# Microsoft® Smooth Streaming Plugin for Open Source Media Framework

Updated: October 16, 2014

# Contents

# Overview

Microsoft Smooth Streaming plugin for OSMF 2.0 (SS for OSMF) extends the default capabilities of "Open Source Media Framework 2.0" and adds Microsoft Smooth Streaming content playback to new and existing OSMF players. The plugin also adds Smooth Streaming playback capabilities to Strobe Media Playback (SMP).

SS for OSMF includes two versions of plugin:

- Static Smooth Streaming plugin for OSMF (.swc)
- Dynamic Smooth Streaming plugin for OSMF (.swf)

This document assumes that the reader has a general working knowledge of OSMF and OSMF plug-ins. For more information about OSMF, please see the documentation on the official OSMF site.

## Smooth Streaming plugin for OSMF 2.0

The plugin supports loading and playback of on-demand Smooth Streaming content with the following features:

- On-demand Smooth Streaming playback (Play, Pause, Seek, Stop) with support for AES-128 bit encrypted content
- Live Smooth Streaming and DVR playback (Play, Pause, Seek, DVR, Go-to-live) with support for AES-128 bit encrypted content
- Windows Azure Media Services on-demand MPEG-DASH (Play, Pause, Seek, Stop)
- Support for video codecs – H.264
- Support for Audio codecs – AAC
- Multiple audio language switching with OSMF built-in APIs
- Playback quality selection with OSMF built-in APIs
- Multi-timescale support
- Adobe® Flash® Player 11.4 or higher
- This version only supports OSMF 2.0

The following are unsupported features:

- VC-1 and WMA codec
- Digital Dolby Plus (EC3) codec
- Content protection (PlayReady)
- Text and Sparse Tracks
- Trickplay (slow motion, fast-forward, and rewind)
- Live DASH playback
- DASH playback with AES-128 bit encrypted content
- Audio only playback

The following is a list of known issues:

- Multiple Smooth Streaming content playback on single page might cause issues. This is a known issue with OSMF.
- Playback of Stage video might cause problems and no video on some machines. As a workaround you can disable hardware acceleration or Stage video.
- Content where there is a discrepancy between start and/or end times between audio and video may result in undesired behavior.
- Only DASH content streamed through Azure Media Services is supported; all other implementations of DASH may result in undesired behavior.

## Loading Plugin

OSMF plugins can be loaded statically (at compile time) or dynamically (at run-time). Smooth Streaming plugin for OSMF includes both dynamic and static versions.

- Static loading: To load statically, a static library (SWC) file is required. Static plugins are added as a reference to the projects and merge inside the final output file at the compile time.
- Dynamic loading: To load dynamically, a precompiled (SWF) file is required. Dynamic plugins are loaded in the runtime and not included in the project output. (Compiled output) Dynamic plugins can be loaded using HTTP and FILE protocols.

For more information, see the official OSMF plugin page here.

## SS for OSMF static loading

The code snippet below shows how to load SS plugin for OSMF statically and play a basic video using OSMF MediaFactory. Before including SS OSMF code, please ensure that the project reference includes "MSAdaptiveStreamingPlugin-v1.x.x-osmf2.0.swc" static plugin.

```
package
{
        // import related OSMF and Flash Runtime classes.

        import com.microsoft.azure.media.AdaptiveStreamingPluginInfo;

        import flash.display.*;
        import org.osmf.media.*;
        import org.osmf.containers.MediaContainer;
        import org.osmf.events.MediaErrorEvent;
        import org.osmf.events.MediaFactoryEvent;
        import org.osmf.events.MediaPlayerStateChangeEvent;
        import org.osmf.layout.*;

        //Sets the size of the SWF

        [SWF(width="1024", height="768", backgroundColor='#405050', frameRate="25")]
        public class TestPlayer extends Sprite
        {
                public var _container:MediaContainer;
                public var _mediaFactory:DefaultMediaFactory;
                private var _mediaPlayerSprite:MediaPlayerSprite;


                public function TestPlayer( )
                {
                        stage.quality = StageQuality.HIGH;

                        initMediaPlayer();

                }

                private function initMediaPlayer():void
                {

                        // Create the container (sprite) for managing display and layout
                        _mediaPlayerSprite = new MediaPlayerSprite();
                        _mediaPlayerSprite.addEventListener(MediaErrorEvent.MEDIA_ERROR,
onPlayerFailed);

        _mediaPlayerSprite.addEventListener(MediaPlayerStateChangeEvent.MEDIA_PLAYER_STATE_CHANGE
, onPlayerStateChange);
                        _mediaPlayerSprite.scaleMode = ScaleMode.NONE;
                        _mediaPlayerSprite.width = stage.stageWidth;
```

```
                        _mediaPlayerSprite.height = stage.stageHeight;
                        //Adds the container to the stage
                        addChild(_mediaPlayerSprite);

                        // Create a mediafactory instance
                        _mediaFactory = new DefaultMediaFactory();

                        // Add the listeners for PLUGIN_LOADING

        _mediaFactory.addEventListener(MediaFactoryEvent.PLUGIN_LOAD,onPluginLoaded);
                        _mediaFactory.addEventListener(MediaFactoryEvent.PLUGIN_LOAD_ERROR,
onPluginLoadFailed );

                        // Load the plugin class
                        loadAdaptiveStreamingPlugin( );

                }

                private function loadAdaptiveStreamingPlugin( ):void
                {
                        var pluginResource:MediaResourceBase;

                        pluginResource = new PluginInfoResource(new AdaptiveStreamingPluginInfo(
));
                        _mediaFactory.loadPlugin( pluginResource );
                }

                private function onPluginLoaded( event:MediaFactoryEvent ):void
                {
                        // The plugin is loaded successfully.
                // Your web server needs to host a valid crossdomain.xml file to allow plugin to
        download Smooth Streaming files.

        loadMediaSource("http://wams.edgesuite.net/media/SintelTrailer_MP4_from_WAME/sintel_trail
er-1080p.ism/manifest")

                }

                private function onPluginLoadFailed( event:MediaFactoryEvent ):void
                {
                        // The plugin is failed to load ...
                }


                private function onPlayerStateChange(event:MediaPlayerStateChangeEvent) : void
                {
                        var state:String;

                        state =  event.state;

                        switch (state)
                        {
                                case MediaPlayerState.LOADING:

                                        // A new source is started to load.

                                        break;

                                case  MediaPlayerState.READY :
                                        // Add code to deal with Player Ready when it is hit the
first load after a source is loaded.

                                        break;

                                case MediaPlayerState.BUFFERING :

                                        break;

                                case  MediaPlayerState.PAUSED :
                                        break;
                                // other states ...
```

```
                }
            }

            private function onPlayerFailed(event:MediaErrorEvent) : void
            {
                // Media Player is failed .
            }

            private function loadMediaSource(sourceURL : String):void
            {
                // Take an URL of SmoothStreamingSource's manifest and add it to the page.

                var resource:URLResource= new URLResource( sourceURL );

                var element:MediaElement = _mediaFactory.createMediaElement( resource );
                _mediaPlayerSprite.scaleMode = ScaleMode.LETTERBOX;
                _mediaPlayerSprite.width = stage.stageWidth;
                _mediaPlayerSprite.height = stage.stageHeight;

                // Add the media element
                _mediaPlayerSprite.media = element;
            }

        }
}
```

## SS for OSMF static loading with support for AES-128 encrypted content

The code snippet below shows how to load SS plugin for OSMF statically and play an AES-128 encrypted video using OSMF MediaFactory. Before including SS OSMF code, please ensure that the project reference includes "MSAdaptiveStreamingPlugin-v1.x.x-osmf2.0.swc" static plugin.

Note: For use with AES-128 encrypted content, you must build with a minimum Adobe Flash Player version of 11.4 and add the additional compiler argument "–swf-version=17".

```
package
{
    import com.microsoft.azure.media.AdaptiveStreamingPluginInfo;
    import com.microsoft.azure.media.resources.AdaptiveStreamingManifestInfo;

    import flash.display.Sprite;
    import flash.display.StageQuality;

    import org.osmf.containers.MediaContainer;
    import org.osmf.events.MediaErrorEvent;
    import org.osmf.events.MediaFactoryEvent;
    import org.osmf.events.MediaPlayerStateChangeEvent;
    import org.osmf.layout.ScaleMode;
    import org.osmf.media.DefaultMediaFactory;
    import org.osmf.media.MediaElement;
    import org.osmf.media.MediaPlayerSprite;
    import org.osmf.media.MediaPlayerState;
    import org.osmf.media.MediaResourceBase;
    import org.osmf.media.PluginInfoResource;
    import org.osmf.media.URLResource;
    import com.microsoft.azure.media.traits.AdaptiveStreamingLoadTrait;
    import org.osmf.traits.*;
    import org.osmf.media.MediaPlayer;

    [SWF(width="1024", height="768", backgroundColor='#405050', frameRate="25")]
    public class TestSampleApp extends Sprite
    {
        public var _container:MediaContainer;
        public var _mediaFactory:DefaultMediaFactory;
        private var _mediaPlayer:MediaPlayer;
        private var _isFirstReadyState : Boolean = true;
        private var _mainMediaContainer:MediaContainer;
```

```
        public function TestSampleApp( )
        {
            stage.quality = StageQuality.HIGH;
            initMediaPlayer();
        }
        private function initMediaPlayer():void
        {
            // Create the container (sprite) for managing display and layout
            _mediaPlayer = new MediaPlayer();
            _mediaPlayer.addEventListener(MediaErrorEvent.MEDIA_ERROR,
                onPlayerFailed);
            _mediaPlayer.addEventListener(MediaPlayerStateChangeEvent.MEDIA_PLAYER_STATE_CHANGE,
onPlayerStateChange, false, 0 , true);

            //Add the container to the stage
            _mainMediaContainer = new MediaContainer();
            _mainMediaContainer.width = 1000;
            _mainMediaContainer.height = 700;
            this.addChild( _mainMediaContainer );

            // Create a mediafactory instance
            _mediaFactory = new DefaultMediaFactory();
            // Add the listeners for PLUGIN_LOADING
            _mediaFactory.addEventListener(MediaFactoryEvent.PLUGIN_LOAD,onPluginLoaded);
            _mediaFactory.addEventListener(MediaFactoryEvent.PLUGIN_LOAD_ERROR,
                onPluginLoadFailed );
            // Load the plugin class
            loadAdaptiveStreamingPlugin( );
        }
        private function loadAdaptiveStreamingPlugin( ):void
        {
            var pluginResource:MediaResourceBase;
            pluginResource = new PluginInfoResource(new AdaptiveStreamingPluginInfo(
            ));
            _mediaFactory.loadPlugin( pluginResource );
        }
        private function onPluginLoaded( event:MediaFactoryEvent ):void
        {
            // The plugin is loaded successfully.
            // Your web server needs to host a valid crossdomain.xml file to allow plugin to
            // download Smooth Streaming files.

            _isFirstReadyState = true;
                loadMediaSource("http://samplescdn.origin.mediaservices.windows.net/cdc285bd-
4f9b-422b-9f9f-c1cabcce04d2/BigBuckBunny.ism/manifest")
        }
        private function onPluginLoadFailed( event:MediaFactoryEvent ):void
        {
            // The plugin is failed to load ...
        }

        private function onPlayerStateChange(event:MediaPlayerStateChangeEvent) : void
        {
            var state:String;
            state = event.state;
            switch (state)
            {
                case MediaPlayerState.LOADING:
                    // A new source is started to load.
                    break;
                case MediaPlayerState.READY :

                    // Add code to deal with Player Ready when it is hit the
                    // first load after a source is loaded.
                    if(_isFirstReadyState)
                    {
                        var loadTrait:AdaptiveStreamingLoadTrait =
_mediaPlayer.media.getTrait(MediaTraitType.LOAD) as AdaptiveStreamingLoadTrait;
                        var _adaptiveManifestInfo: AdaptiveStreamingManifestInfo;
                        if(null != loadTrait)
```

```
                        {
                            _adaptiveManifestInfo = loadTrait.manifest;
                        }

                        if(_adaptiveManifestInfo.isEncrypted)
                        {
                            _adaptiveManifestInfo.setEncryptionKeyToken("Bearer=urn%3amicrosoft%3a
azure%3amediaservices%3acontentkeyidentifier=ca841fd4-b469-4326-9e61-
59b613c63971&Audience=urn%3atest&ExpiresOn=1764547200&Issuer=http%3a%2f%2ftestacs.com%2f&HMACSHA2
56=oPeHvCz%2fLvc2fBXiH9%2bhoOjslCnrWgezq0DT%2btj6lC4%3d");
                        }

                        _isFirstReadyState = false;
                    }
                    break;
                case MediaPlayerState.BUFFERING :
                    break;
                case MediaPlayerState.PAUSED :
                    break;
                // other states ...
            }
        }
        private function onPlayerFailed(event:MediaErrorEvent) : void
        {
            // Media Player is failed .
        }
        private function loadMediaSource(sourceURL : String):void
        {
            // Take an URL of SmoothStreamingSource's manifest and add it to the page.
            var resource:URLResource= new URLResource( sourceURL );
            var element:MediaElement = _mediaFactory.createMediaElement( resource );

            // Add the media element
            _mediaPlayer.media = element;
            _mainMediaContainer.addMediaElement(element);
        }
    }
}
```

## SS for OSMF dynamic loading

The code snippet below shows how to load SS plugin for OSMF dynamically and play a basic video using the OSMF MediaFactory class. Before including the SS for OSMF code, copy the "MSAdaptiveStreamingPlugin-v1.x.x-osmf2.0.swf" dynamic plugin to the project folder if you want to load using FILE protocol, or copy under a web server for HTTP load. There is no need to include "MSAdaptiveStreamingPlugin-v1.x.x-osmf2.0.swc" in the project references.

```
package
{
        // import related OSMF and Flash Runtime classes.

        import flash.display.*;
        import org.osmf.media.*;
        import org.osmf.containers.MediaContainer;
        import org.osmf.events.MediaErrorEvent;
        import org.osmf.events.MediaFactoryEvent;
        import org.osmf.events.MediaPlayerStateChangeEvent;
        import org.osmf.layout.*;
        import flash.events.Event;
        import flash.system.Capabilities;


        //Sets the size of the SWF
```

```
	[SWF(width="1024", height="768", backgroundColor='#405050', frameRate="25")]
	public class TestPlayer extends Sprite
	{
		public var _container:MediaContainer;
		public var _mediaFactory:DefaultMediaFactory;
		private var _mediaPlayerSprite:MediaPlayerSprite;


		public function TestPlayer( )
		{
			stage.quality = StageQuality.HIGH;
			initMediaPlayer();
		}

		private function initMediaPlayer():void
		{

			// Create the container (sprite) for managing display and layout
			_mediaPlayerSprite = new MediaPlayerSprite();
			_mediaPlayerSprite.addEventListener(MediaErrorEvent.MEDIA_ERROR,
onPlayerFailed);

	_mediaPlayerSprite.addEventListener(MediaPlayerStateChangeEvent.MEDIA_PLAYER_STATE_CHANGE
, onPlayerStateChange);

			//Adds the container to the stage
			addChild(_mediaPlayerSprite);

			// Create a mediafactory instance
			_mediaFactory = new DefaultMediaFactory();

			// Add the listeners for PLUGIN_LOADING

	_mediaFactory.addEventListener(MediaFactoryEvent.PLUGIN_LOAD,onPluginLoaded);
			_mediaFactory.addEventListener(MediaFactoryEvent.PLUGIN_LOAD_ERROR,
onPluginLoadFailed );

			// Load the plugin class
			loadAdaptiveStreamingPlugin( );

		}

		private function loadAdaptiveStreamingPlugin( ):void
		{
			var pluginResource:MediaResourceBase;
			var adaptiveStreamingPluginUrl:String;
		// Your dynamic plugin web server needs to host a valid crossdomain.xml file to
	allow loading plugins.

			adaptiveStreamingPluginUrl = "http://yourdoamin/MSAdaptiveStreamingPlugin-
v1.x.x-osmf2.0.swf";

			pluginResource = new URLResource(adaptiveStreamingPluginUrl);
			_mediaFactory.loadPlugin( pluginResource );

		}

		private function onPluginLoaded( event:MediaFactoryEvent ):void
		{
			// The plugin is loaded successfully.

	// Your web server needs to host a valid crossdomain.xml file to allow plugin to download
	Smooth Streaming files.

	loadMediaSource("http://wams.edgesuite.net/media/SintelTrailer_MP4_from_WAME/sintel_trail
er-1080p.ism/manifest")
		}

		private function onPluginLoadFailed( event:MediaFactoryEvent ):void
		{
```

```
                        // The plugin is failed to load ...
                }


                private function onPlayerStateChange(event:MediaPlayerStateChangeEvent) : void
                {
                        var state:String;

                        state =  event.state;

                        switch (state)
                        {
                                case MediaPlayerState.LOADING:

                                        // A new source is started to load.

                                        break;

                                case  MediaPlayerState.READY :
                                        // Add code to deal with Player Ready when it is hit the
first load after a source is loaded.

                                        break;

                                case MediaPlayerState.BUFFERING :

                                        break;

                                case  MediaPlayerState.PAUSED :
                                        break;
                                // other states ...
                        }
                }

                private function onPlayerFailed(event:MediaErrorEvent) : void
                {
                        // Media Player is failed .
                }

                private function loadMediaSource(sourceURL : String):void
                {
                        // Take an URL of SmoothStreamingSource's manifest and add it to the page.

                        var resource:URLResource= new URLResource( sourceURL );

                        var element:MediaElement = _mediaFactory.createMediaElement( resource );
                        _mediaPlayerSprite.scaleMode = ScaleMode.LETTERBOX;
                        _mediaPlayerSprite.width = stage.stageWidth;
                        _mediaPlayerSprite.height = stage.stageHeight;
                        // Add the media element
                        _mediaPlayerSprite.media = element;
                }

        }
}
```

## Strobe Media Playback with SS OSMF dynamic plugin

The Smooth Streaming for OSMF dynamic plugin is compatible with Strobe Media Playback (SMP). You can use SS for OSMF plugin to add Smooth Streaming content playback to SMP. For playback using SMP with SS OSMF dynamic plugin, use the following steps:

1. Copy "MSAdaptiveStreamingPlugin-v1.x.x-osmf2.0.swf" and "StrobeMediaPlayback.2.0.swf" under a web server for HTTP load. (Download link)

2. Create an HTML page using your favorite text editor with the following sample code, editing the highlighted artifacts.
   (**Note**: If your URL contains special characters, you should encode the URL)

```
<html>
<body>
<object width="920" height="640">
<param name="movie" value="http://yourdomain/StrobeMediaPlayback.2.0.swf"></param>
<param name="flashvars"
value="src=http://wams.edgesuite.net/media/SintelTrailer_MP4_from_WAME/sintel_trailer-
1080p.ism/manifest&autoPlay=true"></param>
<param name="allowFullScreen" value="true"></param>
<param name="allowscriptaccess" value="always"></param>
<param name="wmode" value="direct"></param>
<embed src="http://yourdomain/StrobeMediaPlayback.2.0.swf"
    type="application/x-shockwave-flash"
    allowscriptaccess="always"
    allowfullscreen="true"
    wmode="direct"
    width="920"
    height="640"
    flashvars="src=http://wams.edgesuite.net/media/SintelTrailer_MP4_from_WAME/sintel_trail
    er-1080p.ism/manifest&autoPlay=true"></embed>
</object>
</body>
</html>
```

3. Add Smooth Streaming OSMF plugin to the embed code and save.

```
<html>
<body>
<object width="920" height="640">
<param name="movie" value="http://yourdomain/StrobeMediaPlayback.2.0.swf"></param>
<param name="flashvars"
value="src=http://wams.edgesuite.net/media/SintelTrailer_MP4_from_WAME/sintel_trailer-
1080p.ism/manifest&autoPlay=true&plugin_AdaptiveStreamingPlugin=http://yourdoamin/MSAdapti
veStreamingPlugin-v1.x.x-
osmf2.0.swf&AdaptiveStreamingPlugin_retryLive=true&AdaptiveStreamingPlugin_retryInterval=1
0"></param>
<param name="allowFullScreen" value="true"></param>
<param name="allowscriptaccess" value="always"></param>
<param name="wmode" value="direct"></param>
<embed src="http://yourdomain/StrobeMediaPlayback.2.0.swf"
    type="application/x-shockwave-flash"
    allowscriptaccess="always"
    allowfullscreen="true"
    wmode="direct"
    width="920"
    height="640"
flashvars="src=http://wams.edgesuite.net/media/SintelTrailer_MP4_from_WAME/sintel_trailer-
1080p.ism/manifest&autoPlay=true&plugin_AdaptiveStreamingPlugin=http://yourdoamin/MSAdapti
veStreamingPlugin-v1.x.x-
```

```
osmf2.0.swf&AdaptiveStreamingPlugin_retryLive=true&AdaptiveStreamingPlugin_retryInterval=1
0"></embed>
</object>
</body>
</html>
```

4. If you require to play AES encrypted Smooth Streaming with Token authentication add the "AdaptiveStreamPlugin_encryptionKeyToken" to the code and save
   (**Note**: If your URL or Token contains special characters, you should encode the string

   Eg. **Original Token**:

   ```
   Bearer=urn%3amicrosoft%3aazure%3amediaservices%3acontentkeyidentifier=ca8
   41fd4-b469-4326-9e61-
   59b613c63971&Audience=urn%3atest&ExpiresOn=1764547200&Issuer=http%3a%2f%2
   ftestacs.com%2f&HMACSHA256=oPeHvCz%2fLvc2fBXiH9%2bhoOjslCnrWgezq0DT%2btj6
   lC4%3d
   ```

   **Encoded Token**:

   ```
   Bearer%3Durn%253amicrosoft%253aazure%253amediaservices%253acontentkeyiden
   tifier%3Dca841fd4-b469-4326-9e61-
   59b613c63971%26Audience%3Durn%253atest%26ExpiresOn%3D1764547200%26Issuer%
   3Dhttp%253a%252f%252ftestacs.com%252f%26HMACSHA256%3DoPeHvCz%252fLvc2fBXi
   H9%252bhoOjslCnrWgezq0DT%252btj6lC4%253d
   ```

   )

```
<html>
<object width="920" height="640">
<param name="movie" value="http://yourdomain/StrobeMediaPlayback.2.0.swf"></param>
<param name="flashvars"
value="src=http://samplescdn.origin.mediaservices.windows.net/cdc285bd-4f9b-422b-9f9f-
c1cabcce04d2/BigBuckBunny.ism/manifest&autoPlay=true&plugin_AdaptiveStreamingPlugin=http:/
/yourdoamin/MSAdaptiveStreamingPlugin-v1.x.x-
osmf2.0.swf&AdaptiveStreamingPlugin_retryLive=true&AdaptiveStreamingPlugin_retryInterval=1
0&AdaptiveStreamingPlugin_encryptionKeyToken=Bearer%3Durn%253amicrosoft%253aazure%253amedi
aservices%253acontentkeyidentifier%3Dca841fd4-b469-4326-9e61-
59b613c63971%26Audience%3Durn%253atest%26ExpiresOn%3D1764547200%26Issuer%3Dhttp%253a%252f%
252ftestacs.com%252f%26HMACSHA256%3DoPeHvCz%252fLvc2fBXiH9%252bhoOjslCnrWgezq0DT%252btj6lC
4%253d"></param>

<param name="allowFullScreen" value="true"></param>
<param name="allowscriptaccess" value="always"></param>
<param name="wmode" value="direct"></param>
<embed src="http://yourdomain/StrobeMediaPlayback.2.0.swf"
       type="application/x-shockwave-flash"
       allowscriptaccess="always"
       allowfullscreen="true"
       wmode="direct"
       width="920"
       height="640"
```

```
        flashvars="src=http://samplescdn.origin.mediaservices.windows.net/cdc285bd-4f9b-
        422b-9f9f-
        c1cabcce04d2/BigBuckBunny.ism/manifest&autoPlay=true&plugin_AdaptiveStreamingPlugin
        =http://yourdoamin/MSAdaptiveStreamingPlugin-v1.x.x-
        osmf2.0.swf&AdaptiveStreamingPlugin_retryLive=true&AdaptiveStreamingPlugin_retryInt
        erval=10&AdaptiveStreamingPlugin_encryptionKeyToken=Bearer%3Durn%253amicrosoft%253a
        azure%253amediaservices%253acontentkeyidentifier%3Dca841fd4-b469-4326-9e61-
        59b613c63971%26Audience%3Durn%253atest%26ExpiresOn%3D1764547200%26Issuer%3Dhttp%253
        a%252f%252ftestacs.com%252f%26HMACSHA256%3DoPeHvCz%252fLvc2fBXiH9%252bhoOjslCnrWgez
        q0DT%252btj6lC4%253d"></embed>

</object>
</html>
```

5. Save your HTML page and publish to a web server. Browse the published web page using your favorite Flash runtime enabled internet browser (eg. IE, Chrome, Firefox, so on).

Enjoy Smooth Streaming content inside Flash™ runtime.

For more information on general OSMF development, please see official OSMF development page.