button to begin playing the stream.

If all is set up correctly, the **STEP 1** process begins, populating with an Akamai Streaming server IP address. This is followed **STEP 2**, which tests all available port and protocol combinations. Lastly, **STEP 3** begins playing the stream underneath the **VIDEO** tab on the right side of the page.

AKAMAI - Flash Video Streaming Service - Test Player Client IP address: 192.168.0.1

Enter the Akamai URL you would like to test. If you don't have a URL, then select one of these samples: video stream Share

TEST rtmp://cp9391.live.edgefcs.net/live/event@9

☐ Override server IP

☐ Use FastStart (on-demand only) ☐ Set buffer (seconds)

☐ Live Stream Auth

STEP 1: Requesting the IP address of the optimum Akamai server ✓ 76.9.1.102

STEP 2: Testing all available port and protocol combinations

	1935	443	80
rtmp	✓	✓	✓
rtmpt	✓	✓	✓
rtmpe	✗	✗	✗
rtmpte	✗	✗	✗

(Bandwidth measured at 12656 kbps)

☐ Stop after first success

STEP 3: Playing the stream using the first good connection

Success - buffer is full and video is streaming over rtmp:1935

VIDEO **METADATA** **CUEPOINTS** **TRACE**

Akamai Traffic Statistics

apacity 1.016 Bit Rate

Pause FullScreen

Time: 00:00:27|00:22:11 Fps: 23 Buffer: 60s Size: 318x180
Video|Audio datarate unknown/unknown Codec: unknown

Figure 5-2. Flash Configuration Testing Page with Video Playback

If you wish to demonstrate your video's playback to others, you can click the **Share** button, which opens a new window containing a playback URL that you can cut and paste.

In addition, using the player's control, you can pause and restart video playback, and you can use the **VIDEO**, **METADATA**, **CUEPOINTS**, and **TRACE** tabs to display their respective data.

Troubleshooting with the Flash Video Test Player

The Flash video test player is also a useful troubleshooting tool, providing both stream data and the ability to specify streaming from a particular Akamai Streaming server. Using it as a go-to tool for both you and your end users can help expedite the problem-solving process.

Using Trace Data

The Flash video test player generates trace data each time you test a video stream. It is essentially a play-by-play report on your stream's status and can be viewed by clicking the test player's **TRACE** tab.

If you are experiencing problems with your stream, you can test it in the test player, select the **TRACE** tab, click the **Copy all this text to your clipboard** button, and paste the text into an e-mail to your Akamai representative. This will facilitate the troubleshooting process.

Overriding the Server IP

This feature of the Flash video test player is helpful in determining whether a problem is occurring at the first or last mile of a stream.

Normally, when testing a stream with the Flash video test player, the Akamai Streaming server is chosen dynamically, just as it would be in a real-world environment, with the result appearing in the test player's **STEP 1** field. You can, however, override this behavior to test a specific server:

1. Open the test player and set it up using the steps in the "Testing with the Flash Video Test Player" section above, but do not click the **TEST** button yet.
2. Select the **Override server IP** check box, and enter the server IP address in the resulting **Enter IP here** field.

The server IP address can be obtained either using the Flash video test player or the **netstat** command.

3. Click the **TEST** button to initiate playback.

For example, if an end user experiences difficulties with your stream, they can provide you the IP address of the Akamai Streaming server to which they are connecting, and you can try to duplicate their problem by testing the stream using the same server. If you do not experience the same issues as the end user, the problem most likely resides at their end, rather than on the Akamai Streaming network.

Chapter 6. Setting Up the Video Playback Application

In This Chapter



Creating a Flash Playback Application • 60

Expediting the Connection • 65

Dealing with Problematic Client-Side Proxy Servers • 67

The final step in setting up Akamai Streaming is to create a Flash playback application to allow your end users to receive your live video stream. For this task, you have two available options:

- You may build it with the **AkamaiConnection** class.
- You may build it without the **AkamaiConnection** class.

Each of these is discussed in this chapter.



Note: The ActionScript examples provided in this chapter are written in ActionScript 2.0. If you are using ActionScript 3.0, refer to the appropriate Adobe documentation for information on its usage.

Akamai Streaming (for Live Adobe Flash Video) Communications

When playing a video stream, Adobe Flash Player initiates communications with Akamai Streaming on port 1935 using either RTMP (Real Time Messaging Protocol) or RTMPE (Encrypted Real Time Messaging Protocol). If traffic is in some way blocked (an interceding firewall blocks the port, for example), Flash Player will make successive attempts on different ports using different protocols.

- RTMP (or RTMPE) on port 1935
- RTMP (or RTMPE) on port 443
- RTMP (or RTMPE) on port 80
- RTMPT¹ (or RTMPTE) on port 80

Be aware, progression through this sequence can cause a significant delay in video playback, and Akamai recommends using your client-side Flash application to override it (see “Expediting the Connection” on page 65). Also, the RTMP and RTMPE protocols are mutually exclusive where the connection attempt sequence is concerned. For example, if you designate RTMP for your connection, RTMPE connections will not be attempted, and vice versa. The exception to this is if you are using

1. RTMPT is tunnelled RTMP (RTMP packets wrapped in HTTP) also called HTTP tunneling.

the AkamaiConnection class, which allows you to customize your port and protocol sequence.

- ▶ *Note: If you have implemented RTMP- and RTMPT-specific firewall rules, you may need to update them to include the RTMPE and RTMPTE protocols.*
- ▶ *Note: RTMPT is only available with Flash Player version 6.0.65 and higher; RTMPE and RTMPTE require version 9.0.115.0 and higher (use the online tool at <http://www.adobe.com/products/flash/about/> to check your version).*

Creating a Flash Playback Application

The Flash application you create for your end users to play your live Flash stream can be as simple or as complicated as you like. Like the Flash video encoder, you can build the most basic application with Adobe Flash software simply by adding an embedded video instance to a blank stage, naming the instance, and adding an ActionScript.

- ▶ *Note: SWF files are cached on the client side. If your browser does not play the latest version of your application, clear its cache and retry the application. Simply refreshing the page will not work.*

Creating a Flash Playback Application for Dynamic Streaming

You have two choices in creating your playback application for Dynamic Streaming. You can either build it using Adobe's new ActionScript API that enables Dynamic Streaming features, or you can use Akamai's utility classes (Open Video Player), available at <http://openvideoplayer.sourceforge.net>. The latter is recommended, as it facilitates implementation since Akamai has already done the work of integrating Adobe's API to work correctly with Akamai Streaming.

- ▶ *Note: Open Video Player is provided as a reference and a best practice guide only. Your Flash developers may modify it or develop their own Dynamic Streaming heuristics, as desired.*

When setting up your playback application, use a buffer time at least two to three times that of the keyframe interval you set up in your encoder. This will provide an optimal experience to your end users when the video stream changes bit rates. For example, if you use the recommended 2-second keyframe interval in your encoder, you should use at least a 4- to 6-second buffer in your playback application.

Also, to avoid unwanted video glitches during bit rate changes, you should ensure the playback application has received **NetStream.Play.TransitionComplete** for the previous transition command before issuing any new transition commands.

For details on implementing Dynamic Streaming in conjunction with Secure Streaming, see "Using an slist to Handle Secure Streaming/Dynamic Streaming" on page 72.

Building a Playback Application Using the AkamaiConnection Class

For added convenience, Akamai provides the **AkamaiConnection** class—available in both ActionScript 2.0 and ActionScript 3.0—to assist your Flash application developers in establishing a robust connection with the Akamai Streaming (for live Adobe Flash Video) service while also implementing the best practices and methods outlined later in this chapter:

- **NetStream.onStatus Event Handling.** See “A Note on the NetStream.onStatus Event Handler” on page 64
- **Connection Expedition.** See “Expediting the Connection” on page 65
- **Client-Side Proxy Penetration.** See “Dealing with Problematic Client-Side Proxy Servers” on page 67

In addition, the class offers other features such as dynamic buffer management and event reporting.

The class always creates a NetConnection and can optionally create a NetStream on that NetConnection. Connections are initiated by a public method and events are used to notify the parent class of connection success, connection failure, errors, and NetStream- and NetConnection-specific events.



Note: Be aware, the AkamaiConnection class focuses exclusively on the connection layer and does not facilitate user-interface elements.

You can obtain the class file from the EdgeControl portal’s documentation area ([Live Streaming >> Documentation](#)); it is packaged with a specification and usage document, which provides complete details on its use and features. The package also contains a series of detailed, working sample files that implement all the methods, events, and properties described in the specification. These serve as good ActionScript references for using this class in real-world Flash applications.

To use the ActionScript 2.0 version of the AkamaiConnection class, simply copy it to the same directory as your FLA file. When you publish the FLA, ActionScript imports the class and compiles it into the resulting SWF file. If you want to store the class in a different location, you can still pull it into your SWF by specifying a class path in your ActionScript.

Building a Playback Application Without the AkamaiConnection Class

If you decide to build a playback application without the AkamaiConnection class, you must call a method called **FCSubscribe** in Akamai Streaming’s server-side application, **live**. The following describes the construction of a simple playback application that does not use the AkamaiConnection class.

1. In Flash CS3 or Flash 8, open a new Flash document.
 - a. From the **File** menu, select **New**.

The **New Document** dialog box appears.

- b. Select **Flash Document** (Flash 8) or **Flash File (ActionScript 2.0)** (Flash CS3) and click **OK**.

A new Flash document appears.

2. Create a new video instance.

- a. If the Library pane is not already present, select **Library** from the **Window** menu.

The **Library** pane appears.

- b. From the pulldown menu on the right-hand side of the **Library** pane's title bar select **New Video**.

The **Video Properties** dialog box appears.

- c. Enter a name in the **Symbol** text box (if desired), select the **Video (ActionScript-controlled)** radio button and click **OK**.

A new **Video** instance appears in the **Library** pane.

- d. Drag the **Video** instance on to the stage.

A video instance appears on the stage.

3. Name the video instance.

- a. If it is not already expanded, click the **Properties** pane's title bar.

The pane expands.

- b. In the **<Instance Name>** text box, type a name for the video instance.

4. Add the ActionScript.

- a. In the **Timeline** pane, click the first frame of **Layer 1**.

- b. If it is not already expanded, click the **Actions** pane's title bar.

The pane expands.

- c. Select the first frame of the timeline pane.

- d. In the **Actions** pane, type your ActionScript (see below).

5. Test the Flash application.

- a. From the **Control** menu, select **Test Movie**.

A new pop-up window appears and your video begins playback.

Following is a breakdown of ActionScript you insert in the Flash application:

```

1  // Create a new NetConnection object
2  NetConnection_object = new NetConnection();
3  // Connect to the Akamai Streaming server
4  NetConnection_object.connect("hostname/live");
5  NetConnection_object.call("FCSubscribe", null, "stream_name");
6  // This is the callback that the server-side ActionScript will call
   with info.code == NetStream.Play.Start (a play on the stream
   will only be issued if it is a "NetStream.Play.Start"). The
   other error code could be "NetStream.Play.StreamNotFound"
7  NetConnection_object.onFCSubscribe = function(info)
8  {
9      if (info.code == "NetStream.Play.Start")
10     {
11         _root.NetStream_object = new NetStream(NetConnection_object);
12         _root.NetStream_object.setBufferTime(time_in_seconds);
13         _root.instance_name.attachVideo(_root.NetStream_object);
14         _root.NetStream_object.play("stream_name");
15     }
16 }


```

Some of the items in this script are obtained from the **Stream Details** and **Flash Configuration** pages in the EdgeControl portal:

- **hostname.** This is either the Akamai-generated hostname (e.g., cp9391.live.edgefcs.com) or one of the host aliases you created.

This is also available from the EdgeControl portal's **Manage Streams** page.

- **live.** This static value is the name of the server-side application
- **stream_name.** This is the name of your stream.

 *Note: All parameters are case sensitive.*

So, using the previous example, an actual ActionScript might appear as follows:

```

1  nc = new NetConnection();
2  nc.connect("rtmp://cp9391.live.edgefcs.net/live");
3  nc.call("FCSubscribe", null, "event@9");
4  nc.onFCSubscribe = function(info)
5  {
6      if (info.code == "NetStream.Play.Start")
7      {
8          _root.ns = new NetStream(nc);
9          _root.ns.setBufferTime(6);
10         _root.myVideo.attachVideo(_root.ns);
11         _root.ns.play("event@9");
12     }
13 }

```

This ActionScript is very simple and the video stream will continue until the Web page containing the playback application is closed or until it is terminated by the Flash video encoder.

If you wish to add a “Stop” control to your playback application, you must do so by calling the `FCUnsubscribe` method, which is also part of Akamai Streaming’s server-side application, `live`. The following lines of script will stop a live stream:

```
1  // Call the server-side method, FCUnsubscribe, with the stream name.
   This in turn calls the nc.onFCSUnsubscribe method in the
   playback application with the info.code ==
   NetStream.Play.Stop.
2  nc.call("FCUnsubscribe", null, "event@9");
3  nc.onFCUnsubscribe = function(info)
4  {
5      _root.ns_in.play(false);
6      _root.ns_in.close();
7      _root.ns_in = null;
8  }
```

► *Note: Be certain to associate this script with a control. Simply adding it to the previous example will cause the application to subscribe and then immediately unsubscribe.*

A Note on the NetStream.onStatus Event Handler

ActionScript’s `NetStream` object has an event handler, `NetStream.onStatus`, that is useful for producing behaviors when status and error messages are posted for the object. If you use this event handler, you may wish to include an additional parameter in your `NetStream.play` method to expedite video playback.

► *Note: The `AkamaiConnection` class automates this solution. If you are using it in your application, no additional ActionScript is required.*

In the aforementioned simple Flash application, the `NetStream.play` method takes the form of:

```
_root.ns.play("event@9");
```

When passed to Akamai Streaming, the server goes through a default search sequence to find the video content:

1. Search for a live stream with this name. If no stream exists, then...
2. Search for a prerecorded stream with this name. If no stream exists, then...
3. Create a new live stream with this name.

Since the stream is known to be live, you can indicate this by including an additional parameter in the `NetStream.play` method using the “`NetStream.onStatus`” event handler. In this case, the method appears as follows:

```
_root.ns.play("event@9", -1);
```

Adding a positive numeric value to the method represents the time point, in seconds, at which to begin video playback. For example, a value of `0` would start playback at the beginning and `10` would begin the playback 10 seconds from the beginning. By

entering a value of -1, you indicate the stream should begin at whatever data point is available at that time, in essence telling the server the stream is live and causing the server to bypass step 2 in the above sequence.

Expediting the Connection

As explained at the beginning of this chapter, Adobe Flash Player normally attempts to communicate with Akamai Streaming in a particular protocol/port progression—RTMP:1935, RTMP:443, RTMP:80, and RTMPT:80 (or RTMPE:1935, RTMPE:443, RTMPE:80, and RTMPTE:80). This default progression has a disadvantage, however, in that Flash Player can take considerable time to progress completely through the sequence resulting in a potentially lengthy delay in starting video playback. As a best practice to overcome this, Akamai strongly recommends overriding the default sequence to attempt simultaneous connections and expedite playback.

To accomplish this, two connections of different types are attempted concurrently: one using the default protocol/port sequence and the other attempting only HTTP tunneling (RTMPT or RTMPTE) over port 80. Whichever connection succeeds first is the one used and the other is dropped.

The AkamaiConnection class performs this automatically. The following ActionScript example illustrates how to implement it when not using the class:

```

1  // Create two NetConnection objects: one for the default protocol/
   port sequence and the other to try HTTP tunneling on port 80
   (RTMPT or RTMPTE).
2  var ncdefault:NetConnection = new NetConnection();
3  var nctunnel:NetConnection = new NetConnection();
4  // Create a flag object to ensure a race condition does not occur
   between the two connections.
5  var connSuccess = 0;
6  // Create two streaming connections: one for the default protocol/
   port sequence and the other to try HTTP tunneling on port 80
   (RTMPT or RTMPTE).
7  ncdefault.connect("rtmp://cp9391.live.edgefcs.net/live");
8  nctunnel.connect("rtmpt://cp9391.live.edgefcs.net:80/live");
9  // Create a NetStream object.
10 var ns:NetStream;
11 // If the default protocol/port sequence (ncdefault) connects
   successfully first, drop and close the RTMPT:80 (nctunnel)
   connection.
12 ncdefault.onStatus = function(ncObj) {
13     if(ncObj.code == "NetConnection.Connect.Success") {
14         if(connSuccess == 0) {
15             connSuccess = 1;

```

```
16         if(nctunnel) {
17             nctunnel.close();
18             ncdefault.call("FCSubscribe", null, "event@9");
19         }
20     }
21 }
22 }
23 // This is the callback that the server-side ActionScript will call
    // with "info.code == NetStream.Play.Start" (a play on the stream
    // will only be issued if it is a "NetStream.Play.Start"). The
    // other error code could be "NetStream.Play.StreamNotFound".
24 ncdefault.onFCSubscribe = function(info)
25 {
26     if (info.code == "NetStream.Play.Start")
27     {
28         // Attach the default sequence NetStream object to the default
            // sequence NetConnection object and begin streaming the video.
29         ns = new NetStream(ncdefault);
30         ns.setBufferTime(6);
31         _root.myVideo.attachVideo(ns);
32         ns.play("event@9");
33         ns.onStatus = function( info ) {
34             }
35     }
36 }
37 // If the HTTP tunneling attempt (nctunnel) successfully connects
    // first, drop and close the default sequence (ncdefault)
    // connection.
38 nctunnel.onStatus = function(ncObj) {
39     if(ncObj.code == "NetConnection.Connect.Success") {
40         if(connSuccess == 0) {
41             connSuccess = 1;
42             if(ncdefault) {
43                 ncdefault.close();
44                 nctunnel.call("FCSubscribe", null, "event@9");
45             }
46         }
47     }
48 }
```

```

49 // This is the callback that the server-side ActionScript will call
    with "info.code == NetStream.Play.Start" (a play on the stream
    will only be issued if it is a "NetStream.Play.Start"). The
    other error code could be "NetStream.Play.StreamNotFound".

50 nctunnel.onFCSubscribe = function(info)
51 {
52     if (info.code == "NetStream.Play.Start")
53     {
54 // Attach the HTTP tunneling NetStream object to the HTTP tunneling
        NetConnection object and begin streaming the video.

55         ns = new NetStream(nctunnel);
56         ns.setBufferTime(6);
57         _root.myVideo.attachVideo(ns);
58         ns.play("event@9");
59         ns.onStatus = function( info ) {
60             }
61         }
62     }

```

Dealing with Problematic Client-Side Proxy Servers

It is possible for Flash video streamed from Akamai Streaming to encounter problems when confronted by certain client-side proxy servers. The problems occur if the proxy server attempts multiple reconnections during a single stream.

In brief, the proxy allows Flash Player to make an initial connection to Akamai Streaming, which uses its intelligent routing technology to select the Streaming server optimal to the end user. If the proxy server attempts subsequent reconnections during the stream, however, the routing process begins again, likely resulting in selection of a different server. Since the new server has no context for the streaming session, a failure results.

To get around this problem, it is necessary to connect to a single Akamai Streaming server for the stream's duration. Circumventing Akamai's routing technology is not desirable, however, so the solution must allow for both of these.

The next two sections discuss the solution both with and without using the Akamai-Connection class. As a best practice, Akamai strongly recommends implementing this solution to avoid any potential problems.

Solving the Problem Using the AkamaiConnection Class

If you are using the AkamaiConnection class, the solution is implemented automatically. The class was created so that the playback application first sends a query through Akamai's routing process to find the optimal Streaming server. The query

then asks the server to retrieve its **ident** file, which is in XML format and contains the its IP address in the `<ip>` tag.

```
<?xml version="1.0" encoding="utf-8" ?>
<fcs><ip>Akamai_Streaming_server_IP_address</ip></fcs>
```

The ident XML file's contents are in turn sent to the playback application, and the server's IP address is extracted and substituted for your hostname. Flash Player then attempts to connect to Akamai Streaming using the IP address. While the IP address is now being used for connection, the playback application retains your hostname for identification purposes to ensure the correct Akamai Streaming account and Flash video are accessed.

Solving the Problem Without the AkamaiConnection Class

If you have chosen not to use the AkamaiConnection class in your playback application, you can still overcome the issue by including the appropriate ActionScript in your application. The solution, while deployed differently, behaves identically to that described in the previous section, and the deployment of the IP address and retention of your hostname are facilitated by two variables, `_global.serverIp` and `_global.hostName`, respectively (lines 2 and 3 below). When you use the `nc.connect()` method, you reference both variables.

```
1  // Establish global variables for customer hostname, server IP
   address, and video stream name.
2  _global.hostName="cp9391.live.edgefcs.net";
3  _global.serverIp;
4  _global.streamName = "event@9";
5  // Create a new NetConnection object.
6  var nc:NetConnection = new NetConnection();
7  // Create a new NetStream object, but do not assign the
   NetConnection object to it yet.
8  var ns:NetStream;
9  // Retrieve the server's ident XML file by issuing an HTTP request.
10 var myXML:XML = new XML();
11 myXML.ignoreWhite = true;
12 myXML.onLoad = myLoad;
13 myXML.load("http://" + _global.hostName + "/fcs/ident");
14 // If retrieval of the ident XML file was successful, call this
   function to extract the IP address from the file.
15 function myLoad() {
16     output.text += "myLoad Function !!\n";
17     if (this.firstChild.hasChildNodes()) {
18         var aNode:XMLNode;
```



```

19 // Use firstChild to iterate through the child nodes of rootNode
20     for (aNode = this.firstChild.firstChild; aNode !=
        null;aNode=aNode.nextSibling) {
21         output.text +=
            aNode.nodeName+":\t"+aNode.firstChild.nodeValue;
22         if(aNode.nodeName == "ip") {
23             // Assign the IP address to the _global.serverIP variable.
24             _global.serverIp = aNode.firstChild.nodeValue;
25             // Substitute the IP address for the hostname, but retain the
                hostname for reference.
26             nc.connect("rtmp://" + _global.serverIp + "/"
                live?_fcs_vhost="+_global.hostName);
27         }
28     }
29 }
30 }
31 // Connect to the Akamai Streaming server.
32 NetConnection.prototype.onStatus = function(ncObj) {
33     output.text += "\nonStatus rtmp://" + _global.serverIp + ":" +
        ncObj.code;
34     if(ncObj.code == "NetConnection.Connect.Success") {
35         nc.call("FCSSubscribe", null, _global.streamName);
36     }
37 }
38 // Begin streaming the video.
39 NetConnection.prototype.onFCSSubscribe = function(info) {
40     if (info.code == "NetStream.Play.Start") {
41         ns = new NetStream(nc);
42         ns.setBufferTime(6);
43         _root.myVideo.attachVideo(ns);
44         ns.play(_global.streamName);
45         ns.onStatus = function( info )
46     }
47 }

```


Chapter 7. Using Secure Streaming

In This Chapter



Secure Streaming Guidelines for Flash Video • 71

Using an slist to Handle Secure Streaming/Dynamic Streaming • 72

Integrating Secure Streaming for Live Flash Video • 73

Akamai Streaming offers an optional feature for Flash video called Secure Streaming, which provides additional control by giving you the ability to prevent unauthorized access to your streams. Scenarios in which this is desirable include:

- Pay per view
- Deep-linking prevention
- Internal Webcast/IP validation
- Affiliate tracking



Note: Secure Streaming is an additional service that you must purchase separately from Akamai. Otherwise, it is unavailable.

Before proceeding, it is strongly recommended you read the *Akamai Secure Streaming Integration Guide*. While the specifics of that document are oriented toward other streaming formats, it also contains important information regarding general Secure Streaming concepts. Familiarizing yourself with it will prepare you for the information presented in this chapter, which focuses on areas specific to Flash video.

Secure Streaming Guidelines for Flash Video

Following are some Secure Streaming guidelines.

- Secure Streaming for live Flash video is currently applied on a per-play basis, not on a hostname/CP code level as is the case with on-demand Flash streaming.
- Secure Streaming for Flash video supports only D- and E-type tokens, and these must not exceed 511 bytes in length.
- All Secure Streaming tokens must include a fingerprint (aifp) parameter, which must change each time you change the stream's Secure Streaming profile. Following is an example of a Secure Streaming token with its fingerprint parameter:

```
auth=damcPcIbVcscXdQcPbnd7d_dnbSaqaycgay-EDrBic=&aifp=1234
```

- Secure Streaming for Flash video does not support the rendering duration (playback-duration) token parameter; Flash Player will continue playback even after

the specified duration is reached. In addition, it is unable to stream an alternative video should authentication fail.

- For Per Play Secure Streaming, the path for token generation should be the stream's name (e.g., `-f event@s9`).
- If you are using the Dynamic Streaming feature you must use an slist (see below).

Using an slist to Handle Secure Streaming/Dynamic Streaming

Because each stream request requires a Secure Streaming token, using Secure Streaming with the Dynamic Streaming feature can pose a challenge. This comes about because switches between different bitrate streams can occur after the Secure Streaming token has expired, causing a stream failure. You could extend the duration of the token's validity, but this is highly undesirable, as it would make it vulnerable to theft and reuse. The best solution here is to use an slist, which allows you to specify multiple streams to which you want the same Secure Streaming token applied.

Using an slist also adds an extra layer of security in that you include the slist in the token generator, so it is passed in both the token and the Flash playback application's ActionScript's `play()` method. When the Akamai Streaming server detects the **slist** keyword in the stream request, it recognizes that it should look for a match within the token. If no match is present, the content is not served.

► *Note: The parameters included in the token generator and those in the slist argument must match **exactly**, else a failure results.*

Additionally, with Dynamic Streaming, if Flash player attempts to switch to a stream not present in the slist the switch is denied, and the current stream continues playing. Also, if any subsequent play attempts provide a new slist, it is ignored.

Implementing an slist

Normally, a Secure Streaming `play()` string takes the form `video@s34?auth=[token]&aifp=[fingerprint]` (e.g., `video@s34?auth=dagbeuw456vghjkoi_df&aifp=hhgg`). Using Dynamic Streaming, however, requires the addition of an slist query.

For example, if your Dynamic Streaming session has three bitrates—`video1000@s34`, `video500@s34`, and `video100@s34`—your slist will take the form `slist=video1000@s34;video500@s34;video100@s34`. (The bitrate delimiter is a semicolon (;).)

When generating the token, the slist string is used as the path, thusly:

```
./gentoken -y d -f "video1000@s34;video500@s34;video100@s34" -r
"flag_0" dagbeuw456vghjkoi_df
```

So, given the following `NetConnection()` and `NetStream()` instances:

```
var nc = new NetConnection();
nc.connect("rtmp://cp9391.live.edgefcs.net/live");
var ns = new NetStream(nc);
```

Your initial `play()` string will include a query string containing both the authentication token and the slist with all valid stream names for the different bitrates associated with the stream. For Example:

```
ns.play("video1000@s10?auth=dagbeuw456vghjkoi_df&aifp=hhgg&slist=video1000@s34;video500@s34;video100@s34");
```

When you determine you need to change bitrates, issue your `play2()` call. For example:

```
ns.play2("video500@s34");
```

There is no need to include the token or the slist in any `play2()` calls subsequent to the first `play()` call. The token is validated on the first call, and the stream name is validated against the original slist on subsequent calls. As previously stated, if the requested stream name is not present in the original slist, the stream will not switch to the new bitrate.

For a full open source implementation of live Secure and Dynamic Flash streaming, see the Open Video Player at <http://openvideoplayer.sourceforge.com>.

Integrating Secure Streaming for Live Flash Video

Secure Streaming integration results in passing the Secure Streaming authentication token to the Akamai Streaming service, allowing playback of the desired video. This is done by passing the token in the `play()` method of your SWF's ActionScript.

Passing the Token to Akamai Streaming

Passing Secure Streaming tokens in your ActionScripts' `play()` method provides you flexibility in the way you apply Secure Streaming:

- **Real-time authentication.** The token is passed to the server when the video is requested, not when the SWF file is requested. This eliminates the possibility of the token expiring before the end user actually attempts to play a video.
- **Facilitating authenticated playlists.** Using the `play()` method to pass your tokens is useful for playlists. In a typical scenario, the `XML.load()` method is used in the SWF's ActionScript to call an XML-formatted playlist from an origin server, the contents of which are used by the SWF to present a list of videos to the end user. When the end user chooses one, an authentication token is generated and is sent to Akamai Streaming in the `play()` method string.


You can pass the token in the `play()` method whether you are using the `AkamaiConnection` class or not. If you are *not* using it, the `play()` method takes the following form:

```
play("[stream_name]?auth=[token]&aifp=[fingerprint]");
```

So, an actual `play()` method might look like this:

```
play("event@s9?auth=iensTmDeWspaHqMpUmzc9a_jviYardrsmtw-HNuXof=&aifp=1234");
```

If you *are* using the `AkamaiConnection` class, its specification provides information for adding the token to the method.

 *Note: An invalid token passed in the `play()` method results in a `NetStream.Play.Failed` event.*