SMH Player

User Guide

Overview

Websites use media players to give viewers the rich multimedia experiences they need and expect. An ideal player is full-featured and easy to use.

SMH Basic Player is an out-of-the-box, feature-rich, media player suitable for content owners. It is designed to be your simplest deployment solution.

The player is based on **Open Source Media Framework (OSMF)**. OSMF is a pure ActionScript® 3.0 framework that gives developers complete flexibility and control in creating their own rich media experiences. For more information on Open Source Media Framework, go to www.osmf.org.

Features

- Playback for a wide variety of content types, including files of type FLV, SWF, F4V, MOV, MP4, and MP3.
- Support for both standard and advanced delivery methods, including progressive download, RTMP streaming, RTMP dynamic streaming, HTTP dynamic streaming, and live streaming.Content.
- Visual progress bar element for display only, showing current playback position in song
- Controls to pause playback, seek, fullscreen, and adjust or mute volume.
- Control of capabilities such as autoplay, account, media, and more, without the use of Flash authoring tools.
- Live and VOD
- Built-in multi-bitrate logic with customizable switching rules
- Efficient connection logic for quick start playback
- Buffer management for minimal playback interruptions
- Multiple types of media - video, audio, images, and SWFs (.m3u for playlist)
- Feature plug-ins also cover things like closed captioning and SMIL support, analytics, and advertising services.
- Debugger

Flash Security Settings

The Adobe Flash Player by default restricts simultaneous access to items on your local computer and the internet, which prevents websites from reading your files. To test this Flash object, which plays streaming content from the internet, on your local drive you must add the location of this Flash object to the “Always trust files in these locations” list using the link below:

Adobe Flash Global Security Settings Panel
http://www.macromedia.com/support/documentation/en/flashplayer/help/settings_manager04.html

System Requirements

The players support the same operating systems as the Flash Player software does, and each requires Flash Player 10.0 to be installed.

The following are the basic system requirements to run the players. See the Adobe Flash Player system requirements for additional information relevant to your system.

Specification	Minimum	Recommended for High Definition (HD)
Resolution	1024x768, 1280x720	1920x1080
Processor speed	1 Ghz	2 Ghz
RAM	1 GB	2 GB
VRAM	128 MB	512 MB

Installing SMH Basic Player

The SMH player is developed to play only content hosted on streamingmediahosting.com

1. Obtain the Player zip file.
2. Find the root directory for your web server. It is typically named “htdocs”, “html”, or “public_html”.
3. Create a folder inside the root directory. Open the zip file, and extract the zip contents there.
4. Open your browser and point it to the Player.html demo page that is included in the zip file. Run the demo to confirm that you have successfully installed the player.

Running SMH Basic Player

To run SMH Basic Player, you create a small amount of required HTML code yourself, then embed this code in your web page. There are four pieces of information that browsers require to run the SMH Basic player:

- A file name of the content to play.
- An account from SMH.
- A URL for the location where you installed the player.
- Height and width values for the size of the player window to display. You can specify these either in pixels or as a percentage of the size of the browser window.

You specify each of these using HTML object and embed tags. (Using both object and embed is recommended to ensure backward compatibility with older versions of some browsers.)

You will use Flash variables (FlashVars) to pass along configuration information to the player. When you do so, you instruct the player to ignore its default settings and use your customized choices.

Following the exact syntax for FlashVars is required. First, FlashVars must be placed within HTML object and embed tags. Second, the format of the FlashVars must be a set of one or more "name=value" pairs. Third, an ampersand (&) delimits each pair; for example, "name1=value1&name2=value2".

Required Flash variables (FlashVars) for player configuration:

account: Streaming Media Hosting CDN account name.
media: The path to your media file or a playlist.m3u file.

The remaining required variables are:

- The path to the player.swf file.
- The height and width of the player window.

The FlashVars and other required variables marked in bold in the code below provide an example of the proper syntax.

You can customize the following HTML code sample for your page. Simply replace the sample values in the code with values for your own content location, player location, and size.

```
<object width="700" height="290">
  <param name="movie" value="Player.swf"></param>
  <param name="flashvars"
      value="media=mp4:myMp4Media
          &account=myAccount"> </param>
  <param name="allowFullScreen" value="true"></param>
  <param name="allowscriptaccess" value="always"></param>
  <embed
    src="Player.swf"
    allowscriptaccess="always"
    allowfullscreen="true" width="700" height="290"
    flashvars="media=mp4:myMp4Media
          &account=myAccount">
  </embed>
</object>
```

You can also use the SWFObject 2 embed method to embed the player. This method uses HTML and JavaScript to embed the player in your website and requires the path to the swfobject.js file.

An example of the required variables used in JavaScript with the dynamic SWFObject 2 embed method are marked in bold below:

```
<!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></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 = { account:”myAccount”, media:”mp4:myMedia”
    swfobject.embedSWF("Player.swf", "flashContent", "700", "290", "flashvars");
    </script>

  </head>
  <body>
    <div id="flashContent">
      <p>Alternative content</p>
    </div>
  </body>
</html>
```

Configuring SMH Basic Player with FlashVars

The previous section describes how to use FlashVars to specify the location of the content you want to play and the account it's on. You can also use FlashVars to pass along configuration information to the player. When you do so, you instruct the player to ignore its default settings and use your customized choices.

The following is list of supported FlashVars. **Note: they are case sensitive**

- **autoplay** - tells the player whether it would play the stream on start. (defaults to true)
- **buffertime** - sets the buffer time of the streams. (defaults to 2)
- **autohide** - tells the player to hide the controls. If 'autoplay' is false, it will remain showing. Once the play button is hit, it will activate the auto hide feature. (defaults to true)
- **poster** - allows you to specify an image to use for the first frame. This only works when 'autoplay' is set to false.
- **debug** - loads debugger modules from (relative link) modules/DebugModule.swf, This allow users to see why things might not be working.

Creating an SMH Basic Player Playlist

By default, the SMH Player supports a playlist via an (m3u) file. An M3U file is a plain text file that specifies the locations of one or more media files that the media player should play. Each line carries one specification. The specification can be one of:

- an absolute local pathname e.g., C:\My Music\Heavysets.mp3
- a local pathname relative to the M3U file location e.g. Heavysets.mp3

- a URL.
- a file name

The M3U file can also include comments, prefaced by the "#" character. Here is an example of an M3U file on the Windows platform.

```
#EXTM3U
C:\Documents and Settings\I\My Music\Sample.mp3
C:\Documents and Settings\I\My Music\Greatest Hits\Example.flv
```

To create your own M3U file follow these steps:

1. create a plain text document
2. copy and paste the code above and modify the lines to correspond to your media
3. save the file
4. rename the file to change it's extension from .txt to .m3u;

By default, file extensions for known file types are hidden in Windows. You may have to change this setting in order to view and change the file extension.

If you want to see a graphical representation of your playlist file you'll need to change the m3u file to include directives and install the playlist plugin. Sample.mp3 and Example.flv are the media files. 123 and -1 are the lengths in seconds. A length of -1 may be used when the media file is a streaming file, as there is no predefined actual length value. The value after the length is the title to be shown, followed by the artist field and then the path to a thumbnail.

Example playlist.m3u with directives:

```
#EXTM3U

#EXTINF:123,Sample title ^ Sample artist ^ Sample thumbnail
C:\Documents and Settings\I\My Music\Sample.mp3

#EXTINF:-1,Example title ^ Example artist ^ Example thumbnail
rtmp://flash.streamingmediahosting.com/Example
```

Once you've created your M3U file, you can play it by including it as the value for media.

```
Flashvars=”account=myaccount&media=playlist.m3u”
```

Configuring SMH Basic Player with Plugins

An OSMF plug-in is nothing more than code that hooks into the framework to provide additional or customized functionality. The following is a list of plug-ins that SMH will support.

- Playlist plugin
- Watermark plugin

To use a plug-in, just include a flashvar with a url of where the plug-in is located. The flashvar will have to begin with the prefix "plugin_" in order for the player to know it has to load it.

```
Flashvars=”account=myaccount&media=playlist.m3u&plugin_playlist=plugins/PlaylistPlugin.sw
f”
```

Below is an example of using a plug-in with the SWFObject embed method.

```
var flashvars = {account:”myaccount”, media:”myplaylist.m3u”,
plugin_playlist:”plugins/PlaylistPlugin.swf” };
```

Configuring SMH Basic Player with Modules

Modules allow the player to remain low in size. It allows the player to separate the features that won't be used by most users, and allows the player to load them in at run-time. The difference between a plugin and a module is that a plugin loads in to the OSMF framework, the modules load into the SMH Studio Framework. A Module relates to other parts of the player that are not a part of OSMF.

The debug module allows the user to see what's going on under the hood. If the player isn't working for some reason, load up this module.

In order to view the debugger, there are a few steps you must follow:

- Download the module
- Place the module in one of the following directories:
    1. A “modules” folder relative to your players html file.
    2. The root folder.
    3. Relative to the player’s html file.
- Add the FlashVar title “debug” and make it equal to “true”.

The example below shows the correct syntax:

```
Flashvars=”account=myaccount&media=playlist.m3u&debug=true”
```

Things to Note

- Because the player needs variables passed from the browser, you will not be able to run the player in standalone mode, only through the browser.
- Thumbnail images for the playlist plugin are fixed at a resolution of 32x32 pixels.

Technical Support

Streaming Media Hosting Technical Support is available to provide basic assistance learning to use your new custom solution, however please understand that our professional services team is unable to provide comprehensive training to all technical support engineers for every custom solution built! Should you have a question our support team is unable to answer they will be happy to escalate your question to our professional services team and coordinate a quick reply.

Please note that our technical support teams are unable to assist with embedding Flash objects within your website, as every site is unique and may require different considerations. If you are unfamiliar with embedding Flash objects in HTML we suggest the following article to start:

Detecting Flash Player versions and embedding SWF files with SWFObject 2
http://www.adobe.com/devnet/flashplayer/articles/swfobject.html