



OSMF Release Samples

Walkthrough 2: Separating Control

Overview:

In this walkthrough we will break out the main control objects of an OSMF media player. Using the different parts available to create an OSMF experience and how they can be managed independently instead of the all in one MediaPlayerSprite. In the end it all works the same but now there is separate instances to control the connection, playback, and visual display.

Objectives:

This walkthrough will demonstrate the use of:

- **NetLoader:** To manage the external connection/loading for the VideoElement (MediaElement)
- **MediaPlayer:** A non-visual item to manage the playback and control of a MediaElement
- **MediaContainer:** The visual container for managing the display and layout of a MediaElement

Setup

1. Open the file WT02_SeparatingControl.as in the {SAMPLES_PROJECT}/src directory.
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 WT02_SeparatingControl.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 WT02_SeparatingControl.fla. Then change the document class for the file (in the Properties panel) to WT02_SeparatingControl.

Breaking Out the Control Objects

3. First remove the public var playerSprite and replace it with 2 variables - player typed as a MediaPlayer object and container typed as a MediaContainer

```
public var player:MediaPlayer;  
public var container:MediaContainer;
```

4. Next, inside the `initPlayer()` method, remove the line that created the `playerSprite` as well as the other `playerSprite` references - including the `addChild()` method. The `initPlayer()` method should look similar to the following:

```
protected function initPlayer():void
{
    var resource:URLResource = new URLResource( STREAMING_MP4_PATH );
}
```

5. Under the declaration of the resource variable, create a new local variable named `netLoader` and set it equal to a new `NetLoader` object. *The `NetLoader` exposes the ability to have more control over the connection made to load the media.*

```
var resource:URLResource = new URLResource( STREAMING_MP4_PATH );
var netLoader:NetLoader = new NetLoader();
```

6. Create a `VideoElement` object named `element` and pass the `VideoElement`'s constructor the resource and `netLoader` variables as parameters. *Now when the `VideoElement` is created it will use our explicit `NetLoader`.*

```
var resource:URLResource = new URLResource( STREAMING_MP4_PATH );
var netLoader:NetLoader = new NetLoader();
var element:VideoElement = new VideoElement( resource, netLoader );
```

6. Set the `player` property equal to a new instance of the `MediaPlayer` object. Make sure to pass the `element` variable to the constructor.

```
var element:VideoElement = new VideoElement( resource, netLoader );
player = new MediaPlayer( element );
```

7. Set the `container` property equal to a new instance of the `MediaContainer` object, then call the `addMediaElement()` method on the `container` property passing it the `element` variable.

```
var element:VideoElement = new VideoElement( resource, netLoader );
player = new MediaPlayer( element );
container = new MediaContainer();
container.addMediaElement( element );
```

8. Call the `addChild()` method and pass the `container` property to the `addChild()` method. The completed `initPlayer()` method should look like the following:

```
protected function initPlayer():void
{
    var resource:URLResource = new URLResource( STREAMING_MP4_PATH );
    var netLoader:NetLoader = new NetLoader();
    var element:VideoElement = new VideoElement( resource, netLoader );
    player = new MediaPlayer( element );
    container = new MediaContainer();
    container.addMediaElement( element );
    addChild( container );
}
```

9. Save and run the application. The Elephant's Dream video should play just as in walk through 1. What this walkthrough has done is exposed the control

elements - NetLoader,
application.

MediaPlayer, and MediaContainer - of the media playback