



## OSMF Release Samples

### Walkthrough 11: Advertising Integration

#### Overview:

This walk through shows how to use the MAST advertising plugin to manage ad insertion as per-roll or post-roll content. The key aspects of this walk through is the simplicity of integration and the control which can be achieved by using the industry standard MAST (or VAST) data format with external XML driven content and linking that data with OSMF and meta-data. The process is the same as the other samples when it comes to loading a plug-in, but in this case we simply apply the MAST meta-data to the media resource and OSMF does the rest.

**NOTE:** *This sample may have an issue if the compiler variable `CONFIG::LOGGING` is set to true. It appears to throw an error in the `MASTConditionManager` (Line #291) trying to use the logger that is null at that point.*

#### Objectives:

- Load in the static MAST plug-in at runtime
- Once the plug-in is loaded apply the MAST meta-data to the media resource
- Understand the basic pieces and format of a MAST sample
- Be able to change from a pre-roll to a post-roll by utilizing the MAST data

#### Setup

1. Open `WT11_CustomAdverts.as` in the `{SAMPLES_PROJECT}/src` directory.

NOTE: These files have 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 `WT11_CustomAdverts.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 `WT11_CustomAdverts.fla`. Then change the document class for the file (in the Properties panel) to `WT11_CustomAdverts`.

## Specifying advertisements with VAST/MAST

3. Under the "//Marker 1:" comment in the initPlayer() method, call the loadPlugin() method and pass it the MAST\_PLUGIN\_INFO\_CLASS string constant.

```
//Marker 1: Load the plugin
loadPlugin( MAST_PLUGIN_INFOCLASS );
```

4. Open the mast\_sample\_onitemstart.xml file in the {SAMPLES\_PROJECT}/assets/mast/ directory. This is a MAST file that defines simple advertising for the player.

**NOTE:** Media Abstract Sequencing Template (MAST) is "an XML based declarative language to define a set of triggers and conditions to specify sequencing and layout of advertising for a video player". You can find out more about the Akamai based MAST specification here: [http://openvideoplayer.sourceforge.net/mast/mast\\_specification.pdf](http://openvideoplayer.sourceforge.net/mast/mast_specification.pdf)

5. The mast\_sample\_onitemstart.xml file defines 3 main things:
  1. A trigger: 'preroll'
  2. A condition: 'OnItemStart'
  3. A source: The advertising media
6. This defines a short video clip that will play before the original media we have told the player to play plays. After you have reviewed the MAST file close it.
7. In the loadMedia() method, under the "//Marker 2:" create a new Metatadata object named metadata.

```
//Marker 2: Assign to the resource the metadata that indicates...
var metadata:Metadata = new Metadata();
```

8. Call the addValue() method on the metadata object passing MASTPluginInfo.MAST\_METADATA\_KEY\_URI as the first parameter and the static constant string MAST\_URL\_PREROLL as the second parameter.

```
//Marker 2: Assign to the resource the metadata that indicates...
var metadata:Metadata = new Metadata();
metadata.addValue( MASTPluginInfo.MAST_METADATA_KEY_URI,
MAST_URL_PREROLL );
```

9. Add the metadata to the resource by calling the addMetadataValue() method on the resource object passing it MASTPlugininfo.MAST\_METADATA\_NAMESPACE as the first parameter and the metadata object as the second parameter.

```
//Marker 2: Assign to the resource the metadata that indicates...
var metadata:Metadata = new Metadata();
metadata.addValue( MASTPluginInfo.MAST_METADATA_KEY_URI,
MAST_URL_PREROLL );
resource.addMetadataValue( MASTPluginInfo.MAST_METADATA_NAMESPACE,
metadata );
```

10. Save and run the application. A short video 'ad' will be played before the specified video plays.



11. Change the second parameter of the addValue() method call from MAST\_URL\_PREROLL to MAST\_URL\_POSTROLL. This changes the MAST file from mast\_sample\_onitemstart.xml to mast\_sample\_onitemend.xml.
12. If you open the review mast\_sample\_onitemend.xml it defines the same data, trigger, condition, source - and defines a video 'ad' that will play after the original media has played.
13. Save and run the application. The video add should play after the original media.

