



OSMF Release Samples

Walkthrough 5: Basic Serial Elements

Overview:

This walkthrough creates a sequence of media items to create an interactive movie experience complete with a progressive video pre-roll, a dynamic streaming preview, a SWF based quiz and if passed, access to watch the whole clip via dynamic streaming. This is achieved by using the SerialElement to create the sequence of MediaElements, and the StreamingURLResource to create a sub-clip of a streaming media item. Then based on user interaction we can completely overwrite the current media being displayed (the SerialElement) and play 'unlocked' content.

Objectives:

- Create a SerialElement and add different types of media to it
- Create a sub-clip of a streaming item by use of the StreamingURLResource
- Use the metaData property and LayoutMetadata to make the display consistent across the media types.

Setup

1. Open the file WT05_SerialCompositions.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 WT05_SerialCompositions.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 WT05_SerialCompositions.fla. Then change the document class for the file (in the Properties panel) to WT05_SerialCompositions.

Working With Serial Elements

3. Please note that 2 new static constant variables have been added:
 1. LOGO_VID
 2. QUIZ_SWF
4. Locate the comment that begins with "//Marker 1:" in the initPlayer() method.

5. Under the comment create a new local MediaElement variable named preRoll and set this equal to the result of calling the createMediaElement() method on the mediaFactory object. Make sure to create a new URLResource object using the LOGO_VID static const variable.

```
//Marker 1: create a video element for the pre-roll video - use the newly added LOGO_VID const
var preRoll:MediaElement = mediaFactory.createMediaElement( new URLResource( LOGO_VID ) );
```

6. Under the comment starting "//Marker 2:", create a new local StreamingURLResource variable named resourceStart. The StreamingURLResource extends the URLResource class which has been used often before. The can be used when playing back streaming content - such as RTMP based content streamed from a Flash Media Server or Flash Video Streaming CDN service. The StreamingURLResource allows for some additional control over how the content is handled.
7. Set resourceStart equal to a new StreamingURLResource object. Pass the DYNAMIC_STREAMING const variable as the first parameter - which is the URI, the RECORDED static property of the StreamType object as the second parameter - which says the stream is a recorded VOD stream, zero (0) as the third - which indicates to start the stream at the beginning (0 seconds), and twenty (20) as the last parameter - which defines how much of the stream to play. This will create a sub-clip of the first 20 seconds of the video stream.

```
//Marker 2: Create a StreamingURLResource to create a sub clip of the streaming resource - play from 0-20 sec
var resourceStart:StreamingURLResource = new StreamingURLResource( DYNAMIC_STREAMING, StreamType.RECORDED, 0, 20 );
```

8. Under the "//Marker 3:" comment create a local MediaElement variable named element and set it equal to the a new MediaElement created using the mediaFactory and resourceStart objects.

```
//Marker 3: Create the media element from the streaming resource sub clip
var element:MediaElement = mediaFactory.createMediaElement( resourceStart );
```

9. Under the "//Marker 4:" comment, create a new local MediaElement variable named quiz. Use the QUIZ_SWF static const as the path for the new MediaElement.

```
//Marker 4: Create another post roll clip that is an interactive SWF piece.
var quiz:MediaElement = mediaFactory.createMediaElement( new URLResource( QUIZ_SWF ) );
```

10. After the "//Marker 5:" add an event listener on the this object for the a "correct" event and set the handler method to be the _onCorrect() method.

```
//Marker 5: Create a event listener for "correct" bubbled up from the SWF if response is correct
this.addEventListener( "correct", _onCorrect );
```

11. After the "//Marker 6:" comment create a new SerialElement local variable named serialElement and set it equal to a new SerialElement().

```
//Marker 6: Construct the SerialElement adding each media element in
the desired order
var serialElement:SerialElement = new SerialElement();
```

12. Add each of the three MediaElement we have created ('preRoll', 'element' and 'quiz') to the SerialElement by calling the addChild() method and passing it the MediaElement to add.

```
//Marker 6: Construct the SerialElement adding each media element in
the desired order
var serialElement:SerialElement = new SerialElement();
serialElement.addChild( preRoll );
serialElement.addChild( element );
serialElement.addChild( quiz );
```

13. Under the "//Marker 7:" comment, create a local LayoutMetadata variable named layoutData. Set the following properties and values:
1. scaleMode: ScaleMode.LETTERBOX
 2. width: 800
 3. height: 600

```
//Marker 7: Create a LayoutMetaData object to set the unified
display parameters for the serial composition and apply it to
the metadata
var layoutData:LayoutMetadata = new LayoutMetadata();
layoutData.scaleMode = ScaleMode.LETTERBOX;
layoutData.width = 800;
layoutData.height = 600;
```

NOTE: When using composition elements such as the SerialElement the LayoutMetaData can be set directly to the composition element to achieve consistent display of its children.

14. The completed initPlayer() method code should look like the following:

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

//Marker 1: create a video element for the pre-roll video - use the
newly added LOGO_VID const
var preRoll:MediaElement = mediaFactory.createMediaElement( new
URLResource( LOGO_VID ) );

//Marker 2: Create a StreamingURLResource to create a sub clip of
the streaming resource - play from 0-20 sec
var resourceStart:StreamingURLResource = new StreamingURLResource(
DYNAMIC_STREAMING, StreamType.RECORDED, 0, 20 );
```

```

//Marker 3: Create the media element from the streaming resource sub
clip
var element:MediaElement = mediaFactory.createMediaElement(
resourceStart );

//Marker 4: Create another post roll clip that is an interactive SWF
piece.
var quiz:MediaElement = mediaFactory.createMediaElement( new
URLResource( QUIZ_SWF ) );

//Marker 5: Create a event listener for "correct" bubbled up from
the SWF if response is correct
this.addEventListener( "correct", _onCorrect );

//Marker 6: Construct the SerialElement adding each media element in
the desired order
var serialElement:SerialElement = new SerialElement();
serialElement.addChild( preRoll );
serialElement.addChild( element );
serialElement.addChild( quiz );

//Marker 7: Create a LayoutMetaData object to set the unified
display parameters for the serial composition and apply it to the
metadata
var layoutData:LayoutMetadata = new LayoutMetadata();
layoutData.scaleMode = ScaleMode.LETTERBOX;
layoutData.width = 800;
layoutData.height = 600;

serialElement.metadata.addValue( LayoutMetadata.LAYOUT_NAMESPACE,
layoutData );

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

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

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

```

15. Locate the comment that begins "//Marker 8:" in the _onCorrect() event handler method
16. create a new local MediaElement variable and set it equal to a new StreamingURLResource object that creates a sub clip of a DYNAMIC_STREAMING resource that plays from 15 seconds into the clip to the end.

```

//Marker 8: Note the below - and that if correct answer was
triggered full play is accessed
var element:MediaElement = mediaFactory.createMediaElement( new

```

```
StreamingURLResource( DYNAMIC_STREAMING, StreamType.RECORDED, 15 )
);
```

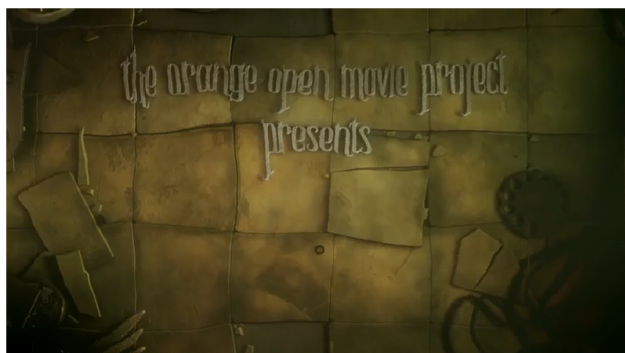
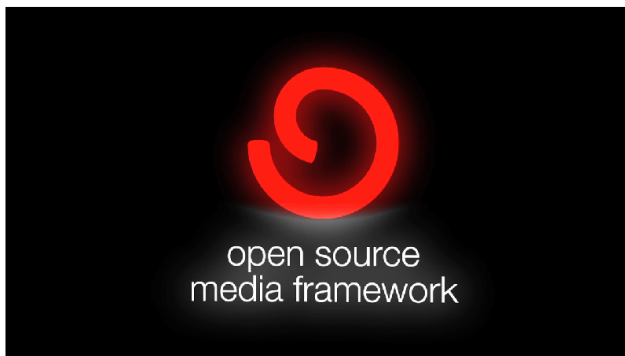
17. Set the media property of the player object equal to the element variable.

```
//Marker 8: Note the below - and that if correct answer was
triggered full play is accessed
var element:MediaElement = mediaFactory.createMediaElement( new
StreamingURLResource( DYNAMIC_STREAMING, StreamType.RECORDED, 15 )
);
player.media = element;
```

18. Call the addMediaElement() method on the container object passing it the element variable.

```
//Marker 8: If correct answer was triggered full play is accessed
var element:MediaElement = mediaFactory.createMediaElement( new
StreamingURLResource( DYNAMIC_STREAMING, StreamType.RECORDED, 15 )
);
player.media = element;
container.addMediaElement( element );
```

19. Save the file and run the application. The player should play the OSMF logo clip, then 20 seconds of the media clip. Then present a form that asks for the title of the movie. If the title is entered correctly ('elephants dream') the rest of the clip is played.



What is the title of the feature?