



OSMF Release Samples

Walkthrough 6: Basic Parallel Element Composition

Overview:

In this walk through you will further your knowledge of composition elements with the use of the ParallelElement to display two MediaElement instances at the same time. A key area of focus when displaying multiple element at the same time is layout. Utilizing the versatile OSMF LayoutContainer and appropriate meta-data this task becomes easily manageable. The goal of this exercise is to play 2 media items generally in sync, and evenly displayed within the constraints of the LayoutContainer no matter what its size.

Objectives:

- Understand the creation and use of the ParallelElement for simultaneous media display
- Gain a better understanding of how to control the layout with meta-data for composite elements
- Utilize advanced layout meta-data capabilities such as:
 - Percent based layout constraints
 - Vertical/Horizontal layoutMode rendering and alignment

Setup

1. Open the file WT06_BasicParallelComposition.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 WT06_BasicParallelComposition.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 WT06_BasicParallelComposition.fla. Then change the document class for the file (in the Properties panel) to WT06_BasicParallelComposition.

Working with Parallel Elements

3. Locate the comment that begins "//Marker 1:" in the initPlayer() method.
4. Under the comment create a LayoutMetadata variable named layoutData that sets the percentWidth to 50, the percentHeight to 50 and the scaleMode to

letterbox. This will provide a reusable LayoutMetadata that can be applied to both media items that we want to display in unison, and make them of equal scale.

```
//Marker 1: Create a LayoutMetaData object to even out the 2
parallel streams initially
var layoutData:LayoutMetadata = new LayoutMetadata();
layoutData.percentWidth = 50;
layoutData.percentHeight = 50;
layoutData.scaleMode = ScaleMode.LETTERBOX;
```

5. Under the "//Marker 2:" comment, create a leftElement MediaElement variable using the LOGO_VID static constant.
6. Add the layoutData to this element by calling the addValue() method on the metadata property of the left element object.

```
//Marker 2: Create the left side Media Element to play the LOGO_VID
and apply the meta-data
var leftElement:MediaElement = mediaFactory.createMediaElement( new
URLResource( LOGO_VID ) );
leftElement.metadata.addValue( LayoutMetadata.LAYOUT_NAMESPACE,
layoutData );
```

7. Under the "//Marker 3:" comment, create a MediaElement variable named rightElement using the STREAMING_PATH static constant and apply the layoutData to this element as well.

```
//Marker 3: Create the right side Media Element to play the
STREAMING_PATH and apply the meta-data
var rightElement:MediaElement = mediaFactory.createMediaElement( new
URLResource( STREAMING_PATH ) );
rightElement.metadata.addValue( LayoutMetadata.LAYOUT_NAMESPACE,
layoutData );
```

8. After the "//Marker 4:" comment, create a ParallelElement variable named parallelElement. And add the leftElement and rightElement as children.

```
//Marker 4: Create the ParallelElement and add the left and right
elements to it
var parallelElement:ParallelElement = new ParallelElement();
parallelElement.addChild( leftElement );
parallelElement.addChild( rightElement );
```

9. Reuse the layoutData variable to create a new LayoutMetadata object that we will apply to the parallelElement object.
10. Set the following properties and values on the layoutData object:
 1. layoutMode: LayoutMode.HORIZONTAL
 2. horizontalAlign: HorizontalAlign.CENTER
 3. verticalAlign: VerticalAlign.MIDDLE
 4. width: 800
 5. height: 600

NOTE: Setting the layoutMode to LayoutMode.HORIZONTAL

or `LayoutMode.VERTICAL` can be very powerful for dynamically managing the layout of `MediaElement` instances within a `LayoutContainer`. It dynamically sets their positions appropriately based on the the elements display and individual constraints. This is very similar to the Flex framework layout capabilities.

11. Add the `layoutData` to the `metadata` property of the `parallelElement` object.

```
//Marker 5: Re-instantiate the layoutData to clear it out and set the
layout data for the parallel element
layoutData = new LayoutMetadata();
layoutData.layoutMode = LayoutMode.HORIZONTAL;
layoutData.horizontalAlign = HorizontalAlign.CENTER;
layoutData.verticalAlign = VerticalAlign.MIDDLE;
layoutData.width = 800;
layoutData.height = 600;
parallelElement.metadata.addValue( LayoutMetadata.LAYOUT_NAMESPACE,
layoutData );
```

12. The completed code should look like the following:

```
protected function initPlayer():void
{
// Create a MediaFactory instance
mediaFactory = new DefaultMediaFactory();

//Marker 1: Create a LayoutMetaData object up to even out the 2
parallel streams initially
var layoutData:LayoutMetadata = new LayoutMetadata();
layoutData.percentWidth = 50;
layoutData.percentHeight = 50;
layoutData.scaleMode = ScaleMode.LETTERBOX;

//Marker 2: Create the left side Media Element to play the LOGO_VID
and apply the meta-data
var leftElement:MediaElement = mediaFactory.createMediaElement( new
URLResource( LOGO_VID ) );
leftElement.metadata.addValue( LayoutMetadata.LAYOUT_NAMESPACE,
layoutData );

//Marker 3: Create the right side Media Element to play the
STREAMING_PATH and apply the meta-data
var rightElement:MediaElement = mediaFactory.createMediaElement( new
URLResource( STREAMING_PATH ) );
rightElement.metadata.addValue( LayoutMetadata.LAYOUT_NAMESPACE,
layoutData );

//Marker 4: Create the Parallelelement and add the left and right
elements to it
var parallelElement:Parallelelement = new Parallelelement();
parallelElement.addChild( leftElement );
parallelElement.addChild( rightElement );

//Marker 5: Re-instantiate the layoutData to clear it out and set
```

```
the layout data for the parallel element
layoutData = new LayoutMetadata();
layoutData.layoutMode = LayoutMode.HORIZONTAL;
layoutData.horizontalAlign = HorizontalAlign.CENTER;
layoutData.verticalAlign = VerticalAlign.MIDDLE;
layoutData.width = 800;
layoutData.height = 600;
paralleElement.metadata.addValue( LayoutMetadata.LAYOUT_NAMESPACE,
layoutData );

//the simplified api controller for media
player = new MediaPlayer( paralleElement );

//the container (sprite) for managing display and layout
container = new MediaContainer();
container.addMediaElement( paralleElement );

//Adds the container to the stage
this.addChild( container );
}
```

13. Save the file and run the application. Both media clips should play side-by-side.

