OSMF Release Samples

Walkthrough: Getting Started with the Strobe Media Playback

In this walkthrough we will download and configure and playback media with the Strobe Media Playback player. The Strobe Media Playback player is a pre-compiled, ready-to-deploy SWF file that is configured via the HTML source and the FlashVars.

The currently supported FlashVars are:

NAME	Value	DESCRIPTION
src		The location of the media file
loop	true, false*	Restart media when complete
autoPlay	true, false*	Automatically start media
controlBarPosition	bottom*, none, over	Position of the control bar
backgroundColor		Background color of the player
streamType	liveOrRecoded*, live, recorded, dvr	Media stream type. The DVR functionality only works if streamType = dvr. Otherwise you would only see the recorded part (if the mode is recorded) or the live part (if mode is live OR if mode is liveOrRecorded.
autoHideControlBar	true*, false	Auto-hide the control bar
scaleMode	letterbox, none, stretch, zoom	Controls how the media should be scaled

* Default value

This walk through will show a basic use of some of the most common configuration properties, including loading dynamic plugins and basic namespaces for their meta data.

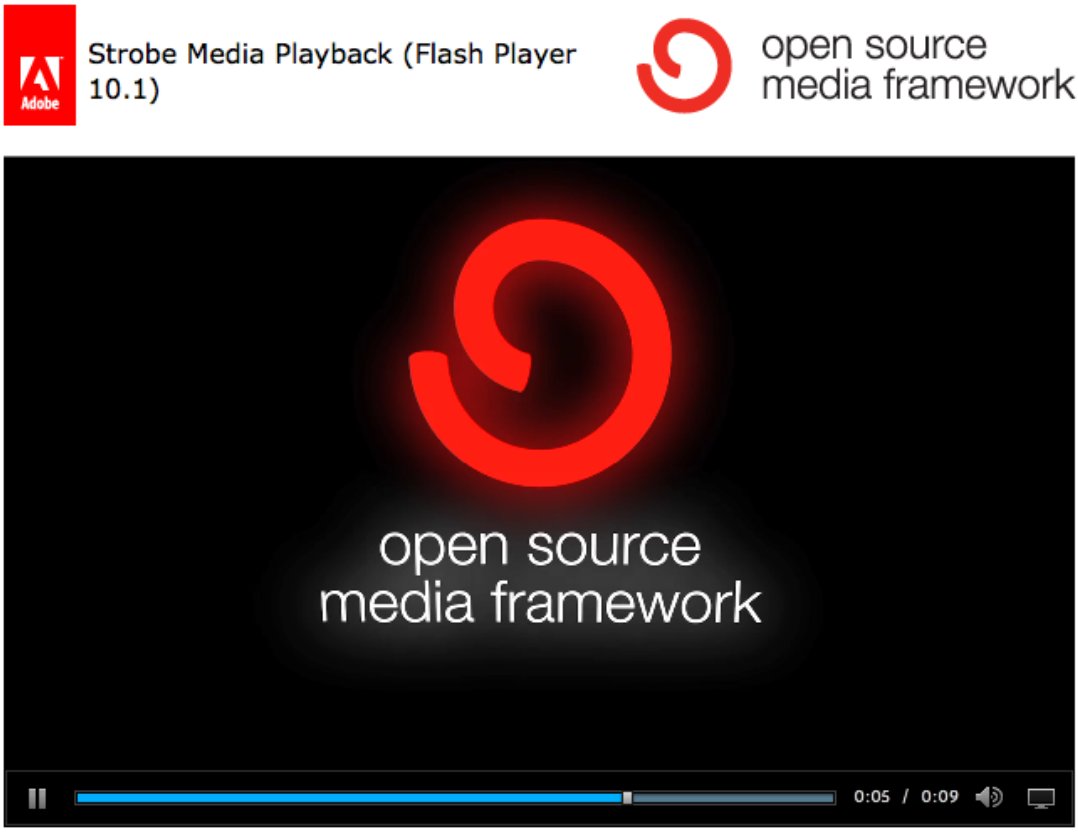
Downloading the Strobe Media Playback

1. Download the StrobeMediaPlayback.zip from http://www.osmf.com/downloads/StrobeMediaPlayback.zip

2. Unzip the contents of the Player to a directory of your web server.

3. The zip file contains 2 versions of Strobe Media Playback - one version compiled for Flash Player 10.0 and another version compiled for Flash Player 10.1 (The 10.1 version is required for DRM & HTTP Streaming). You must have Flash Player 10.1 installed for the 10.1 version of the player to work correctly (http://labs.adobe.com/downloads/flashplayer10.html).

4. Open one of the StrobeMediaPlayback.html files (/10.0/ StrobeMediaPlayback.html or /10.1/StrobeMediaPlayback.html) in a browser. You should see the OSMF logo video play.



Configuring The Strobe Media Playback

5. Save the StrobeMediaPlayback.html file as a new file named StrobeMediaPlayback_config.html and then open this file in a text editor.

6. Locate the 'autoHideControlBar' property and change this from 'false' to 'true'.

7. Update the property 'controlBarPosition' from 'bottom' to 'over'. The code will now look something like this:

8. Open StrobeMediaPlayback_config.html file in a browser. Now the control bar should be hidden until you roll over the player and the control bar should now be positioned over the video that is playing.

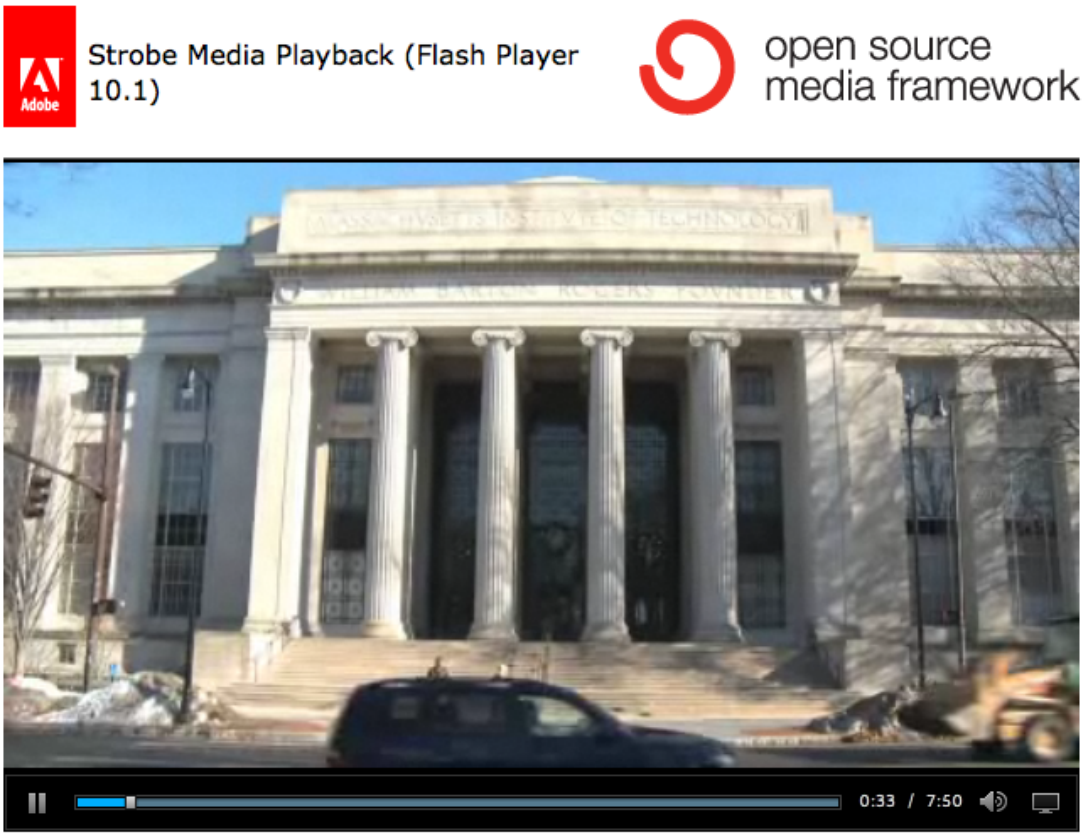
9. Return to the StrobeMediaPlayback_config.html file.

10. Locate the JavaScript that sets the value of the parameters variable. This is where the Strobe Media Playback is configured.

11. Notice that the src attribute is set to an http path - currently the media player is loading and playing a progressive video.

12. Update the source value to rtmp://cp67126.edgefcs.net/ondemand/mediapm/osmf/content/test/akamai_10_year_f8_512K or another streaming URL of your own.

13. Save the file and run it in a browser. The new video should play without any issues.



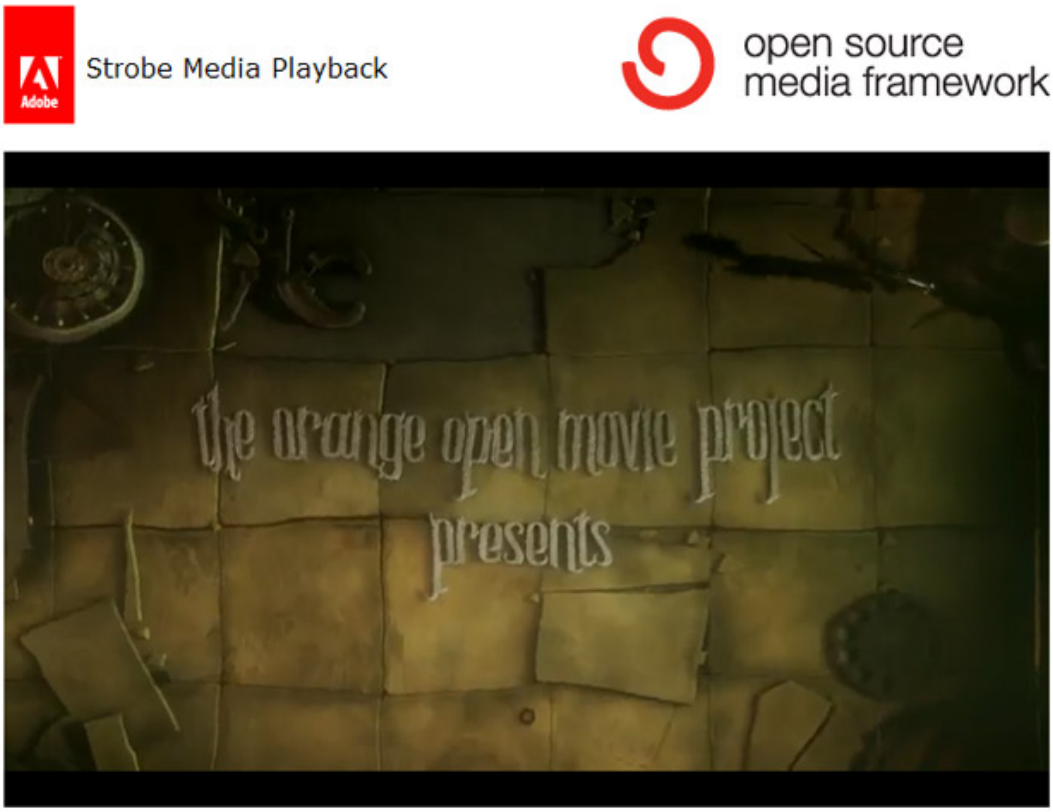
14. Be sure to check out the PDF Using_Strobe_Media_Playback.pdf included in the StrobeMediaPlayback.zip for additional configuration settings.

Dynamic Streaming with Strobe Media Playback

The Strobe Media Playback player also supports dynamic streaming. If the source is set to the URL for a F4M file, you can specify multiple streams and Strobe will switch between the streams as the user's bandwidth fluctuates in order to use the most efficient stream for the amount of bandwidth available. F4M is an XML file in a particular format, which you can find the specifications for on the OSMF website (http://opensource.adobe.com/wiki/display/osmf/F4MElement+Specification).

15. Open the StrobeMediaPlayback_config.html again and switch the src attribute in the HTML to "http://mediapm.edgesuite.net/osmf/content/test/manifest-files/dynamic_Streaming.f4m".

16. Save the file and open it up in a browser. The new video should play and adjust for any bandwidth changes



It is also possible to do dynamic streaming using a SMIL file as the source. However, this requires you to load in the SMIL plug-in, which is available from the OSMF SVN repository. Fortunately, loading plug-ins with Strobe is the next subject we'll cover.

Configuring Plug-ins

The following section will illustrate how to configure the Strobe Media Playback player to load in an external Plugin from a remote location. It requires the use of a local or remote web server to properly test due to local FlashPlayer security.

17. Return to the StrobeMediaPlayback_config.html add the following key value pairs to the parameters object.
    1. plugin_simple: "http://osmf.realeyes.com/plugins/simple/SimpleOSMFPlugin.swf"
    2. simple_namespace: "http://osmf.realeyes.com/plugins/simple"
    3. simple_message: "The plugin has loaded."
    4. The code should look something like this:

```
var parameters =
     { id: "1"
     , src: "http://mediapm.edgesuite.net/osmf/content/test/
logo_animated.flv"
     , autoPlay: "true"
     , width: "638"
     , height: "400"
     , autoHideControlBar: "true"
     , controlBarPosition: "over"
```

```
      , plugin_simple: "http://osmf.realeyes.com/plugins/
simple/SimpleOSMFPlugin.swf"
      , simple_namespace: "http://osmf.realeyes.com/plugins/
simple"
      , simple_message: "The plugin has loaded."
      };
```

NOTE: Plugins are configured by first naming the plugin 'plugin_simple' (the '_' character separates the plugin name definition as well as the property name for the value to be sent to the plugin.

18. This configures the player to load the SimpleOSMFPlugin.swf. When the plugin loads it calls the JavaScript alert() method via ExternalInterface and passes the message defined by the 'simple_plugin' property.

19. Save the file and run it in the browser (*NOTE: To load plugins the StrobeMediaPlayback_config.html must be loaded from an http:// url*), not directly from your filesystem. This is due to Flash Player cross domain security. The video should play back after the JavaScript alert has displayed and OK is clicked to clear the alert.