Open Video Player

Getting Started For Adobe Flash and Adobe Flex Developers

Last Update: 1/21/10 2:30 PM

Overview

The Open Video Player (OVP) code base for ActionScript 3.0 enables Adobe Flash and Adobe Flex developers to quickly create media players for streaming content from Adobe Flash Media Server (FMS), including Content Delivery Networks (CDNs), or progressively downloaded content over HTTP.

For an explanation of the differences between *progressive download* and *streaming*, see "Flash Video Learning Guide".

Why Do I Need This?

"Adobe Flash has the *FLVPlayback*, *MediaDisplay*, *MediaPlayer*, and other components for playing Flash Video while Adobe Flex Builder ships with the *VideoDisplay* component, so why do I need the OVP classes?"

The OVP ActionScript 3 classes remove some of the pain of delivering Flash Video over the Internet that the components mentioned above do not natively address.

Specifically, the OVP classes provide:

1. **A robust connection quickly**. You don't need to worry about dropping back to tunneling over HTTP if your users are behind a corporate firewall and are unable to connect to an FMS server over RTMP. The *OVPConnection* class returns a working connection quickly, trying several ports and protocols in a non-linear fashion.
2. **Bandwidth measurement**. A simple asynchronous method call and a corresponding event give you estimated bandwidth for an FMS or CDN hostname connection. For progressive downloads over HTTP, a class called *HTTPBandwidthEstimate* is available.
3. **A leaner SWF file.** Simply dragging the FLVPlayback component onto an empty stage in Flash CS4 and compiling creates a 60KB SWF file. The bare bones OVP sample mentioned in Step 3 below is only 12KB.
4. **Some events are missing from the Flash Player.** Currently, the Flash Player does not dispatch pause and resume events for progressively downloaded content. The OvpNetStream class will dispatch these events for progressive content if you ask it to.
5. **A complete dynamic streaming (multi-bit rate) solution.** The OvpDynamicNetStream class provides a client side solution to the new dynamic streaming features of FMS for ondemand streams.
6. **ActionScript 3 classes implementing client-side logic specific to CDN behavior and special features**, such as connection and stream authentication, connecting and playing live streams, and the Akamai RandomSeek™ service, for example.
7. **Media RSS parsers**. ActionScript 3 classes that can load a Media RSS playlist parse it and allow you to filter the result set with "like" or "all" queries. An ActionScript 3 class that can load and parse XML metafiles returned by CDN-specific services such as Akamai's Stream OS service.

Step 1 – Get the Latest Release

Get the latest releases as a zip files from sourceforge.net.

You can find them here: https://sourceforge.net/projects/openvideoplayer/

Step 2 – Get Familiar with the code and the documentation

Unzip the release files from Step 1 on your local file system; you will see the following directory structure:

```
/ovp
    /trunk
        /flash
            /core
                /bin
                /docs
                    /asdocs
                /src
                    /com
                        /akamai
                            /net
                            /rss
                    /org
                        /openvideoplayer
                            /advertising
                            /cc
                            /events
                            /net
                            /parsers
                            /plugins
                            /rss
                            /utilities
                            /version
            /players
            /samples
```

The */core* directory

The */src* directory contains the core OVP classes. These classes handle connecting to, and playing live, ondemand, and multi-bit rate streams, closed captioning support, dynamic cue points, RSS parsing, advertising and plugin interfaces, utility classes and more.

Under the */docs/asdocs* sub-directory you will find the documentation generated from Adobe's *ASDoc* utility (for more information on *ASDoc* click here). To begin perusing the code documentation open the file called *index.html* in the */asdocs* sub-directory. This documentation shows the ActionScript classes, their properties, methods, and events.

The */players* directory

This directory contains an Akamai-specific media player capable of playing every type of live and ondemand media possible over the Akamai network. Simply load the FLA file into Flash CS4 and hit Publish.

The */samples* directory

This directory contains both Flash CS4 and Flex 3 sample media players demonstrating various aspects of the OVP core code base.

The Flash CS4 samples will load and compile in Flash CS4 only. Simply load the FLA files into the Flash authoring environment and publish (shift-F12). The publish settings for each FLA file contain a relative path to the OVP SWC file (static library).

For the Flex samples, open Flex Builder 3 and select "File->Import->Flex Project…" and browse to the */samples/flex* directory. This directory contains a Flex project that includes all of the Flex samples as Flex application files (MXML files). To run a sample, simply right click and choose "Run Application".

Step 3 – Understand a very Simple HTTP Media Player in Flash CS4

Find the two files "SimplePDL.fla" and "SimplePDL.as" in the */samples/flash* folder mentioned above.

This Flash project and its corresponding ActionScript 3 class are very bare bones, but demonstrate the basic steps needed to get you playing media quickly using the OVP code base.

1. Open "SimplePDL.fla" in Adobe Flash CS4.
2. Click on the stage and notice in the properties window for the FLA file the "Class:" field contains "SimplePDL". The compiler will look for a file called "SimplePDL.as" and within that file, a class called "SimplePDL".
3. From the File menu, open the "SimplePDL.as" file.
4. Look at the constructor (the method called "SimplePDL()"), notice we "new" an OvpConnection object and add the minimal, necessary listeners. Next we call the "connect" method passing "null" as the only argument, this is consistent with the Adobe NetConnection class for playing progressive downloads over HTTP.
5. Look at the method called "netStatusHandler". This method is our listener for NET_STATUS events fired by the OvpConnection class we've instantiated in the constructor. When we receive a status of "NetConnection.Connect.Success" we call the "connectedHandler" method.
6. Take a look at the "connectedHandler" method. Here we instantiate an OvpNetStream object and give it the OvpNetConnection object, add our necessary event listeners, attach the OvpNetStream object to the Video object on the stage, and call the "play" method.

For more comprehensive examples of the OVP core code, take a look at the other Flash CS4 and Flex samples included in the */samples* directory.

[End of document]