Akamai MultiPlayer for Adobe Flash: User Guide

Contents

Introduction

This document is a quick reference for using the Akamai MultiPlayer for Adobe Flash. The Akamai MultiPlayer is a ready-to-deploy media player application for Flash media content. It is light, flexible, and based on open standards. This user guide will walk you through the features in the Akamai MultiPlayer and describe how to deploy it for use with your Flash media content.

MultiPlayer features

The Akamai MultiPlayer offers a robust platform based on Adobe ActionScript 3 for the playback of a wide variety of streaming and progressive media delivered by the Akamai platform. The figure below shows the playback of a play list of on-demand Flash video streaming content in the MultiPlayer.

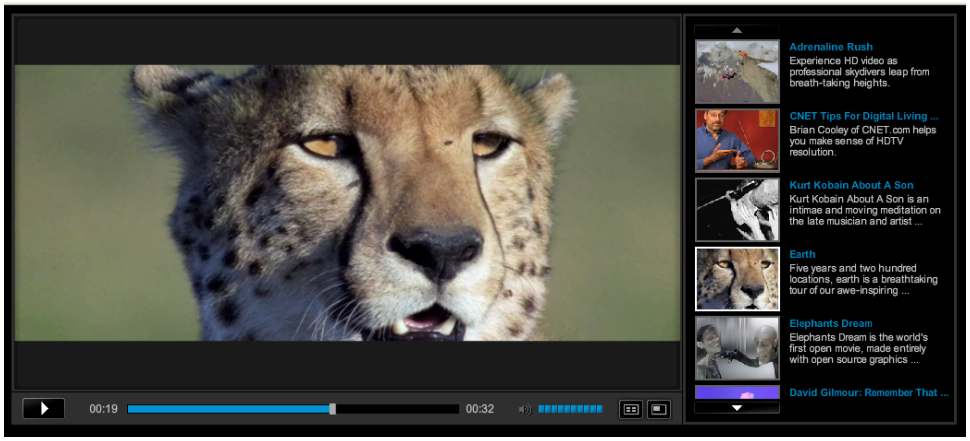

Figure 1: The playback of a play list through the Akamai MultiPlayer.

Specific features include:

- **Dynamically scalable.** All views are re-scaled and positioned each time the flash player is resized.
- **Two layout modes.** Different layouts for the controls and play list that can be toggled at anytime during playback:
  - Overlay, where the controls are overlaid over the video and hide themselves if you mouse-off the player.
  - Side-by-side, where the controls are permanently on the screen and the playlist view is located to the right of the video content area.
- **Four video scaling modes.** Different video scaling that can be toggled at anytime during playback:
  - Fit, where the video is scaled as large as possible to fit within the confines of the player while still preserving its native aspect ratio.
  - Stretch, where the video is is stretched to fit exactly within the confines of the player. Native aspect ratio is not preserved.

  - Native, where the video is scaled to the native size it was encoded at. Note that this size may be larger than the player that is trying to render the video, in which case the video will be centered within the available player space.
  - Native or smaller, where the video will be scaled to its native size unless that is larger than the player in which case "fit" scaling will be invoked.
- **Standard playback controls**. Common playback functionality including play/pause, volume, scrub (seek), current position, duration.
- **Full screen mode.** Maximize the video display dimensions and retain access to the playback controls.
- **Social media options.** Provides access to "link" and "embed" data for player re-distribution.
- **Playlist visibility.** Hide or show the play list during playback.
- **Troubleshooting.** Built-in debug screen to assist with debugging connection and play back problems.
- **Multiple delivery formats.** Play content using a variety of delivery options:

	On-demand Streaming	Live Streaming	Progressive Download	Yahoo! Media RSS Play Lists	Stream OS Metafiles	Dynamic Stream Switching Play Lists
Akamai Media Delivery (AMD)	Yes	Yes	Yes	No	No	Yes
Stream OS	Yes	Yes	Yes	Yes	Yes	No

Notes

1. Play list support currently includes any play list that is compliant with the Yahoo! Media RSS specification and has a file extension of '.xml' or '.rss'.
2. Stream OS metafiles includes types 1, 2, and 4.
3. Dynamic streaming switching includes support for the SMIL 1.0 definition supported by the Adobe FLVPlayback component. The section 'Configuration and deployment' provides more details and Appendix C provides a sample.

Player components

The MultiPlayer has several components. These include:

1. Video display
2. Playback control bar
3. Play list panel
4. Context menu

Figure 2 below shows these components.

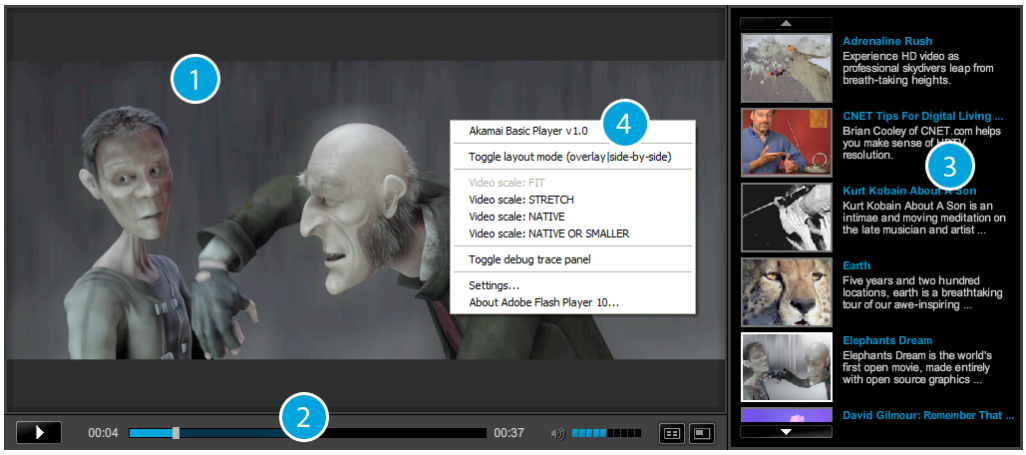

Figure 2: MultiPlayer components.

Video display

The video display is where the video can be viewed.

Playback control bar

The playback control bar enables the end-user to do the following:

- Play and pause the current content playing
- Scrub (seek) through the timeline of the current content playing
- Adjust the volume (audio level)
- View the current and remaining time for the current content playing
- View an indication of how much of the current content playing has downloaded (progressive download only)
- Toggle the play list display mode (displayed only when a play list is in use)
- Toggle full screen display mode
- Provide link and embed code for the player
- View an indication of quality (dynamic streaming only)

Play list panel

The play list panel lists the content in a play list. The play list panel enables end-users to do the following:

- View the metadata for each item in the play list
- View an indication of which item is currently playing
- Scroll up and down through the play list
- Select an item in the play list for playback

Context menu

The context menu is a floating menu that can be activated by right clicking anywhere on the player. The context menu enables end-users to do the following:

- Toggle the layout mode
- Toggle the video scaling mode
- Toggle the debug trace panel
- Toggle the switching mode (manual versus automatic, dynamic streaming only)
- Manually switch to content with a higher data rate (dynamic streaming only)
- Manually switch to content with a lower data rate (dynamic streaming only)

Content preparation

The following should be considered when preparing content:

- The MultiPlayer supports all Flash video and audio formats and in particular all streaming formats supported by FMS 3.5. Encoding guidelines can be found in the Adobe Developer Connection community site at http://www.adobe.com/devnet/.
- The MultiPlayer supports RSS feeds for play lists. Specifically this means that you must comply with the RSS 2.0 and Yahoo! Media RSS specifications. For more information on these specifications, check out the links in Appendix D of this document. It is important to note that if you host the RSS feed on a different domain than where your player (SWF) resides, you will need a cross-domain file in the root path of the web server where the RSS feed resides.
- Dynamic streaming requires a reference to various properties for all of your content derivatives. The MultiPlayer requires the use of a SMIL document for this purpose. For more information on the syntax of the SMIL document, check out the sample in Appendix C of this document. If you host the SMIL document on a different domain than where your player (SWF) resides, you will need a cross-domain file in the root path of the web server where the SMIL document resides.

Configuration and deployment

Use of the MultiPlayer requires configuration and deployment like any Flash (SWF) application. A configuration tool is available, which enables a simple, intuitive approach to configuring and deploying the player.

MultiPlayer configuration tool

This tool will enable you to:

- Intuitively configure the customizable options in the player.
- Preview the player with the options configured.

- Generate and export the syntax (based on SWFObject) necessary to reference the player in your web page.

The tool is written in Flex and requires the Flash Player plug-in. The tool can be accessed from the OVP SourceForge site here:

http://openvideoplayer.sourceforge.net/get-started

For more details on the configuration options, please refer to 'Appendix A: Configuration options' in this document.

Advanced configuration and deployment considerations

Using the MultiPlayer does not require the configuration tool. Configuration and deployment of the player outside of the configuration tool can provide additional flexibility but requires a good understanding of how to deploy a Flash (SWF) application. Specifically:

1. Create a web page that references the MultiPlayer configuration
2. Configure the MultiPlayer
3. Test the player on your web server
4. Deploy the web page

Create a web page

Using the Akamai Basic Media Player requires you to reference the player in a web page just like any Flash/Flex application. Commonly known as "embedding", this can be accomplished many ways and can involve several languages such as HTML and JavaScript, all of which depends on the considerations you make around the configuration options for the Flash Player plug-in and the Basic Media Player, including version support, end-user installation experience, and more. For more details on Flash Player detection and installation, visit the Flash Player Developer Center in the Adobe Developer Connection, a resource for learning about Adobe products and services:

http://www.adobe.com/devnet/flashplayer/detection_installation.html

For a great perspective on the considerations for embedding Flash content, visit:

http://www.alistapart.com/articles/flashembedcagematch

SWFObject is highly recommended for embedding the Akamai Basic Media Player. Appendix B provides an example of the syntax that is required using SWFObject. For more details on this method, visit:

http://code.google.com/p/swfobject/

Configure the MultiPlayer

After you have created your web page, you next need to define the configuration options for the player. All options for the MultiPlayer are configured using the FlashVars property of the Adobe Flash Player plug-in. This property provides an efficient method of dynamically passing the configuration options for the MultiPlayer from the web page when the player is loaded. For more details on the FlashVars property, visit:

http://kb.adobe.com/selfservice/viewContent.do?externalId=tn_16417

Test the player

After you have configured the player the next step is to test the web page that references the player. If possible, you should test it on various versions of web browser applications (Fire Fox 2/3, Internet Explorer 6/7, Safari, and/or Opera) and various versions of operating systems (Windows XP/Vista, Mac OS, and/or Linux) to ensure the optimal end-user experience. Keep in mind that SWFObject uses syntax that provides a wide range of browser and operating system compatibility.

Deploy the web page

Once you have determined that your player is configured and functions as desired, you simply need to deploy your web page into your production web site.

Troubleshooting

Debug trace panel

The built-in debug trace panel provides a log for troubleshooting playback issues. Figure 3 below shows the debug trace panel.

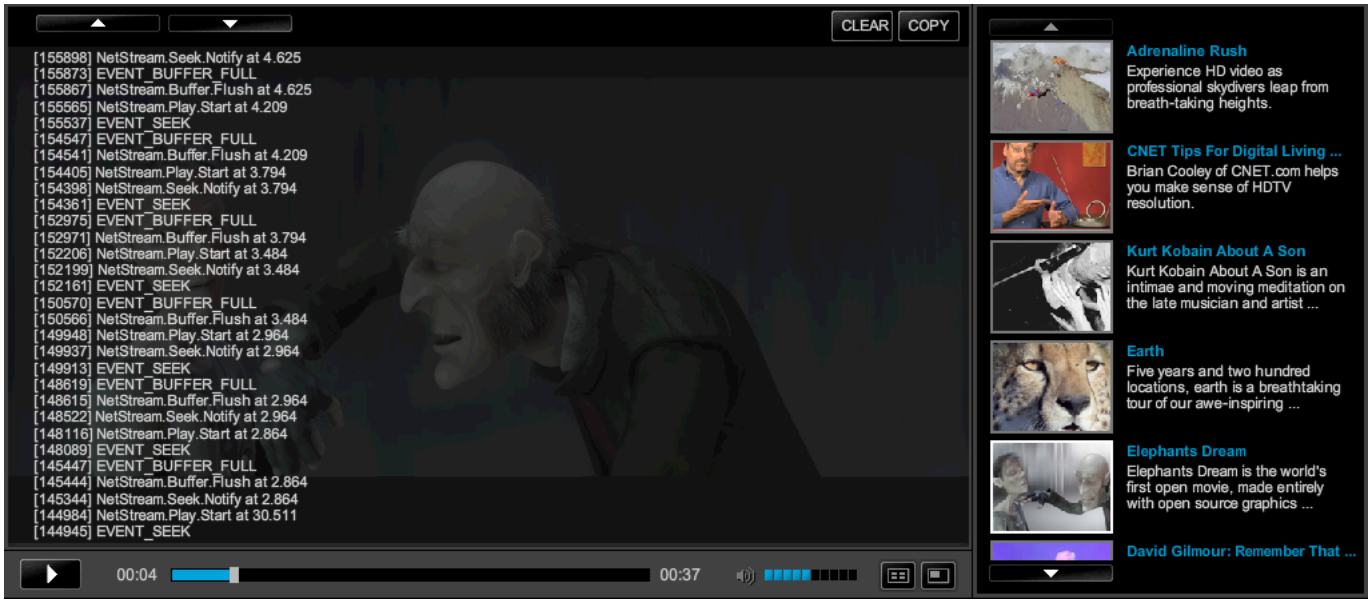

Figure 3: Debug trace panel

The panel displays trace statements for significant application events, including all underlying NetConnection and NetStream status events, with latest events at the top. This can be critical along with end-user details such as operating system, browser, and plug-in. Clear and Copy buttons are provided to respectively clear the contents of the panel and copy the contents to the clipboard to send to others for further analysis.

Appendix A: Configuration options

The following is a list of required and optional configuration options.

Name	Description	Necessity	Defined Values	Default Value	Notes
src	The source URI of the content which the player is expected to play.	Required			This must be a valid and well structured reference to AMD or Stream OS content, or a HTTP link to progressive content on any web server.
mode	The layout mode.	Optional	• overlay • sidebyside	sidebyside	
scaleMode	The starting scaling mode for the video.	Optional	• fit • stretch • native • nativeorsmaller	fit	
frameColor	The HEX value for the frame color.	Optional		333333	"0x" or "#" are not required.

fontColor	The HEX value for the frame color.	Optional		CCCCCC	“0x” or “#” are not required.
themeColor	The HEX value for the frame color. Used for button mouse-over highlights, scrub bar shading, volume control shading, and playlist title font color.	Optional		0395D3	“0x” or “#” are not required.
autostart	Defines whether the video starts playing after the player is loaded.	Optional		true	If set to "false", the player will render the first keyframe in the video to create a splash screen and then stop.
link	The URL which the user can use to link to a mounted instance of the player.	Optional			This parameter must be escaped (URL-encoded) or else it will mask other FlashVar attributes. Note that the button to surface the link/embedded panel will only be visible if either the link or embed parameter has a

					non empty-string value.
embed	The URL which the user can use to embed the player.	Optional			This parameter must be escaped (URL-encoded) or else it will mask other FlashVar attributes. Note that the button to surface the link/embedded panel will only be visible if either the link or embed parameter has a non empty-string value.

Notes

1. In order for full screen mode to be functional, the player must be embedded in the web page with the 'allowFullScreen' set to "true".
2. The minimum embed width for the player varies depending on how it is configured. See the table below for the minimum width in various configurations:

	Overlay Mode		Side-by-Side Mode	
	Full screen or Share (pixels)	Full screen and Share (pixels)	Full screen or Share (pixels)	Full screen and Share (pixels)
Play list content	390	420	680	720
Single file playback	360	390	360	390

Appendix B: Example deployment web page

As an example, view the HTML script below. This requires that the swfobject.js and expressInstall.swf files to be located in the same folder as the HTML file:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">

  <head>

  <title>SWFObject dynamic embed</title>

  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

  <script type="text/javascript" src="swfobject.js"></script>

  <script type="text/javascript">

    var flashvars = {

      src: "rtmp%3A//cp27886.edgefcs.net/ondemand/14808/nocc_small307K.flv",

      autostart: "true",

      themeColor: "0395d3",

      mode: "sidebyside",

      scaleMode: "fit"

    };

    var params = {};

      var attributes = {

      id: "myPlayer",

      name: "myPlayer",

      allowFullScreen: "true"

    };

    swfobject.embedSWF("AkamaiFlashPlayer.swf", "myPlayerGoesHere", "802", "489", "9.0.0",
"expressInstall.swf", flashvars, params, attributes);

  </script>

  </head>

  <body>

  <div id="myPlayerGoesHere">

  <a href="http://www.adobe.com/go/getflashplayer">

  <img src="http://www.adobe.com/images/shared/download_buttons/get_flash_player.gif" alt="Get Adobe Flash
player" />

  </a>
```

```
  </div>

  </body>

</html>
```

Appendix C: Dynamic stream switching

Adobe dynamic stream switching requires derivatives of your source content at specific data rates you define and the media player application must know about these derivatives. This can be accomplished using a SMIL document that has a format supported by Adobe and the FLVPlayback component. Below is a sample file:

```
<smil>
  <head>
    <meta base="rtmp://cp67126.edgefcs.net/ondemand" />
  </head>
  <body>
    <switch>
      <video
src="mp4:mediapm/ovp/content/demo/video/elephants_dream/elephants_dream_768x428_24.0fps_408kbps.mp4"
system-bitrate="408000"/>
      <video
src="mp4:mediapm/ovp/content/demo/video/elephants_dream/elephants_dream_768x428_24.0fps_608kbps.mp4"
system-bitrate="608000"/>
      <video
src="mp4:mediapm/ovp/content/demo/video/elephants_dream/elephants_dream_1024x522_24.0fps_908kbps.mp4"
system-bitrate="908000"/>
      <video
src="mp4:mediapm/ovp/content/demo/video/elephants_dream/elephants_dream_1024x522_24.0fps_1308kbps.mp4"
system-bitrate="1308000"/>
      <video
src="mp4:mediapm/ovp/content/demo/video/elephants_dream/elephants_dream_1280x720_24.0fps_1708kbps.mp4"
system-bitrate="1708000"/>
    </switch>
  </body>
</smil>
```

NOTE: If you host the SMIL document on a different domain than where your player (SWF) resides, you will need a cross-domain file in the root path of the web server where the SMIL document resides.

Appendix D: References

Articles

Adobe Developer Connection: Flash Player Developer Center: Detection and Installation

http://www.adobe.com/devnet/flashplayer/detection_installation.html

Flash Embedding Cage Match

http://www.alistapart.com/articles/flashembedcagematch

Google Code: SWFObject

http://code.google.com/p/swfobject/

Adobe Tech Note: Using FlashVars to pass variables to a SWF

http://kb.adobe.com/selfservice/viewContent.do?externalId=tn_16417

RSS 2.0 specification

http://www.rssboard.org/rss-specification

Yahoo! Media RSS specification

http://search.yahoo.com/mrss

Sample content links

The following sample content links are provided for testing purposes:

- AMD on-demand streaming video (H.264):
  - rtmp://cp67126.edgefcs.net/ondemand/mp4:mediapm/ovp/content/demo/video/elephants_dream/elephants_dream_768x428_24.0fps_608kbps.mp4
- AMD progressive download video (H.264):
  - http://mediapm.edgesuite.net/ovp/content/demo/video/elephants_dream/elephants_dream_768x428_24.0fps_608kbps.mp4
- AMD live streaming video (H.264):
  - rtmp://cp34973.live.edgefcs.net/live/Flash_Live_Benchmark@632
- AMD dynamic streaming:
  - http://mediapm.edgesuite.net/ovp/content/demo/smil/elephants_dream.smil
- AMD on-demand streaming audio only (MP3):
  - rtmp://cp67126.edgefcs.net/ondemand/mediapm/ovp/content/test/video/nocc_small.mp3
- Stream OS on-demand streaming video, type 4 metafile (H.264):
  - http://products.edgeboss.net/flash/products/content/demo/video/elephants_dream_768x428_24.0fps_608kbps.mp4?xmlvers=4
- Stream OS progressive download video (H.264):
  - http://products.edgeboss.net/download/products/content/demo/video/elephants_dream_768x428_24.0fps_608kbps.mp4
- Stream OS RSS feed (on-demand streaming video, VP6):
  - http://rss.streamos.com/streamos/rss/genfeed.php?feedid=1675&groupname=products
- Stream OS RSS feed (progressive download video, VP6):
  - http://rss.streamos.com/streamos/rss/genfeed.php?feedid=1679&groupname=products