



OSMF Release Samples

Walkthrough 1: The Simplest Player

Overview:

In this walkthrough we will create a very simple media player using the OSMF and the MediaPlayerSprite. The purpose is to see how to create the simplest form of an OSMF player and playback progressive, streaming FLV and H.264 media.

Objectives:

- Create the simplest form of a OSMF media player
- Use the MediaPlayerSprite to handle the heavy lifting and makes for a simple component implementation
- Simply and directly create a URLResource and VideoElement
- Playback Progressive and RTMP Streaming using the same player
- Playback a streaming H.264 MPEG based file by applying the proper path prefix

Setup

1. Open the file WT01_SimplestPlayer.as in the {SAMPLES_PROJECT}/src directory.

NOTE: This file has been provided as a starting point for these walkthroughs.

2. Set the class file as the application file to compile. There are two different ways of doing this depending on which program you are building your application in.
Flash Builder
Right-click the WT01_SimplestPlayer.as file and select Set as Default Application from the context menu that appears. This will add the project to the list of compilable applications. A blue dot on the file icon indicates that the file is the default application file.
Flash Professional
Open the OSMF_SampleTemplate.fla and save it as WT01_SimplestPlayer.fla. Then change the document class for the file (in the Properties panel) to WT01_SimplestPlayer.

Building the Simplest Player

3. Under the "DECLARATION" comment, create a public variable named playerSprite that is typed as a MediaPlayerSprite object.

```
public var playerSprite:MediaPlayerSprite;
```

4. Locate the `initPlayer()` method. This method will be called when the media player starts up.

5. Inside the `initPlayer()` method, set the `playerSprite` property equal to a new `MediaPlayerSprite` object.

```
playerSprite = new MediaPlayerSprite();
```

6. Create a local variable named `resource` that is typed as a `URLResource` object. Set this equal to a new `URLResource` object and make sure you pass the static const variable `PROGRESSIVE_PATH` as the only parameter to the constructor.

```
playerSprite = new MediaPlayerSprite();  
var resource:URLResource = new URLResource( PROGRESSIVE_PATH );
```

7. Set the `media` property of the `playerSprite` object equal to a new `VideoElement` object passing the `resource` as the only parameter to the constructor.

```
playerSprite = new MediaPlayerSprite();  
var resource:URLResource = new URLResource( PROGRESSIVE_PATH );  
playerSprite.media = new VideoElement( resource );
```

8. Call the `addChild()` method passing it the `playerSprite`.

```
playerSprite = new MediaPlayerSprite();  
var resource:URLResource = new URLResource( PROGRESSIVE_PATH );  
playerSprite.media = new VideoElement( resource );  
addChild( playerSprite );
```

9. The completed `initPlayer()` method should look like the following:

```
protected function initPlayer():void  
{  
    playerSprite = new MediaPlayerSprite();  
    var resource:URLResource = new URLResource( PROGRESSIVE_PATH );  
    playerSprite.media = new VideoElement( resource );  
    addChild( playerSprite );  
}
```

10. Save the file and run `WT1_SimplestPlayer.as` as a Web Application. The player should start up and play the progressive *A Faery's Tale* video.



11. Change the path that is passed to the URLResource constructor to be STREAMING_PATH. Notice that this path starts with "rtmp://" and does not have a file extension. Save the file and run it as a Web Application. Now the Akamai Anniversary video should play. The same code for the progressive video playback works great for streaming media.



12. Update the path that is passed to the URLResource to be STREAMING_MP4_PATH. Make note of the "mp4:" that is part of this path and that this path does have a file extension. When this version is run the streaming H.264 Elephants Dream video should play.

