OSMF HLS Plugin

osmfhls.kutu.ru

osmfhls@kutu.ru

November 28, 2012

Contents

About

OSMF HLS Plugin – is a plugin for OSMF based players, which provide capabilities to playback Apple's HLS format M3U8 files.

Features

- Allows Playback of Apple's HLS Format in OSMF based player;
- Works with Multiple Qualities (Multi-Bitrate);
- Supports VOD and Live streams;
- Decodes H.264 Video and AAC / MP3 Audio;
- Decrypt AES-128 Encrypted streams;
- Alternate audio streams support.

M3U8 Tags supported

- `#EXTINF`
- `#EXTM3U`
- `#EXT-X-DISCONTINUITY`
- `#EXT-X-ENDLIST`
- `#EXT-X-KEY` (not recommended for streams greater than 2mbit/s)
- `#EXT-X-MEDIA` (only for `TYPE=AUDIO`, `DEFAULT=NO` and `AUTOSELECT=NO`)
- `#EXT-X-MEDIA-SEQUENCE`
- `#EXT-X-STREAM-INF` (with `BANDWIDTH` and `RESOLUTION` parameters)
- `#EXT-X-TARGETDURATION`

Examples

This section will describe some use cases of OSMF HLS Plugin. Means that you have minimal experience with HTML and JavaScript.

Simple embed with default SMP player and OSMF HLS Plugin

You find complete code of this example in `simple.html`.

First we start with importing `swfobject` in head section of html file:

```
<script type="text/javascript" src="http://yandex.st/swfobject/2.2/swfobject.min.js"></script>
```

Then add JavaScript code for embed SMP player on page:

```
var flashvars = {
    // M3U8 url, or any other url which compatible with SMP player (flv, mp4, f4m)
    src: ""
    // url to OSMF HLS Plugin
    , plugin_m3u8: "OSMFHLSPlugin.swf"
};
var params = {
    // hardware rendering video
    wmode: "direct"
    // self-explain parameter
    , allowFullScreen: true
};
var attrs = {
    name: "player"
};

swfobject.embedSWF(
    // url to SMP player
    "StrobeMediaPlayback.swf",
    // div id where player will be placed
    "player",
    // width, height
    "800", "488",
    // other parameters
    "10.2", null, flashvars, params, attrs
);
```

And finally add `div#player` in body section of html file:

```
<div id="player">
    <!-- this paragraph will be shown if FlashPlayer unavailable -->
    <p>
        <a href="http://www.adobe.com/go/getflashplayer">
            <img src="http://wwwimages.adobe.com/www.adobe.com/images/shared/download_buttons/
                get_flash_player.gif" alt="Get Adobe Flash player" />
        </a>
    </p>
```

```
</div>
```

GTrack plugin

You find complete code of this example in `gtrack.html`.

First, complete previous section. After that, add this JavaScript code before `flashvars` to define GTrack configuration parameters:

```
// http://code.google.com/p/reops/wiki/GTrackPlugin
var gTrackPluginConfig = '<value key="reTrackConfig" type="class" class="com.realeyes.osmf.plugins.
    tracking.google.config.RETrackConfig"> \
    <account>UA-12345678-9</account> \
    <url>http://example.com</url> \
    <event name="percentWatched" category="video" action="percentWatched"> \
        <marker percent="0" label="start" /> \
        <marker percent="25" label="25PercentView" /> \
        <marker percent="50" label="50PercentView" /> \
        <marker percent="75" label="75PercentView" /> \
    </event> \
    <event name="complete" category="video" action="complete" value="1" /> \
    <event name="pageView" /> \
    <updateInterval>250</updateInterval> \
</value>';
```

Don’t forget to change `account` with your Google Analytic ID, and `url`.

This parameters said:

- send tracking in category `video` with action `percentWatched` every 0%, 25%, 50% and 75% of watched video;
- send tracking in category `video` with action `complete` and value 1 every time when video finished;
- send `flashvars.src` as `pageView` when player loaded.

You can find more information about this and other parameters here.

And finally add parameters in `flashvars`:

```
var flashvars = {
    ...

    // Google Analytics settings

    // url to GTrack plugin
    , plugin_ga: "GTrackPlugin.swf"
    // pass parameters to GTrack plugin, which we defined above
    , "ga_http://www.realeyes.com/osmf/plugins/tracking/google": escape(gTrackPluginConfig)

    // you can provide custom page url for tracking as "pageView" instead "flashvars.src"
    // , src_namespace_realeyes: "http://www.realeyes.com/osmf/plugins/tracking/google"
    // , src_realeyes_pageURL: "my_custom_page_url"
    ...
}
```

JavaScript bridge with quality selector

You find complete code of this example in `qualities.html`.

How it works:

- in `flashvars` you add new parameter `javascriptCallbackFunction` with name of function in JavaScript, which will be invoke every time when player send standard events;
- create this function with 3 parameters `(playerId, event, obj)` where:
  - `playerId` - we set this id in `attrs` variable above;
  - `event` - event name;
  - `obj` - data of that event.
- inside this function you first get `onJavaScriptBridgeCreated` event when player ready;
- on this event you subscribe on `isDynamicStreamChange`, `switchingChange`, `autoSwitchChange` and optional on `mediaSizeChange` like this: `player.addEventListener("isDynamicStreamChange", "onDynamicStream");`
- `onDynamicStream` is a JavaScript function which will be invoke from player, in this function you will create quality buttons and change it state (switching, switched);
- in `onDynamicStream` i recommend not do any HTML DOM operation, and `setTimout` with function where HTML DOM operation will be done, it prevent stuck player on switching;
- to change quality stream index you call `player.switchDynamicStreamIndex(index)` where:
  - `player` - you get this variable in `onJavaScriptBridgeCreated` with `document.getElementById(playerId)`;
  - `index` - you get in `onDynamicStream` with `player.getStreamItems()`.

Alternate audio-streams

Limits:

- Your main stream with video+audio is default, and all alternate audio-streams will replace main audio-stream;
- `DEFAULT` and `AUTOSELECT` parameters of `#EXT-X-MEDIA` tag is always `NO`;

Example of M3U8 file with multi-bitrate video+audio streams, and 2 alternate audio-streams:

```
#EXTM3U
#EXT-X-MEDIA:URI="de_vod.m3u8",TYPE=AUDIO,LANGUAGE="de",NAME="German"
#EXT-X-MEDIA:URI="es_vod.m3u8",TYPE=AUDIO,LANGUAGE="es",NAME="Spanish"
#EXT-X-STREAM-INF:BANDWIDTH=481677,RESOLUTION=640x360
0440_vod.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=1308077,RESOLUTION=640x360
1240_vod.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=2650941,RESOLUTION=960x540
2540_vod.m3u8
```

To switch alternate audio-stream through JavaScript see `alternate.html` as example. For this functionality you will need two more JavaScriptBridge events, which i provide to you with minor changes of default SMP player.

Load plugin to custom OSMF player

Be sure you use OSMF 2.0 version, you can get osmf.swc library here.

```
package {

    import flash.display.Sprite;

    import org.osmf.containers.MediaContainer;
    import org.osmf.events.MediaFactoryEvent;
    import org.osmf.media.DefaultMediaFactory;
    import org.osmf.media.MediaElement;
    import org.osmf.media.MediaFactory;
    import org.osmf.media.MediaPlayer;
    import org.osmf.media.URLResource;

    public class OSMFHLSPluginLoadExample extends Sprite {

        private var factory:MediaFactory;

        public function OSMFPluginLoadTest() {
            factory = new DefaultMediaFactory();
            factory.addEventListener(MediaFactoryEvent.PLUGIN_LOAD, onPluginLoad);
            factory.addEventListener(MediaFactoryEvent.PLUGIN_LOAD_ERROR, onPluginLoadError);
            // for local testing
            var resource:URLResource = new URLResource("file:///D:/PATH_TO_PLUGIN_SWF/OSMFHLSPlugin.
                swf");
            // for online
            // var resource:URLResource = new URLResource("OSMFHLSPlugin.swf");
            factory.loadPlugin(resource);
        }

        private function onPluginLoad(event:MediaFactoryEvent):void {
            trace("The plug-in loaded successfully.");
            var resource:URLResource = new URLResource("vod.m3u8");
            var media:MediaElement = factory.createMediaElement(resource);
            mediaPlayer = new MediaPlayer();
            mediaPlayer.media = media;
            mediaPlayer.autoPlay = true;
            var mediaContainer:MediaContainer = new MediaContainer();
            mediaContainer.addMediaElement(media);
            addChild(mediaContainer);
        }

        private function onPluginLoadError(event:MediaFactoryEvent):void {
            trace("The plug-in failed to load.");
        }
    }

}
```

Load plugin to Flex AIR app

CAUTION: Plugin will not work on iOS devices.

Be sure you use OSMF 2.0 version, you can get osmf.swc library here.

```
<?xml version="1.0" encoding="utf-8"?>
<s:WindowedApplication xmlns:fx="http://ns.adobe.com/mxml/2009"
                       xmlns:s="library://ns.adobe.com/flex/spark"
                       width="640" height="360" showStatusBar="false"
                       applicationComplete="onApp()">

    <fx:Script>
        <![CDATA[
            import mx.core.mx_internal;

            import org.osmf.events.MediaFactoryEvent;
            import org.osmf.media.MediaFactory;
            import org.osmf.media.PluginInfoResource;

            private function onApp():void {
                var factory:MediaFactory = player.videoDisplay.mx_internal::mediaFactory;
                factory.addEventListener(MediaFactoryEvent.PLUGIN_LOAD, onPluginLoad);
                factory.loadPlugin(new PluginInfoResource(new OSMFHLSPlugin().pluginInfo));
            }

            private function onPluginLoad(event:MediaFactoryEvent):void {
                player.source = "vod.m3u8";
            }
        ]]>
    </fx:Script>

    <s:VideoPlayer id="player" width="100%" height="100%" autoPlay="true" />

</s:WindowedApplication>
```

FAQ

What is HLS?

HTTP Live Streaming is an HTTP-based media streaming communications protocol implemented by Apple Inc. as part of their QuickTime X and iPhone software systems. more

What is OSMF?

Open Source Media Framework simplifies the development of media players by allowing developers to assemble components to create high-quality, full-featured video playback experiences. This open framework enables development focused on web-based video monetization, with lower costs and faster turnaround. more

What is SMP?

Strobe Media Playback is an OSMF based media player that you can quickly and easily integrate into your website. The compiled SWF and its source code are available for free download. more

Should I do some changes on content server side?

No. If you have infrastructure to deliver video content on iPhone/iPad, then with my plugin you can deliver same content to all browsers with flash plugin installed.

What about analytics?

You can integrate GTrack plugin developed by Realeyes company, and it open source. more

Can I add IFrame with player/plugin on another domain?

No, you can use OSMF HLS Plugin only on domain you were buy license

Troubleshootings

Before investigate any problem, open **Developer Tools** in Chrome, Safari or Firefox and check **Network** tab on errors while download m3u8-playlist or ts-chunks.

I get error when player loaded but video not started yet; I get error while playing video; Player play only low bitrate stream, and not auto-switch up

Check your server response headers, your server on `.swf` and `.ts` files, must not respond with `Transfer-Encoding: chunked`, and must respond with `Content-Length` headers.