Combined Beacon Sample Integration

JW Player

JavaScript, Action Script 3/Flash & Silverlight

Introduction

This document describes steps for integrating Longtail Video JW Player with Nielsen's Combined Beacon library described in the "Combined Beacon JS API Integration" document.

For "JavaScript Native Mode" approach/example, *please refer to page 3 of this document.*

For "Action Script 3 / Flash" approach/example, *please refer to page 11 of this document.*

For "Action Script 3 / Flash" approach/example, *please refer to page 21 of this document*

Table of Contents

JW Player INTEGRATION (JavaScript Native Mode)

Prerequisites and Assumptions

Knowledge of JavaScript and JW Player.

The document “Combined Beacon JS API Integration” is prerequisite reading.

Implementation

***NOTE* “nielsenonlinesupport.com” is not a production environment. Please COPY the example code and CHANGE the src setting in step one below to your local/staging/production environment.**

The following steps are required:

1. Include the JW Player Example Library like this:

```
<script type="text/javascript"
    src="http://www.nielsenonlinesupport.com/clientsupport/jwplayer/ggjw353.js">
</script>
```

ggjwxxx.js (where xxx is the version number) is our Example Library for JW Player – simply including that typically automates data capture for core events like LoadVideo, Play and Pause. Please contact your Nielsen representative for the latest version.

2. Initialize clientid and other parameters like this (these values will be supplied to you by Nielsen):

```
<script type="text/javascript">
        var _nolggGlobalParams = {
        clientid: “ my Nielsen assigned client id ”,
        vcid: “ my Nielsen assigned vc id ”,
        cisuffix: ““,
        sfcode: "Nielsen assigned data node “,
        prod: " Nielsen assigned product code "
};
</script>
```

3. Include and initialize Combined Beacon Javascript library for Native mode:

```
<script type="text/javascript"
    src=" http://secure-XX.imrworldwide.com/novms/js/2/ggcmbXXX.js">


</script>
```

Note 1: “XX” is the Nielsen supplied country code for your region. For example, it is “us” for customers in USA. Please contact your Nielsen representative for the proper region code.
Note 2: “XXX” is the latest version number of the general JS beacon.

4. Complete the initialization by including the following lines:

```
<script>
        var canUseSWF = false;  // forces native mode
        var uid = 0;
        var gg1 = new gg();
        gg1.ggInitialize(_nolggGlobalParams,uid,canUseSWF);

</script>
```

5. Once you have filled in all the required information, it will look similar to the example below:

**Note* The CID & VCID values are examples only. Please the real values that are given by Nielsen representative to you..*

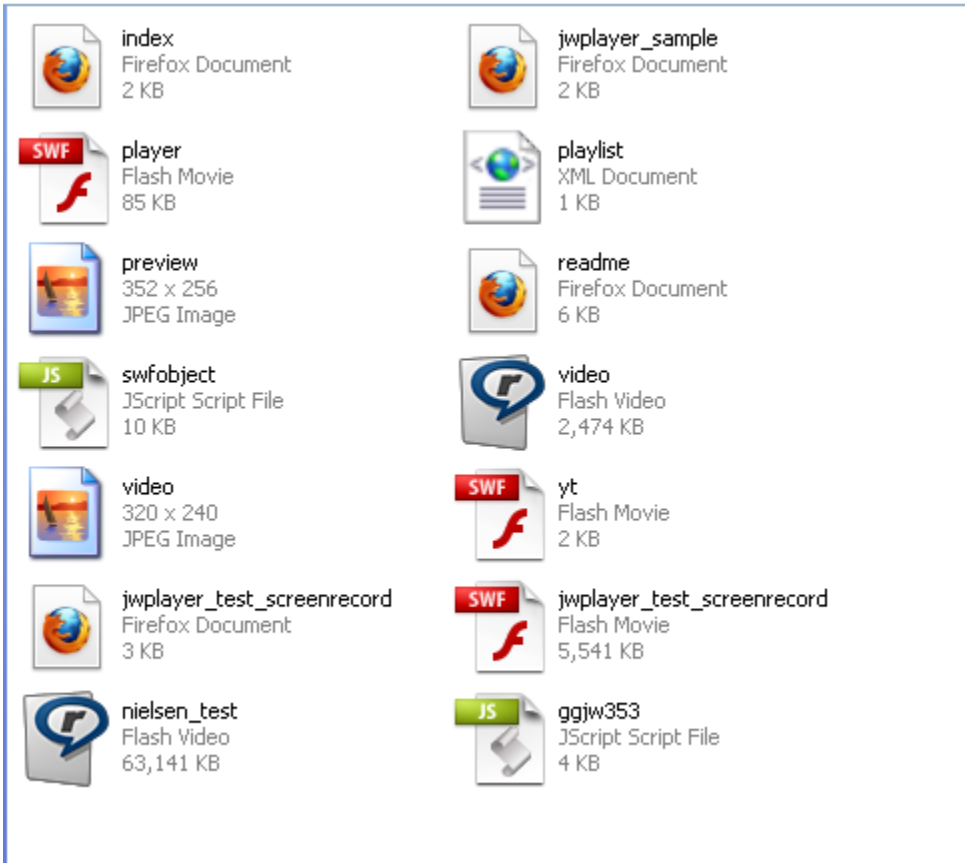
```
<script type="text/JavaScript">

        var _nolggGlobalParams = {
                clientid : "us-123456",
                vcid : "co1",
                sfcode : "us",
                cisuffix : "",
                prod : "sc"
        };

</script>

</head>
<body>

<script type="text/javascript" src="http://secure-it.imrworldwide.com/novms/js/2/ggcmb353.js"></script>
        <script type="text/javascript">
                var canUseSWF = false;
                var uid = 0;
                var gg1 = new gg();
                gg1.ggInitialize(_nolggGlobalParams,uid,canUseSWF);
        </script>
```

Tutorial & Example (Using JavaScript Native Mode)

In this section you will find a basic example of the implementation section detailed previously, plus how to pass video meta-data to the API. Please amend references to http://secure-us.imrworldwide.com to your local Nielsen collection node.

***NOTE* "nielsenonlinesupport.com" is not a production environment. Please COPY the example code and CHANGE the location settings to your local/staging/production environment.**

TUTORIAL FILES TO DOWNLOAD→
http://www.nielsenonlinesupport.com/clientsupport/jwplayer/jwplayer.zip



Please extract the zip file. Files you will be working with are:

jwplayer_sample.htm – your location URL

nielsen_test.flv – Video Footage

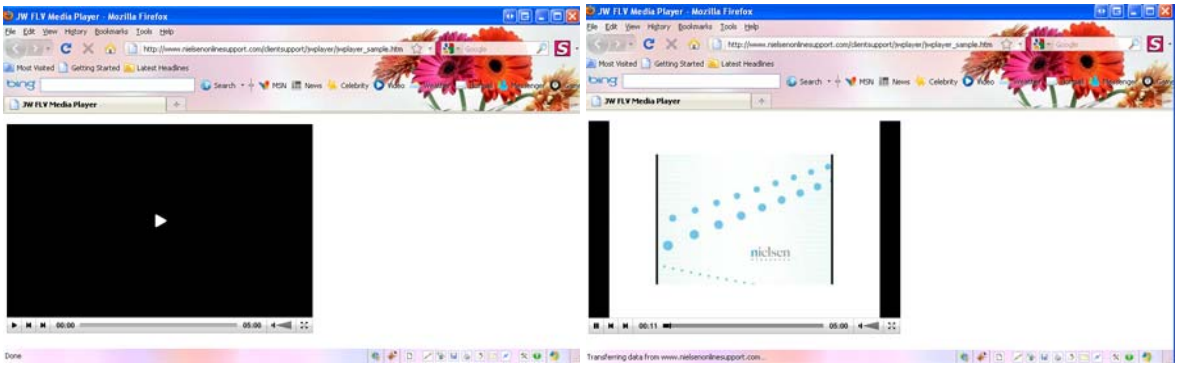
playlist.xml – XML feed of the video

ggjw353.js – JavaScript / Beacon

The following is a live example of JW Player VA Implementation -

Steps:

1. Open your browser and Go to →
http://www.nielsenonlinesupport.com/clientsupport/jwplayer/jwplayer_sample.htm



2. Look under View → Page Source (Ctrl-U)

```
22  <body>
23
24          <script type="text/javascript" src="http://secure-it.imrworldwide.com/novms/js/2/ggcmb353.js"></script>
25          <script type="text/javascript">
26              var canUseSWF = false;
27              var uid = 0;
28              var ggl = new gg();
29              ggl.ggInitialize(_nolggGlobalParams,uid,canUseSWF);
30          </script>
31
32  <p><object classid='clsid:D27CDB6E-AE6D-11cf-96B8-444553540000' width='470' height='320' id='single1' name='single1'>
33  <param name='movie' value='player.swf'>
34  <param name='allowfullscreen' value='true
35  <param name='allowscriptaccess' value='always'>
36  <param name='wmode' value='transparent'>
37  <param name='flashvars' value='file=playlist.xml&playlist=none'>
38  <embed
39  type='application/x-shockwave-flash'
40  id='single2'
41  name='single2'
42  src='http://www.nielsenonlinesupport.com/clientsupport/jwplayer/player.swf'
43  width='470'
44  height='320'
45  bgcolor='undefined'
46  allowscriptaccess='always'
47  allowfullscreen='true'
48  wmode='transparent'
49  flashvars="file=playlist.xml&playlist=none"
50  />
51  </object>
52     <script type="text/javascript" src="ggjw353.js"></script>
53
54  </body>
55  </html>
```

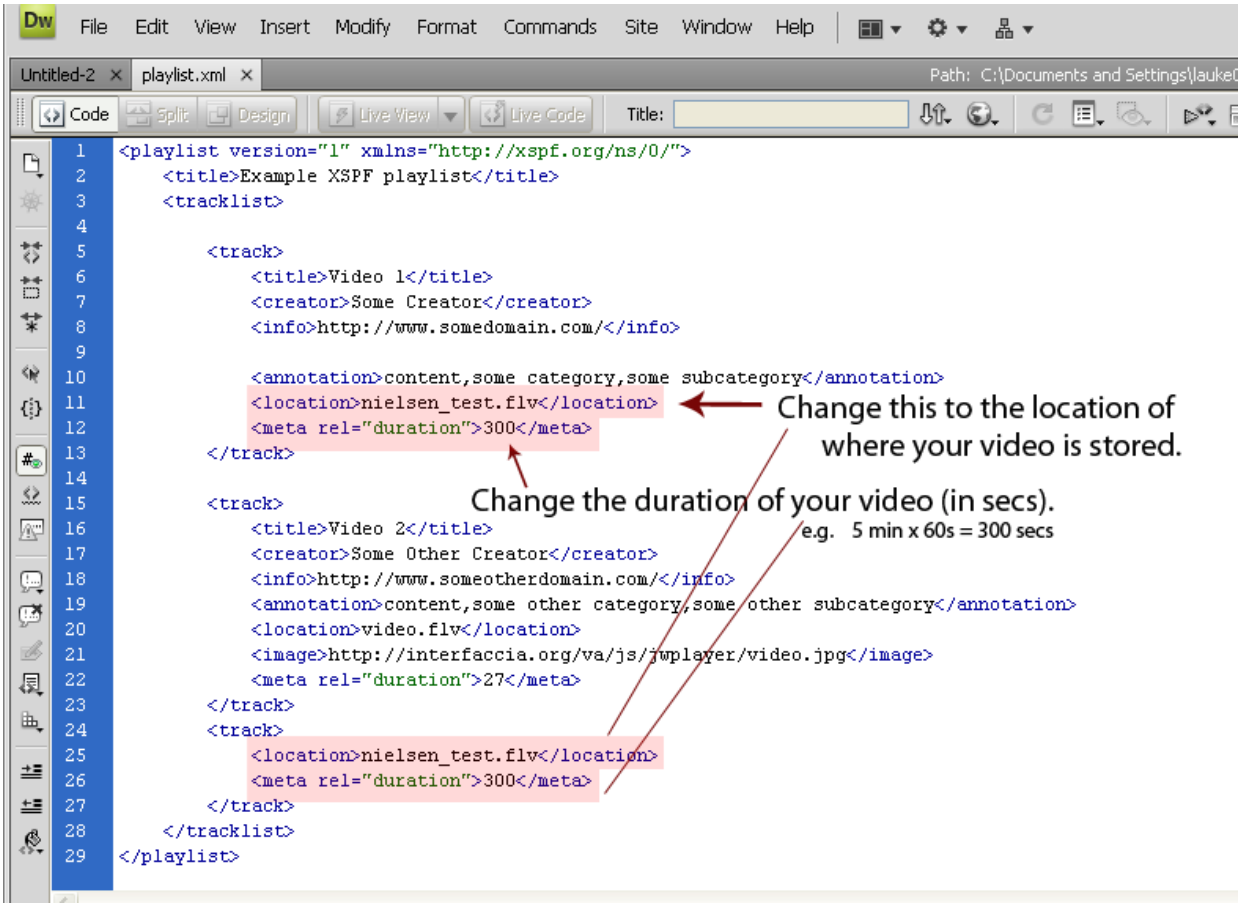
JS Beacon

src="http://secure-it.imrworldwide.com/novms/js/2/ggcmb353.js"></script>

Change this to the location of where your files ('player.swf') are stored.

Location of your JW Player Handler

3. Open → http://www.nielsenonlinesupport.com/clientsupport/jwplayer/playlist.xml




```
<playlist version="1" xmlns="http://xspf.org/ns/0/">
    <title>Example XSPF playlist</title>
    <tracklist>

        <track>
            <title>Video 1</title>
            <creator>Some Creator</creator>
            <info>http://www.somedomain.com/</info>

            <annotation>content,some category,some subcategory</annotation>
            <location>nielsen_test.flv</location>
            <meta rel="duration">300</meta>
        </track>

        <track>
            <title>Video 2</title>
            <creator>Some Other Creator</creator>
            <info>http://www.someotherdomain.com/</info>
            <annotation>content,some other category,some other subcategory</annotation>
            <location>video.flv</location>
            <image>http://interfaccia.org/va/js/jwplayer/video.jpg</image>
            <meta rel="duration">27</meta>
        </track>
        <track>
            <location>nielsen_test.flv</location>
            <meta rel="duration">300</meta>
        </track>
    </tracklist>
</playlist>
```

You will need to change the location of video file and duration of your video (in secs) under playlist.xml, in order for your video to be linked and tracked properly.

"Annotation" contains optional information about the clip. This is composed by three values separated by a comma, namely:
- vidtype
- category
- subcategory

Please refer to the Javascript API document for further information about metadata.

If you want to add more videos to your playlist, you may go to the following links for additional information:

JW API Example

http://developer.longtailvideo.com/contributors/nyboe/JW_API_xmpl_6-2-0-0.html

http://developer.longtailvideo.com/contributors/nyboe/index.html

JW Player Wizard

http://www.longtailvideo.com/support/jw-player-setup-wizard

4. Open → http://www.nielsenonlinesupport.com/clientsupport/jwplayer/ggjw353.js

```
function timeTracker(obj) {
    tempTime = obj.position;
    if (tmp49int > 1) {
        if ((tempTime > (tmp49intTime * tmp49intMultiplier))){
            ggl.ggPM(49, Math.round(tempTime));
            tmp49intMultiplier+=1;
        };
    };
    if (tempSeekBoolean == true){
        if (tempSeekto > tempTime){
            ggl.ggPM(5, Math.round(tempTime));
        }
        tempSeekto = null;
        tempSeekBoolean = false;
    };
};

function playTracker(obj) {
    if (obj.newstate == "PLAYING" && obj.oldstate == "PAUSED") {
        ggl.ggPM(5, Math.round(tempTime));
    };

    if ((obj.newstate == "PLAYING") && (obj.oldstate != "PAUSED") && (tempBuffBoolean != true))
    {
        ggl.ggPM(15, plst[plstItem].file, tempVidtype, tmpXMLinfo);
//      ggl.ggPM(5, Math.round(tempTime));
    };
    tempBuffBoolean = false;
    if (obj.newstate == "BUFFERING" && obj.oldstate == "PLAYING") {
        tempBuffBoolean = true;
        };
    if (obj.newstate == "PAUSED") {
        ggl.ggPM(6, Math.round(tempTime));
        };
    if (obj.newstate == "IDLE") {
        ggl.ggPM(7, Math.round(tempTime));
```

You can see that some of the 'Player Events' are being tracked by function timeTracker and playTracker.

For example: Event 49 'Playing', Event 5 'Play Video', Event 15 'Load and Play Video', Event 6 'Pause Video', Event 7 'Stop', etc.

(*Continue next page)*

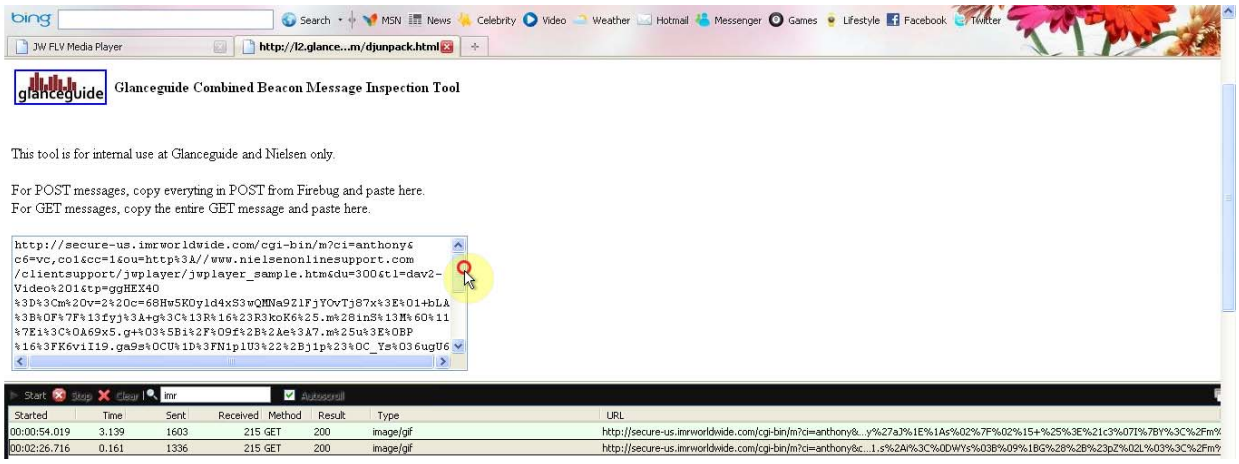
```
83
84  function seekTracker(obj) {
85      tempSeekto = Math.round(obj.position);
86      ggl.ggPM(8, Math.round(tempTime), tempSeekto);
87      if (tmp49int > 1) { tmp49intMultiplier = Math.round(tempSeekto / tmp49int)};
88      tempSeekBoolean = true;
89  };
90
91  function muteTracker(obj) {
92      ggl.ggPM(9, obj.state);
93  };
94
95  function fscreenTracker(obj) {
96      ggl.ggPM(10, obj.fullscreen);
97  };
98  function volumeTracker(obj){
99      ggl.ggPM(11, obj.percentage);
100 };
101
102 function errTracker(obj){
103     alert(obj.message);
104 };
105
106 function playerReady(obj) {
107
108     ggl.ggPM(51, "http://www.test.com");
109     player = document.getElementById(obj['id']);
110     player.addModelListener("STATE","playTracker");
111     player.addModelListener("TIME","timeTracker");
112     player.addViewListener("SEEK","seekTracker");
113     player.addControllerListener("RESIZE","fscreenTracker");
114     player.addControllerListener("ITEM","plstTracker");
115     player.addControllerListener("VOLUME","volumeTracker");
116     player.addViewListener("MUTE","muteTracker");
117 //  player.addModelListener("ERROR","errTracker"); // DEBUG ONLY!
118     plst = player.getPlaylist();
119 };
```

Function seekTracker is tracking events: Event 8 ‘Seek’, Event 9 ‘Mute’, Event 10 ‘Full screen’, Event 11 ‘setVolume’, and Event 51 ‘setPageURL’ (Ex. www.test.com).

To view the full list of all the ‘Player Events’, you may refer this to “Combined Beacon Media Player Integration, JavaScript API Guide.”

5. Upload your own video footage. In this example, nielsen_test.flv is the video footage, which we will be testing with.

6. Upload and test your player. After you've completed all the steps, you may now test it on your FireFox browser with tools such as HTTPFOX. Click the following link to see how testing can be done:



http://www.nielsenonlinesupport.com/clientsupport/jwplayer/jwplayer_test_screenrecord.html

For full tutorial on how to use the test tool, you may refer this to "Combined Beacon Media Player Integration, JavaScript API Guide."

JW Player INTEGRATION (Action Script 3/Flash)

Prerequisites and Assumptions

Knowledge of Action Script 3.0, Flash and JW Player.

The document "Combined Beacon Media Player Integration, Flash Players, Action Script 3 Guide" is prerequisite reading.

Implementation

***NOTE* "nielsenonlinesupport.com" is not a production environment. Please COPY the example code and CHANGE the src setting in step one below to your local/staging/production environment.**

Copy **ggCom.as**, **ggNetStream.as** and **ggSoundTransform.as** files (supplied by Nielsen) to the same directory as the .FLA file for your media player.

Setting the Brand/Channel Id (ClientID / VCID) & Initialize Beacon

Open **ggCom.as** in a Text Editor. Find *_nolggGlobalParams* and change initial value of *clientid* to Nielsen-supplied *clientid* for your company.

e.g. clientid: "us-12345". Also change initial value of *vcid* to Nielsen-supplied VideoCensus id for the lowest level in the Marketview hierarchy. E.g. vcid: "c01" case-sensitive.

```
public var _nolggGlobalParams:Object = {

        clientid:"us-123456",                    // Required; Nielsen assigned client ID

        vcid:    "c01",                          // Required; Nielsen assigned vcid

        msgint: ""                               // Optional. to specify additional messages per stream. by default,
start and end streams get a msg each

  };
```

DO NOT edit other variables and functions in ggCom.as, which include initializing Combined Beacon for Flash:

```
GGSWFADDRESS = "http://secure-" + _nolggGlobalParams.sfcode +
".imrworldwide.com/novms/gn/3/ggce354.swf";
```

Example: http://secure-us.imrworldwide.com/novms/gn/3/ggce354.swf

Website Configuration Required For Flash Based Players

In order for the beacon to detect the web page state change, it requires that the flash beacon have outbound URL access to the page that it is loaded into. This is set with the **AllowScriptAccess** command.

For example:

```
<script type="text/javascript">
  var vp = new SWFObject("player.swf","videoplayer","470","320","9");
  vp.addParam("allowfullscreen","true");
  vp.addParam("allowscriptaccess","always");
  vp.addParam("wmode","opaque");

vp.addParam("flashvars","file=http://www.nielsenonlinesupport.com/clientsuppo
rt/jwplayer_flash/nielsen_test.flv&title=Nielsen Tutorial - Flash
Implementation for JW Player
4.3v&description=<vidtype>content</vidtype><category>Training
Video/category><subcategory></subcategory>");

  vp.write("player");
</script>
```

Tutorial & Example (Using Action Script 3 / Flash)

In this section you will find a basic flash example of the implementation section detailed previously, plus how to pass video meta-data to the API.

Please amend references to http://secure-us.imrworldwide.com to your local Nielsen collection node.

***NOTE* "nielsenonlinesupport.com" is not a production environment. Please COPY the example code and CHANGE the location settings to your local/staging/production environment.**

This example is built specifically for **JW player 4.3 version**. For other versions of JW player, please download the latest player from "Longtail Video." You may also want to download the original player and compare to the one in this tutorial to help better understand the code changes.

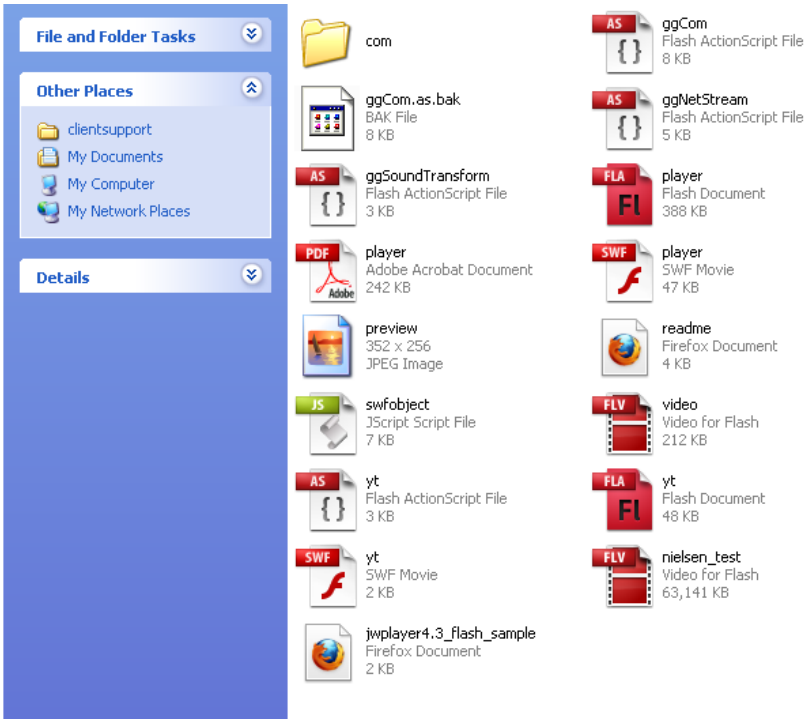
Go to: http://www.longtailvideo.com/players/jw-player-4-for-flash

If you are new to JW player, you can also use the **JW Player Wizard** to get started.

http://www.longtailvideo.com/support/jw-player-setup-wizard

TUTORIAL FILES TO DOWNLOAD→

http://www.nielsenonlinesupport.com/clientsupport/jwplayer_flash/jwplayer_flash.zip



Please extract the zip file. Files you will be working with are:

jwplayer4.3_flash_sample.html – your location URL

nielsen_test.flv – Video Footage

ggCom.as, ggNetStream.as, ggSoundTransform – Action Script to place the beacon

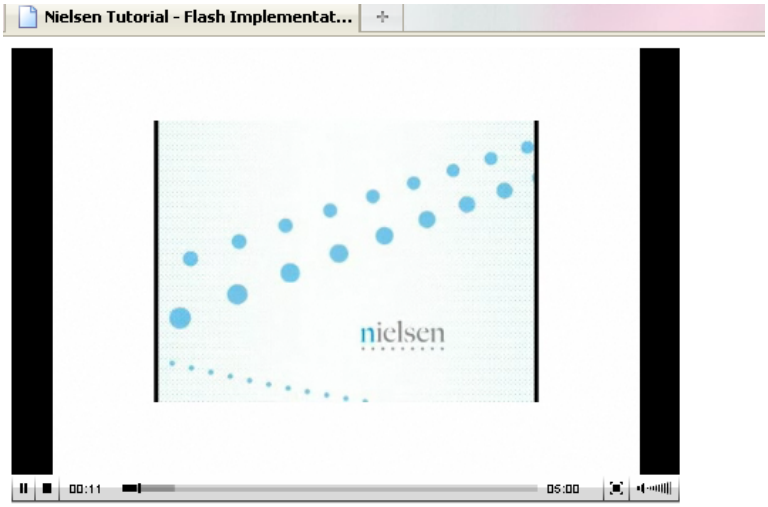
com/jeroenwjering/models/VideoModel.as – Tag Implement & Metadata Treatment

player.fla – Flash file to export your new player (player.swf)

The following is a live example of JW Player VA Implementation using Flash/Action Script 3.0-

Steps:

1. Open your Firefox browser and Go to → http://www.nielsenonlinesupport.com/clientsupport/jwplayer_flash/jwplayer4.3_flash_sample.html



Nielsen Tutorial - Flash Implementation for JW Player 4.3v

2. Look under View → Page Source (Ctrl-U)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Nielsen Tutorial - Flash Implementation for JW Player 4.3v</title>
</head>

<body>

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

<div id="player">This text will be replaced</div>

<script type="text/javascript">
  var vp = new SWFObject("player.swf","videoplayer","470","320","9");
  vp.addParam("allowfullscreen","true");
  vp.addParam("allowscriptaccess","always");
  vp.addParam("wmode","opaque");
  vp.addParam("flashvars","file=http://www.nielsenonlinesupport.com/clientsupport/jwplayer_flash/nielsen_test.flv&title=Nielsen Tutorial - Flash
Implementation for JW Player 4.3v&description=<vidtype>content</vidtype><category>Training Video/category><subcategory></subcategory>");

  vp.write("player");
</script>

Nielsen Tutorial - Flash Implementation for JW Player 4.3v




</object>
</body>
</html>
```

Change it to your video link file, setting your content and subcatergory.

3. Open the action script file "**ggCom.as**", which is under the zip file that you have previously downloaded. Change the clientid and vcid value, which you were given from your Nielsen Technical Representative.

```
27      // ********************** Start Customer Editable variables and functions. DO NOT EDIT ABOVE T
28      public var _nolggGlobalParams:Object = {
29         clientid:"anthony",                          // Required; Nielsen assigned client ID
30         vcid:    "c01",                            // Required; Nielsen assigned vcid
31         msgint: ""                                   // Optional, to specify additional messages per s
32      };
33
34           // set to true is Javascript interface is needed for calls to gg; if enableJSInterface is
35           // make sure that "allowScriptAccess" parameter is set to "always". Runtime error will ha
36           private var enableJSinterface: Boolean = false;
37           public function getUserID( ) : String
38           {
39               return null;   // optionally set userid here (max 32 chars)
40           }
41           // Additional custom video information can be sent to Glance Guide by using this method.
42           // NOTE: Use the following xml syntax: <pl>value</pl> .... <pn>value</pn>
43           function setVideoInfo( videoInfo ) : void
44           {
45               PM( "videoInfo", videoInfo );
46           };
47
48   //*
49   // ********************** End Customer Editable variables and functions. DO NOT EDIT BELOW THIS L
50   //*
51      //additional config parameters (set by NOL/GG)
52      function setNOLconfigparams() {
53        _nolggGlobalParams.sfcode = "us";
54        _nolggGlobalParams.cisuffix = "";
55        _nolggGlobalParams.prod = "sc";
56      }
57
```

Enter the client id & vcid, for example: us-123456, c01

4. Open the action script file “**ggNetStream.as**”, this file include the list of events being tracked under the beacon.

```
/**
* ggNetStream class extends NetStream (Actionscript 3)
* Intercepts the video stream and records important metrics before returning control back to the NetStre
*
* Copyright (c) 2008 Glanceguide, Inc.  All rights reserved.
*    http://glanceguide.com
* .  Decompiling, reverse engineering, copying or unauthorized redistribution is prohibited.
**/

package
{
    import flash.net.NetStream;
    import flash.net.NetConnection;
    import flash.events.NetStatusEvent;
    import flash.utils.setInterval;
    import flash.media.SoundTransform;

    public class ggNetStream extends NetStream
    {
        public var copyright: String = "Copyright (c) 2008 Glanceguide, Inc";
        private var _ggClient:ggClient;
        private var _flvName:String;
        private var _videoInfo:String;
        public function ggNetStream( nc:NetConnection )
        {
            this.addEventListener( NetStatusEvent.NET_STATUS, netStatusHandler );
            setInterval( function( stream:Object ){ ggCom.getInstance( ).PM( "E49", stream.time ); }, 20
            super( nc );
        }
        public function set ggSoundTransform( value:Object ) : void
        {
            this.soundTransform = value.getSoundTransform( );
        }
        // videoinfo for the subsequent Play call; title, vidtype etc.
        public function setVideoInfo(vi:String)
        {
      _videoInfo = vi;
        }

```

(Continue next page)

```
        {
            if ( arguments[ 0 ] != null )
                _flvName = arguments[ 0 ];

            if ( _ggClient == null || _ggClient != this.client )
                this.client = _ggClient = new ggClient( this.client, _flvName, _videoInfo, this );

            ggCom.getInstance( ).PM( "E4", _flvName );
            ggCom.getInstance( ).PM( "E5", 0 );

            super.play( arguments[0] );
        }
        public override function pause( ) : void
        {
            ggCom.getInstance( ).PM( "E6", this.time );

            super.pause( );
        }
        public override function seek( offset:Number ) : void
        {
            ggCom.getInstance( ).PM( "E8", offset, this.time );

            super.seek( offset );
        }
        public override function togglePause( ) : void
        {
            ggCom.getInstance( ).PM( "togglePause", this.time );

            super.togglePause( );
        }
        public override function resume( ) : void
        {
            ggCom.getInstance( ).PM( "E5", null );

            super.resume( );
        }
        public override function close( ) : void
        {
            ggCom.getInstance( ).PM( "close", this.time );
            super.close( );
```

You can see that some of the ‘Player Events’ are being tracked by different functions highlighted above.

For example: Event 5 ‘Play Video’, Event 6 ‘Pause Video’, Event 8 ‘Seek’, etc.

To view the full list of all the ‘Player Events’, you may refer this to “Combined Beacon Media Player Integration, Flash Players, Action Script 3 Guide.”

5. The action script file "**ggSoundTransform.as**", is a script that tracks important metrics of sound change events that happen during the audience viewing of the video. This is optional for the implementation process.

```
13      import flash.media.SoundTransform;
14
15      public class ggSoundTransform
16      {
17          private var _soundTransform:SoundTransform;
18
19          // Constructor
20          public function ggSoundTransform( vol:Number = 1, panning:Number = 0 )
21          {
22              // Create SoundTransform
23              _soundTransform = new SoundTransform( vol, panning );
24          }
25
26          // Return the SoundTransfomr
27          public function getSoundTransform( )
28          {
29              return _soundTransform;
30          }
31
32          // Volume getters and setters
33          public function set volume( value:Number ) : void
34          {
35              // Send sound event to GlanceGuide
36              ggCom.getInstance( ).PM( 11, value * 100 );
37
38              // Set sound in SoundTransform
39              _soundTransform.volume = value;
40          }
41
42          public function get volume( ) : Number
43          {
44              return _soundTransform.volume;
45          }
46
47          // Remaining necessary wrapper methods
48          public function set leftToLeft( value:Number ) : void
49          {
50              _soundTransform.leftToLeft = value;
51          }
52
```

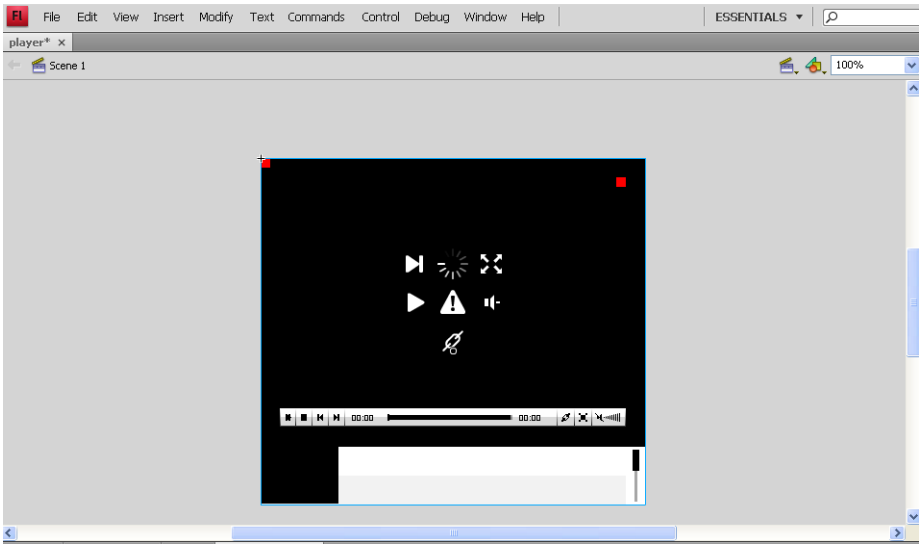
6. Open the action script file "**VideoModel.as"** under the folder "com/jeroenwjering/models/". This is file where you going to place and call up your tag, along metadata treatment.

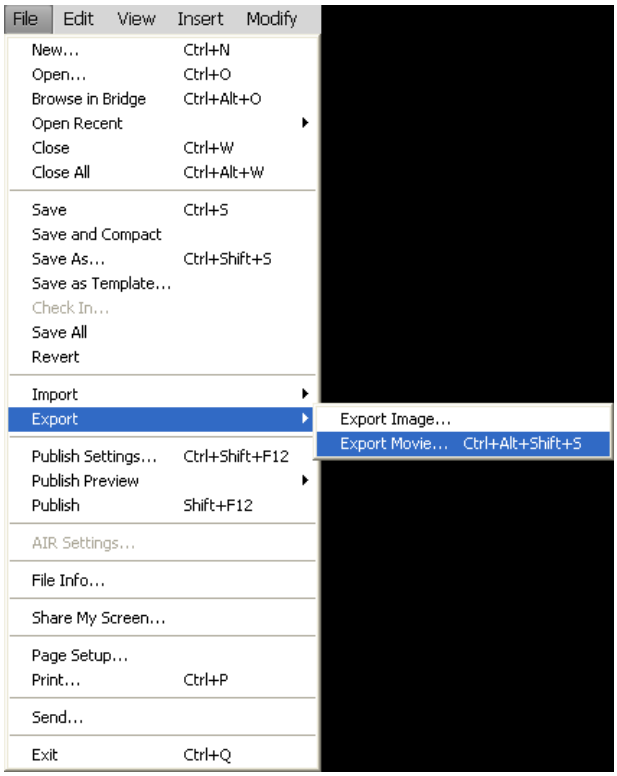
```
/**
* Wrapper for playback of progressively downloaded video.
**/
package com.jeroenwijering.models {


import com.jeroenwijering.events.*;
import com.jeroenwijering.models.ModelInterface;
import com.jeroenwijering.player.Model;
import com.jeroenwijering.utils.NetClient;
import flash.events.*;
import flash.display.DisplayObject;
import flash.media.SoundTransform;
import flash.media.Video;
import flash.net.*;
import flash.utils.clearInterval;
import flash.utils.setInterval;


public class VideoModel implements ModelInterface {


    /** reference to the model. **/
    private var model:Model;
    /** Video object to be instantiated. **/
    private var video:Video;
    /** NetConnection object for setup of the video stream. **/
    private var connection:NetConnection;
    /** NetStream instance that handles the stream IO. **/
    private var stream:ggNetStream;
    /** Sound control object. **/
    private var transform:SoundTransform;
    /** Interval ID for the time. **/
    private var timeinterval:Number;
    /** Interval ID for the loading. **/
    private var loadinterval:Number;
    /** Metadata received switch. **/
    private var metadata:Boolean;
```

```
    /** Destroy the video. **/
    public function stop():void {
        stream.pause();
        if(stream.bytesLoaded != stream.bytesTotal) {
            stream.close();
            ggCom.getInstance().PM (7,timeinterval);
        }
        metadata = false;
        ggCom.getInstance().PM (7,timeinterval);
        clearInterval(loadinterval);
        clearInterval(timeinterval);
    };
```

'Stop' event tag added.

7. After you are done with all your changes, you will need to re-export your player (player.swf). Open player.fla (Flash file) with your flash program.





9. Upload all your latest file and test your player. You may refer to *page 9* of this document on how to test your player.

JW Player INTEGRATION (Silverlight)

Prerequisites and Assumptions

Knowledge of JavaScript and Silverlight Player.

The document “Combined Beacon JS API Integration” is prerequisite reading.

Implementation

***NOTE* “nielsenonlinesupport.com” is not a production environment. Please COPY the example code and CHANGE the src setting in step one below to your local/staging/production environment.**

The following steps are required:

1. Include the JW Player Bridge like this:

```
<script type="text/javascript"
    src=
“http://www.nielsenonlinesupport.com/clientsupport/player_example/jwplayer_silverlight/ggsl.js">
</script>
```

ggsl.js is the bridge object that listens to the player, then formats the proper message before sending it to the beacon. For any beacon implementations, it requires to contain the player, the beacon, and the bridge (ggsl.js).

2. Initialize clientid and other parameters like this (these values will be supplied to you by Nielsen):

```
<script type="text/javascript">
        var _nolggGlobalParams = {
        clientid: “ my Nielsen assigned client id ”,
        vcid: “ my Nielsen assigned vc id ”,
        cisuffix: ““,
        sfcode: "Nielsen assigned data node “,
        prod: " Nielsen assigned product code "
};
</script>
```

3. Include and initialize Combined Beacon JavaScript library for Native mode:

```
<script type="text/javascript"
    src=" http://secure-XX.imrworldwide.com/novms/js/2/ggcmbXXX.js">


</script>
```

Note 1: “XX” is the Nielsen supplied country code for your region. For example, it is “us” for customers in USA. Please contact your Nielsen representative for the proper region code.
Note 2: “XXX” is the latest version number of the general JS beacon.

4. Complete the initialization by including the following lines:

```
<script>
        var canUseSWF = false;  // forces native mode
        var uid = 0;
        var gg1 = new gg();
        gg1.ggInitialize(_nolggGlobalParams,uid,canUseSWF);

</script>
```

5. Once you have filled in all the required information, it will look similar to the example below:

**Note* The CID & VCID values are examples only. Please the real values that are given by Nielsen representative to you..*

```
<script type="text/JavaScript">

        var _nolggGlobalParams = {
                clientid : "us-123456",
                vcid : "co1",
                sfcode : "us",
                cisuffix : "",
                prod : "sc"
        };

</script>

</head>
<body>

<script type="text/javascript" src="http://secure-it.imrworldwide.com/novms/js/2/ggcmb353.js"></script>
        <script type="text/javascript">
                var canUseSWF = false;
                var uid = 0;
                var gg1 = new gg();
                gg1.ggInitialize(_nolggGlobalParams,uid,canUseSWF);
        </script>
```

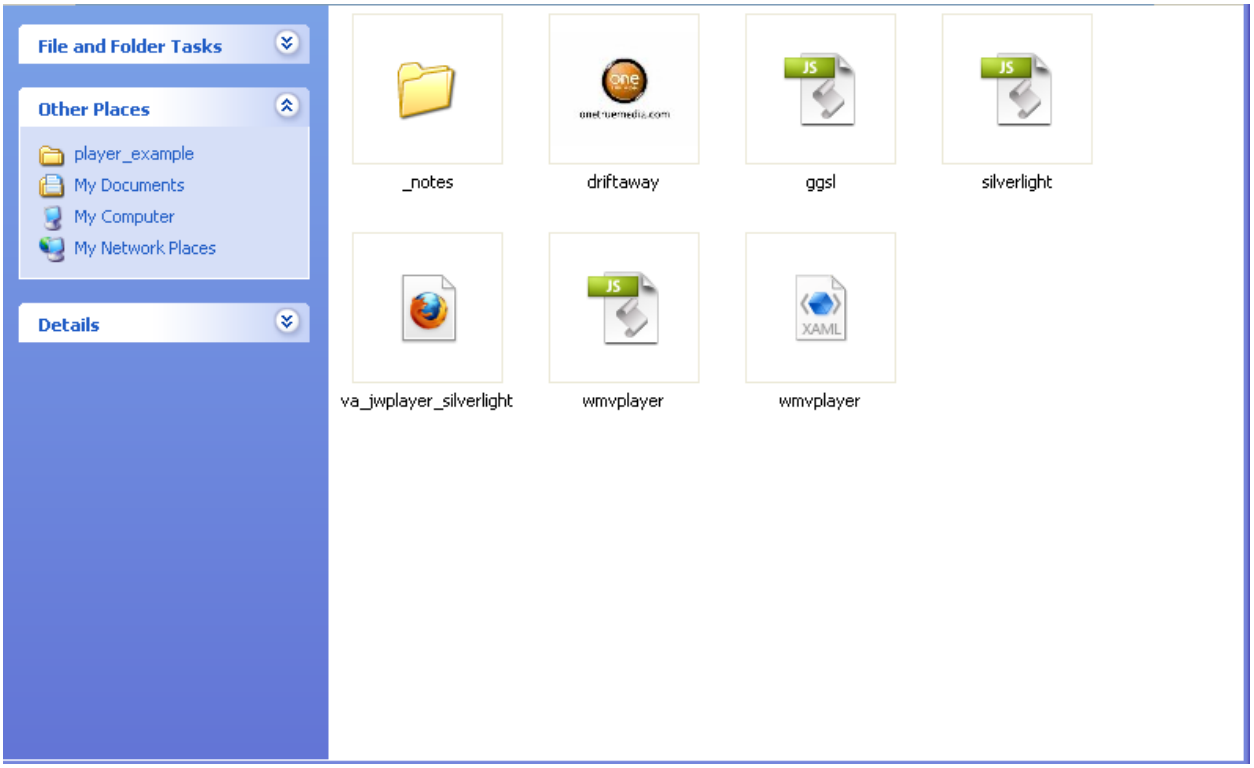
Tutorial & Example (with Silverlight)

In this section you will find a basic example of the JW player implementation using Silverlight. Please amend references to http://secure-us.imrworldwide.com to your local Nielsen collection node.

***NOTE* "nielsenonlinesupport.com" is not a production environment. Please COPY the example code and CHANGE the location settings to your local/staging/production environment.**

TUTORIAL FILES TO DOWNLOAD→

http://www.nielsenonlinesupport.com/clientsupport/player_example/jwplayer_silverlight/jwplayer_silverlight.zip



Please extract the zip file. Files you will be working with are:

silverlight.html – your location URL

driftaway.wmv. – Video Footage

ggsl.js – Bridge Object

silverlight.js, wmvplayer.js, wmvplayer.xaml – Original player files from LongTail Video

If you would like to learn more about JW Player with Silverlight, you can visit:

http://www.longtailvideo.com/players/jw-wmv-player/

The following is a live example of JW Player VA Implementation using Silverlight-

Steps:

1. Open your Firefox browser and Go to → http://www.nielsenonlinesupport.com/clientsupport/player_example/jwplayer_silverlight/va_jwplayer_silverlight.html

JW Player VA Implementation using Silverlight

2. Look under View → Page Source (Ctrl-U)

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.c
2  <html xmlns="http://www.w3.org/1999/xhtml">
3  <head>
4  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5  <title>Nielsen JW Player with Silverlight</title>
6
7  <script type="text/JavaScript">
8
9                  var _nolggGlobalParams = {
10                     clientid : "anthony",
11                     vcid : "col",
12                     sfcode : "us",
13                     cisuffix : "",
14                     prod : "sc"
15                 };
16
17         </script>
18 </head>
19 <body>
20
```

Enter your clientid & vcid

```
21  <script type="text/javascript" src="http://secure-it.imrworldwide.com/novms/js/2/ggcmb353.js"></script>
22
23  <script type="text/javascript">
24  var canUseSWF = false; // forces native mode
25  var uid = 0;
26  var ggl = new gg();
27  ggl.ggInitialize(_nolggGlobalParams,uid,canUseSWF);
28
29  </script>
30  <script type="text/javascript" src="ggsl.js"></script>
31  <script type="text/javascript" src="silverlight.js"></script>
32  <script type="text/javascript" src="wmvplayer.js"></script>
33
34
35  <div id="myplayer">the player will be placed here</div>
36  <p>  </p>
37  <p>JW Player VA Implementation using Silverlight</p>
38  <script type="text/javascript">
39          var elm = document.getElementById("myplayer");
40          var src = 'http://www.nielsenonlinesupport.com/clientsupport/player_example/jwplayer_silverlight/wmvplayer.xaml';
41          var cfg = {
42              file:'http://www.nielsenonlinesupport.com/clientsupport/player_example/jwplayer_silverlight/driftaway.wmv',
43              width:'480',
44              height:'320'
45          };
46          var ply = new jeroenwijering.Player(elm,src,cfg);
47      </script>
```

Nielsen VA Beacon

Bridge Object

Change it to your file location

Change it to your file location of your video clip

2. Open the JavaScript file "**ggsl.js**", this is the bridge object file, which include the list of events being tracked under the beacon. You **do not need** to make any edits for this file, unless you would like to add in additional events being tracked.

```
46          }
47          switch (media.CurrentState) {
48              case 'Stopped':
49                  if(!this.movieStatus.stopped) {
50                      this.movieStatus.stopped = true;
51                      this.gg.ggPM(7,send_pos);
52                      if(this.timer) {
53                          clearInterval(this.timer);
54                      }
55                      if(this.playStateTimer) {
56                          clearInterval(this.playStateTimer);
57                      }
58                  }
59              break;
60              case 'Paused':
61                  if(this.prevState == "Playing") {
62                      this.gg.ggPM(6,send_pos);
63                      if(this.timer) {
64                          clearInterval(this.timer);
65                      }
66                      if(this.playStateTimer) {
67                          clearInterval(this.playStateTimer);
68                      }
69                  }
70              break;
```

Event: "Stop"

Event: "Pause"

```
85                      this.type = type;
86                      var l_metaData = "";
87                      if(attributes.getItemByName("category")) {
88                          l_metaData += '<category>'+attributes.getItemByName("category").va
89                      }
90                      if(attributes.getItemByName("subcategory")) {
91                          l_metaData += '<subcategory>'+attributes.getItemByName("subcategor
92                      }
93                      this.prevState = 0;
94                      this.gg.ggPM(3, this.cur_movie_url, type, "<title>"+this.cur_movie+"</
    this.duration)+"</length>"+l_metaData);
95                      this.movieStatus.played = true;
96                  }
97                  this.movieStatus.stopped = false;
98                  this.gg.ggPM(5,send_pos);
99
100                 if(this.timer) {
101                     clearInterval(this.timer);
102                 }
103                 if(this.playStateTimer) {
104                     clearInterval(this.playStateTimer);
105                 }
106                 var that = this;
107                 this.timer = setInterval(function() {that.updatePos(that.timer)}, 250);
108                 this.playStateTimer = setInterval(function() {that.updatePos(that.playState
109             break;
```

Event: "Load"

Event: "Play"

```
216         this.movieStatus.loaded = false;
217     }
218 }
219
220 ggSL.prototype.updatePos = function(caller) {
221     this.cur_position = parseFloat(this.media.Position.Seconds.toFixed(1)) > parseFlo
    toFixed(1)) : parseFloat(this.cur_position);
222     if(caller == this.timer) {
223         if(this.cur_mute != this.media.IsMuted) {
224             this.cur_mute = this.media.IsMuted;
225             var curMute = this.cur_mute ? 1:0;
226             this.gg.ggPM(9, curMute);
227         }
228         if(this.cur_volume != this.media.Volume) {
229             this.cur_volume = this.media.Volume;
230             this.gg.ggPM(11, String(parseInt(this.cur_volume*100)));
231         }
232     } else {
233         //this.gg.ggPM(49, this.cur_position);
234     }
235 }
236
237 ggSL.prototype.OnFullScreenChange = function(sender, args) {
238         this.fullScreen = sender.getHost().content.fullScreen;
239         this.gg.ggPM(10, String(this.fullScreen));
240 }
```

Event: "Mute"

Event: setVolume

Event: Full Screen

3. Upload all your latest files and test your player. You may refer to *page 9* of this document on how to test your player.