



360|Flex – San Jose 2010

Going Deep with OSMF

David Hassoun
(RealEyes Media)




Contact Info



DAVID HASSOUN
Email: david@realeyes.com
Blog: <http://david.realeyes.com>
RE|OPS:

Presentation:






ABOUT REALEYES

- FMS Solutions Partner & Value Added Reseller
- FMS Community Advisory Board Member
- Selected Clients:
 - American Express
 - Pfizer
 - Beatport
- Video Experience
 - Encoding
 - Delivery/Application Development
 - Consulting



WHAT WE'LL COVER

- **New World Order:** Intro to OSMF
- **Understanding the Machine:** The Parts
- **The City of Industry:** Factories
- **The Keystone:** Plugins
- **“Too big to fail”:** Conclusion

NWO w/OSMF 

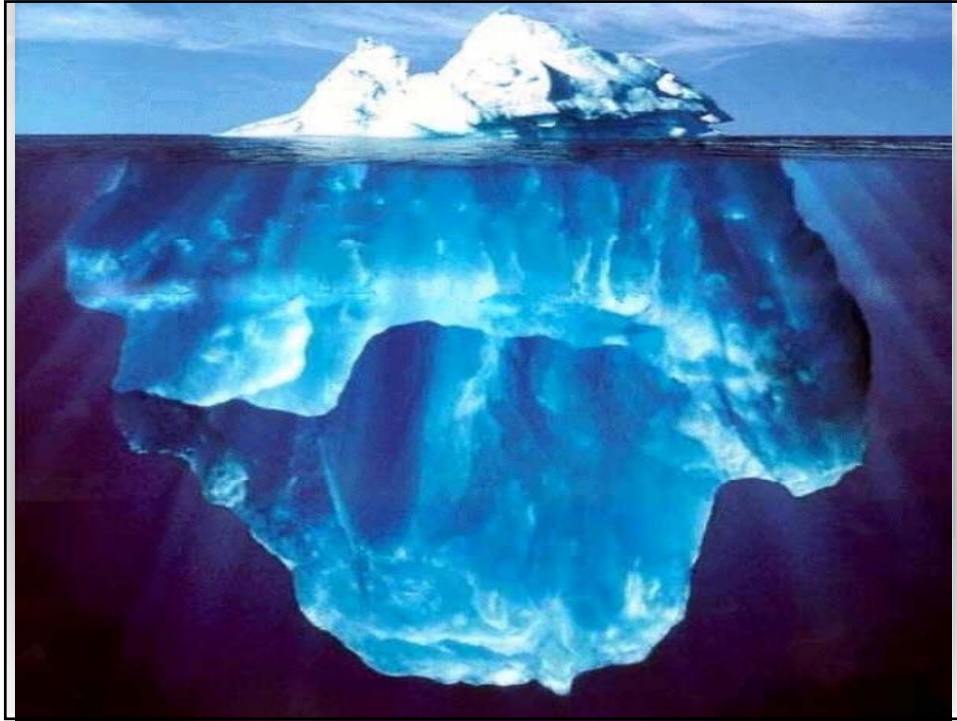
The Tip of the Iceberg

- URL – Parses the String into its parts
- URLResource – External Pointer
- **MediaElement** – Prepared Content
- **MediaPlayerSprite** – Playback Control & Display

NWO w/OSMF 

The Simplest Player Sample


- [DEMO]



(realeyes)

UNDERSTANDING THE MACHINE

Understanding how the parts work together

A 3D rendered metal gear with a central hole and several teeth, positioned to the right of the main text.




Under the hood

- **MediaResourceBase**
 - Base for the input of **MediaElement** – pointer to content
- **NetLoader**
 - Loading support
- **MediaElement**
 - Is part of the core of the OSMF
 - **VideoElement > LoadableElementBase > MediaElement**
- **MediaPlayer** (Note: used in **MediaPlayerSprite**)
 - Wraps up the complexity into simplicity - controller
- **MediaContainer**
 - Display Object – utilizes layout system



The Basic Parts Player Sample

- [DEMO]


The Machine 

Traits

- Describe the functionality of a MediaElement
- Change dynamically based on type and state
- Check to see if exist
- Traits dispatch their own events
- Use event handlers to know when added/removed
 - traitAdd/traitRemove on MediaElement's

“Traits make compositioning scalable. (Compositioning is the ability to temporally and spatially composite collections of media)” – ASDocs

- See: MediaElement, MediaTraitBase, MediaTraitTypes

The Machine 

Review Trait Management in the VideoElement - finishLoad

&

Traits Player

- [Demo]



MetaData

- NOT the media files Meta Data (client.onMetaData)
- Every Resource & MediaElement has MetaData
- Tells how to load or provides data for Plugins
- Metadata can be added, removed, updated and queried
- Uses namespaces to organize metadata
- Similar to traits, metadata can come and go
- Events can be used to track events in metadata changes ('facetAdd' & 'facetRemove')
- Metadata is aggregated for composite MediaElements



Working With Meta Data

- Managed via Facets
- Facets are an individual MetaData entries

- *addFacet()
- *removeFacet()
- *getFacet()

* Names are changing in sprint 10 (Facet term is going away)

The Machine

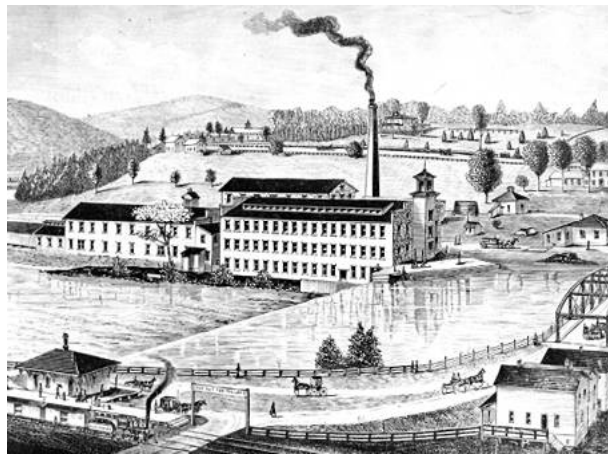



Show traits and facets in use

- [Demo]

THE CITY OF INDUSTRY

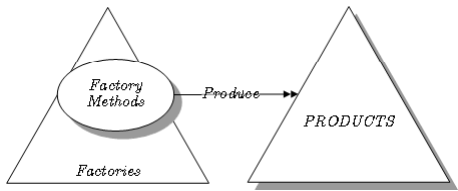
Enhancing the productivity and control with Factories




City of Industry 

What Are Factories

- **Factory Pattern:**
 - **Wiki** - deals with the problem of creating objects (products) without specifying the exact class of object that will be created



The diagram illustrates the Factory Pattern. It consists of two triangles. The left triangle is labeled 'Factories' at its base and contains an oval labeled 'Factory Methods'. An arrow labeled 'Produce' points from the 'Factory Methods' oval to the right triangle, which is labeled 'PRODUCTS' at its base.

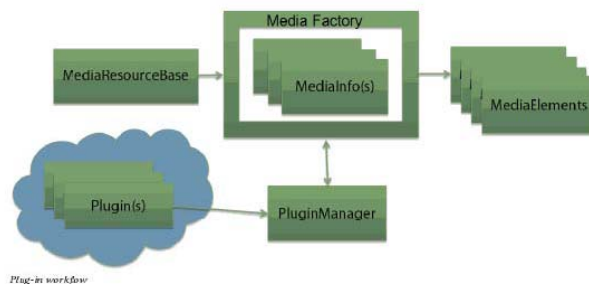
City of Industry 

NetConnectionFactory

- **Manages and generates NetConnections**
- **Enables connection sharing**
 - Can be passed to NetLoader
- **Can handle simultaneous requests**
- **Internally uses the NetNegotiator**
 - Port traversal
 - Low level NetConnection Management
 - NC.client linking as a NetClient

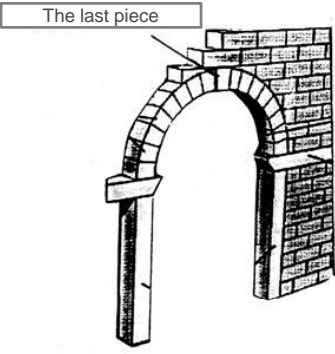
MediaFactory

- Generates MediaElement instances
- Creates the appropriate MediaElement from a registered MediaResources type
- Types registered by a MediaFactoryItem
- Used in conjunction also with Plugins (PluginManager)



Review Factory Sample

- [Demo]




(realeyes)

THE KEYSTONE

Plugins – completing the big picture with extensibility

The Keystone



(realeyes)

OSMF Plugins

- What are they
- How they work
 - Dynamic (SWF)
 - Static (AS Source or SWC)
- Types
 - Standard – creates media elements
 - Reference – given a reference to all created elements
 - Proxy – wrap a certain type and receive as created
 - wraps any media element that `canHandleResource` returns true for




Plugin Concepts

- **Provider Categories**
 - Advertising
 - Publishing
 - Analytics
 - CDN
- **Use Cases**
 - Closed Captioning
 - Custom Logging/Debugging
 - Custom Advertising
 - CDN Failover




Simplified Plugin Sequence

- A MediaFactory is created
- A PluginManager is created and passed a reference to the MediaFactory
- The PluginManager loads plugins via the loadPlugin() method
- The PluginManger gives the MediaFactory references to the loaded Plugins
- The MediaFactory creates new elements
- Depending on the plugin type - *Standard, Reference & Proxy*
 - The plugin is then used to create new MediaElements – Standard Plugin
 - The plugin is given a reference to the MediaElements – Reference & Proxy Plugins

The Keystone 

Creating a Plugin


- The basic workflow for creating a plugin is:
 - Create a PluginInfo Object
 - Add MediaFactoryItem(s) to PluginInfo.mediaFactoryItems
 - Expose the PluginInfo Object at the plugin root level

The Keystone 

Creating a Plugin Sample

- [DEMO]



Contact Info 

DAVID HASSOUN
Email: david@realeyes.com
Blog: <http://david.realeyes.com>

