OSMF Sprint 9 Release Notes

What's New

Getting Started Instructions

Sample Applications

Compatibility

Known Issues

Documentation

What's New

Feature Specifications for all features can be found on the OSMF Open Source site:

http://opensource.adobe.com/wiki/display/osmf/Specifications

OSMF Sample Player

The OSMF Sample Player defines an application that can be embedded on a web page in order to play back media. It contains a control bar that manages the various supported aspects of the media. It also uses the ChromeLibrary, which serves as a reference on implementing a user interface for an OSMF based player.

HTTP Streaming

We've added HTTP streaming support in this build of OSMF. If you would like to test this new functionality, please sign up for our pre-release at: https://www.adobe.com/cfusion/mmform/index.cfm?name=prerelease_interest Please note that the prerelease enrollment form at this link will not be available until approximately 2/12/10.

Enhanced Plug-in Support

Added CREATE_ON_LOAD plugin type and the initializePlugin() method to PluginInfo.

Display Class Refactoring

Removed duplicate APIs. The MediaElement has been replaced with the MediaContainer. The MediaContainer also left the scalableSprite as unneeded.

SMIL Plug-in

Adds standards-based support to the OSMF for SMIL (pronounced "smile"), the Synchronized Multimedia Integration Language.

Sprint 9 API Changes

We've made a significant number of API changes in Sprint 9. For details on the status of our efforts to lock down APIs and commit to backwards compatibility, please see our API Lockdown Status page here:

http://opensource.adobe.com/wiki/display/osmf/API+Lockdown+Status

The following sections cover the API changes in detail, starting with a brief explanation of how to update your plug-in to integrate with the latest OSMF release.

Updating Plug-ins

To update your plug-in to work with the OSMF Sprint 9 source, you need to do the following:

1. Modify your plug-in implementation so that you subclass PluginInfo rather than implement IPluginInfo.
2. Modify the "pluginInfo" property that the root of your plug-in exposes to be of type PluginInfo rather than IPluginInfo.
3. Remove the implementations of getNumMediaInfos, getMediaInfoAt, and isFrameworkVersionSupported. The base class (PluginInfo) provides suitable implementations, provided that you pass the version and MediaInfos in to the constructor.
4. Integrate your plug-in's code with the latest Sprint 9 API changes (detailed above).
5. For an example of a refactored plug-in, see the Akamai plug-in located at]http://opensource.adobe.com/svn/opensource/osmf/trunk/plugins/akamai/AkamaiBasicStreamingPlugin/].

API Changes

Project Structure

The MediaFramework project has been renamed to OSMF.

The MediaFrameworkTest project has been renamed to OSMFTest.

The MediaFrameworkIntegrationTest project has been renamed to OSMFIntegrationTest.

The MediaFrameworkAirTest project has been renamed to OSMFAIRTest.

org.osmf.display

MediaElementSprite and ScalableSprite have been removed. Use MediaPlayerSprite or MediaContainer instead.

ScaleModeUtils has been removed.

MediaPlayerSprite's element property has been renamed to mediaElement.

org.osmf.events

ViewEvent has been renamed to DisplayObjectEvent. The oldView and newView properties have been changed to oldDisplayObject and newDisplayObject, respectively.

MediaError now subclasses Error. The description and errorCode properties have been removed. Use message and errorID (from the base class), respectively.

GatewayChangeEvent has been renamed to ContainerChangeEvent.

SwitchEvent has been renamed to DynamicStreamEvent.

org.osmf.gateways

org.osmf.gateways has moved to org.osmf.containers

RegionGateway has been renamed to MediaContainer.

RegionGateway's validateContentNow method has been renamed to validateNow.

HTMLGateway has been renamed to HTMLMediaContainer.

org.osmf.layout

ILayoutRenderer has been renamed to LayoutRenderer, and made into a class.

org.osmf.media

MediaPlayer's API has undergone significant changes:

- Properties:
    - audible has change to hasAudio
    - bufferable has changed to canBuffer
    - element has changed to media
    - loadable has changed to canLoad
    - playable has changed to canPlay
    - seekable has changed to canSeek
    - switchable has changed to isDynamicStream
    - viewable has been removed
    - autoSwitch has changed to autoDynamicStreamSwitch
    - currentStreamIndex has changed to currentDynamicStreamIndex
    - maxStreamIndex has changed to maxAllowedDynamicStreamIndex
    - switchUnderway has changed to dynamicStreamSwitching
    - getBitrateForIndex has changed to getBitrateForDynamicStreamIndex
    - switchTo has changed to switchDynamicStreamIndex
    - pan has changed to audioPan
    - height has changed to mediaHeight
    - width has changed to mediaWidth
- Events:
    - audibleChange has changed to hasAudioChange
    - bufferableChange has changed to canBufferChange

- downloadableChange has been removed, use bytesLoadedChange and/or bytesTotalChange instead
- loadableChange has changed to canLoadChange
- pausableChange has changed to canPauseChange
- playableChange has changed to canPlayChange
- seekableChange has changed to canSeekChange
- switchableChange has changed to isDynamicStreamChange
- dimensionChange has changed to mediaSizeChange
- durationReached has changed to complete
- indicesChange has changed to numDynamicStreamsChange
- panChange has changed to audioPanChange
- spatialChange has been removed, use mediaSizeChange instead
- switchingChange has changed to dynamicStreamSwitchingChange
- loadStateChange has been removed, use mediaPlayerStateChange instead

IContainerGateway has been renamed to IMediaContainer.

IMediaContainer's addElement/removeElement/containsElement have been renamed to addMediaElement/removeMediaElement/containsMediaElement.

IMediaGateway has been removed. Use IMediaContainer instead.

MediaElement's gateway property has been renamed to container, and made read-only. MediaElement are added to IMediaContainers, but IMediaContainers are no longer set on MediaElements.

IMediaResource has been renamed to MediaResourceBase, and made into a class.

IURLResource has been removed. Use URLResource instead.

org.osmf.metadata

IFacet has been renamed to Facet, and made into a class.

org.osmf.plugin

IPluginInfo has been renamed to PluginInfo, and made into a class. Note that PluginInfo provides default implementations of all methods, and that subclasses should not need to override anything, but instead provide some params to the constructor.

PluginClassResource has been removed. Use PluginInfoResource instead.

org.osmf.traits

ContentProtectionTrait has been renamed to DRMTrait. Many of the properties, methods, and events have changed accordingly.

DynamicStreamTrait's maxIndex property has been renamed to maxAllowedIndex. The switchUnderway property has been renamed to switching.

ModifiableTimeTrait has been removed.

TimeTrait's durationReached event has been renamed to complete.

ViewTrait has been renamed to DisplayObjectTrait. The view property has been renamed to displayObject. The viewChange event has been renamed to displayObjectChange. The dimensionChange event has been renamed to mediaSizeChange.

MediaTraitType's CONTENT_PROTECTION type has been renamed to DRM.

MediaTraitType's VIEW type has been renamed to DISPLAY_OBJECT.

Much of the protected API for traits has undergone a nomenclature change:

- processXXX has changed to xxxChangeStart. For example, processPlayStateChange has been renamed to playStateChangeStart.
- postProcessXXX has changed to xxxChangeEnd. For example, postProcessPlayStateChange has been renamed to playStateChangeEnd.

Samples

MediaPlayer sample project has been renamed to MediaPlayerWrapper.

Getting Started Instructions

Using OSMF in Flex Builder 3

1. Open Flex Builder and choose File > New > Flex Project. Name the project. Select web application as application type. Click finish.
2. Select the project folder and choose File > Properties. In the Properties window, select Flex Compiler.
3. Verify the Flex SDK version. OSMF should work on any Flex SDK version, but the sample apps that we ship with may be created with Flex SDK 3.2 or above.
    - To download the SDK, go to www.adobe.com/go/flex3_sdk.
4. Set Require Flash Player version to 10.0.0. In order to install/update Flash player, please go to http://get.adobe.com/flashplayer
5. Do one of the following to use the OSMF source.
    - Copy OSMF.swc on to your local machine.
    - In your Properties window of your project, select the Flex Build Path -> Library Path tab.
    - Select Add SWC. Add OSMF and click Finish.

    OR
    - Copy the OSMF source to your local machine.
    - Open Flex Builder and choose Import > Existing projects into workspace.
    - Click browse to go to where the OSMF package is copied.
    - Select OSMF and Click Finish.

- In the project Properties window, select Flex Build Path and select the Library path tab.
- Click Add Project and browse to the folder that contains OSMF package.

Using OSMF in Flash CS4

1. Create your Flash project
2. File > Publish settings. Go to the Flash tab. Set the player version to Flash Player 10 and ActionScript3.0 and click on Settings.
3. Move to Library path tab and click on swc file icon.
4. Add OSMF.swc.
5. If you choose to publish an HTML file that detects the version of Flash Player, detect version 10.0.0 or later.

NOTE: To run OSMFTest (unit test suite) include the following files in your project library folder:

- FlexUnit.swc
- FlexUnitOptional.swc
- NetMocker.swc

You may also need to update the SWC file path when you import the OSMFTest project.

Sample Applications

The Sprint 9 drop of OSMF contains a number of sample applications which demonstrate new and old features. Each sample application is located in apps/samples/framework (for framework samples) or apps/samples/plugins (for plug-in samples). The root directory of each sample project holds a readme.txt file with installation and usage instructions.

Sample apps include:

AkamaiPluginSample:
Demonstrates the use of plug-ins. Integrates with other OSMF-provided plug-ins (SMIL, captioning, Akamai, etc.).

CaptioningSample:
Demonstrates loading the OSMF captioning plug-in and using an external captioning document to show captions over a video. Specifically, the sample app loads the OSMF captioning plug-in, places the URL location of a WC3 Timed Text DFXP file on the metadata of the video resource, and listens for the metadata TemporalFacet to be added to the VideoElement. When the TemporalFacet is added to the VideoElement, an event listener is added for events of type TemporalFacetEvent. In that event handler, the caption data is included in the event and the sample app renders the caption using the style information found in the caption object that was passed to the event listener.

ChromelessFlexPlayer:
This application is a Flex-based SWF that represents a chromeless SWF video player.

ChromelessPlayer:
This application is a pure AS3 SWF that represents a chromeless SWF video player.

CuePointSample:
Demonstrates temporal metadata within the OSMF framework.

DynamicStreamingSample:
Demonstrates dynamic streaming support in OSMF.

ExamplePlayer:
Demonstrates playback of a wide variety of media.

GGTrackingPlugin:
Demonstrates (along with the GGTrackingSample application) the use of a proxy plug-in to do non-invasive tracking of another MediaElement using GlanceGuide's tracking service.

GGTrackingSample:
Demonstrates the use of a proxy plug-in to do non-invasive tracking of another MediaElement using GlanceGuide's tracking service

HTMLMediaContainerSample:
Demonstrates how to use the HTML Bridging feature.

HelloWorld:

Demonstrates the simplest possible application that can be built with OSMF (see HelloWorld.as).

MetadataSample:

Demonstrates how to view and modify the metadata associated with media.

NestedMediaContainersSample and MediaContainerSample:

Demonstrates part of the framework's gateway feature which allows media elements to be routed.

OSMF Sample Player:

Defines an application that can be embedded on a web page in order to play back media. It contains a control bar that manages the various supported aspects of the media. It also uses the ChromeLibrary, which serves as a reference on implementing a user interface for an OSMF based player.

PluginSample:

Demonstrates how to load and work with plug-ins.

ReferenceSample:

Demonstrates how to have one piece of media reference another piece, even when the media are in separate plug-ins.

VASTSample:

The VASTSample sample application demonstrates the use of the VAST ActionScript library to retrieve a VAST document, parse it into a VAST object model, and generate one or more MediaElements that correspond to the playback instructions of that VAST document.

MASTSample:

The MASTSample sample application demonstrates the use of the MAST ActionScript plug-in to retrieve a MAST document, parse it into a MAST object model, and play a pre-roll ad before a video.

Note: Flex 3.2 or higher is recommended to run the Flex sample applications.

Compatibility

Flash Player 10 is required.

Known Issues

Subclip:

FM-23: Cannot define a subclip or seek in an RTMP audio only stream

Core Framework:

FM-238: When the video finishes playing the currentTime should be equal to duration.

Media Player:

FM-331: Unable to seek to a point within a temporal element

Preassigned Durations:

FM-250: Proxy image and video in parallel element sets only the image duration before playing the video.

Manifest File Format:

FM-269: Composite element doesn't play on manifest file.

Documentation

The API Reference and the OSMF Developer's Guide (browsable HTML versions) are available on the open source site here: http://opensource.adobe.com/wiki/display/osmf/Developer+Documentation

A zipped version of the API Reference is also available on the open source site the Download section here: http://opensource.adobe.com/wiki/display/osmf/Downloads