Applying Flash to Java: Flex and OpenLaszlo

Dustin Marx
Raytheon Company
Gratitude to Colleagues

- I appreciate the direct and indirect assistance I received from colleagues:
  - William C. Jackson
  - Kent A. Jones
  - Varian S. Little
  - Michael Martin

- Discussions with these and other colleagues have led to new ideas for discussion points for this presentation.
Topics to Be Covered

- Advantages of Flash Player
- Introduction to Flex
- Flex and Java
- Introduction to OpenLaszlo
- OpenLaszlo and Java
- Flex / OpenLaszlo – Common and Conclusion

Presentation’s emphasis is on free and open source portions of Flex and OpenLaszlo.
Agenda

- Advantages of Flash Player
  - Abstraction of Browser Idiosyncrasies
  - Ubiquity
  - User Experience
  - Drawbacks and Disadvantages of Flash Player

- Introduction to Flex
- Flex and Java
- Introduction to OpenLaszlo
- OpenLaszlo and Java
- Flex / OpenLaszlo – Common and Conclusion
Why Flash for the Web?

- Advantages of Flash Player
  - Hides browser differences and idiosyncrasies
    - Hides them from developers and users
  - Highly ubiquitous web browser runtime environment
  - Provides user experience beyond traditional HTTP request-response
  - Visual effects and "richness" rivaling that of non-browser desktop applications
Flash: What Browser Differences?

- Flash runtime frees developers from specific browser concerns
  - Adobe’s Flash developers worry about browser idiosyncrasies
  - Developers don’t worry about differences in browsers’ DOM, ECMAScript, or CSS
  - Flash runtime abstraction of browser differences can benefit users as well
  - Virtually no more “Browser such-and-such required for this site” messages
The Ubiquitous Flash Player

- Adobe Flash Player market/browser penetration statistics available
  - All Versions of Flash Player (99% penetration)
  - Flash Player 9 (96%-98% penetration)

- Flash Player 9 appears to be on as many Internet-connected computers as JavaScript is enabled on
Flash Player 9 Availability

- Flash Player 9 as of June 2008:
  - Nearly 98% of browsers in “mature markets” and in “U.S./Canada”
  - Highest in Japan: nearly 99%
  - “Europe” and “Emerging Markets”: 96%

<table>
<thead>
<tr>
<th>Flash Version</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flash 9 Versions</td>
<td>~ 95%</td>
</tr>
<tr>
<td>Flash 10 Versions</td>
<td>~ 1%</td>
</tr>
<tr>
<td>Flash Version Prior to 9</td>
<td>~ 0.5%</td>
</tr>
<tr>
<td>Flash Version Not Set</td>
<td>~ 3.5%</td>
</tr>
</tbody>
</table>
Flash: Is This the Web?

- Flash frees users from limitations of traditional web applications
  - Request-Response is largely thing of the past
  - Ajax-like behavior was built-in from the beginning
  - Provides user experiences rivaling non-web experiences
Flash: Flashy Effects for Web

- Flash animation good enough for online games
- Flash video support used extensively
  - YouTube
- Flash effects are fluid and easy to apply
- Flash easily brings the Richness to RIA
Flash Criticisms

- Flash applications have had reduced
  - Search Engine Optimization (SEO)
  - Bookmarking and URL history support
- Other Flash Player criticisms
  - Not open (source or otherwise)
  - No significant alternative player
  - No 64-bit player
  - Loading performance
  - Only useful for games, movies, and annoying advertisements
Flash Limitations and Drawbacks

- Search Engine Optimization
  - Embed Flash .swf files in HTML files with searchable text
  - SWF Searchability initiative with Google and Yahoo!

- Bookmarking and URL History Support
  - “Deep Linking” with BrowserManager
    - mx.managers.BrowserManager
    - mx.managers.HistoryManager
Some of the Flash Player issues have been addressed:

- **Flash Open Screen Project**
  - Removes usage restrictions for SWF and FLV/F4V and removes device license fees

- **64-bit Player in Adobe Labs**
  - Some use 32-bit player on 32-bit browser

- **Flash 9 performance is much improved**
  - Flash 10 performance likely to be even better than Flash 9
More Flash Player Issues Addressed

- Alternatives to Adobe Flash Player
  - Alternatives to Flash Player can now be developed to “play SWF files”
    - Alternatives could previously only generate SWF files
    - See Adobe’s Open Screen Project FAQ

- Flash Player More Open
  - Open Screen Project (SWF/FLV/F4V), AMF, FlashCast, ActionScript/Tamarin
Agenda

- Advantages of Flash Player
- Introduction to Flex
  - What is Flex and What Isn’t Flex
  - MXML and ActionScript 3.0
  - Components, Compiling, Debugging
- Flex and Java
- Introduction to OpenLaszlo
- OpenLaszlo and Java
- Flex / OpenLaszlo – Common and Conclusion
What is Flex? Not Flash!

- Flex ≠ Flash
  - Flex and Flash are not the same thing
  - Flex requires Flash (Flash Player 9)
  - Flash does not require Flex
- Flex is a language and framework
  - Flex applications compile into .swf files
- Flash is the runtime environment
  - .swf files execute on Flash Player
So What is Flex?

- Web development framework
  - Includes an XML-based layout language (MXML)
  - Includes scripting language (ActionScript)
  - Includes compiler, debugger, tools
- Class library
  - Rich class library provided as part of Flex
  - Classes written in ActionScript
- Flex brings Flash to the Java developer
Flex to Flash

MXML and ActionScript Source Files

Flex Compiler

.mxml
and
.as

mxmlc

Flash Application

.swf
Flex: Based on Standards

- Flex is XML-based
  - MXML is a Flex-specific XML grammar
  - MXML files are well-formed XML
  - MXML files typically have `.mxml` suffix

- Flex is ECMAScript-based
  - ActionScript 3.0 is potential ECMAScript 4 implementation
  - MXML tags implemented in ActionScript
  - ActionScript used for business logic and event handling
Flex: Open Source and Free

- Flex 3 is FREE
  - Flex SDK, compiler, debugger are open source
  - Flex SDK, compiler, debugger have no license cost
  - BlazeDS is open source and has no license cost
Flex MXML

- Macromedia XML (Adobe now)
- XML-based presentation / layout language
- Editable with any text editor / IDE
- MXML is to ActionScript 3 as JavaServer Pages are to Java
  - MXML provides the layout
  - ActionScript provides the dynamic business logic
Simple MXML Example:
The MXML Code

```xml
<mx:Application
    xmlns:mx="http://www.adobe.com/2006/mxml"
    width="750" height="500">
    <mx:Panel id="MainPanel"
        title="Colorado Software Summit 2008 Flex MXML Example">
        <mx:Label id="mainLabel"
            text="This example is very simple."
            width="200" height="25" />
        <mx:Label text="Isn't it great being in Keystone?" />
    </mx:Panel>
</mx:Application>
```
Simple MXML Example: The Flash Presentation

This example is very simple.

Isn't it great being in Keystone?
ActionScript 3.0

- ECMAScript Implementation
  - Proposed Edition 4 implementation
  - ECMA-262 Edition 4 future less certain due to Edition 3.1/Harmony consensus

- Not Your Older Sibling’s JavaScript!
  - Class-based object-oriented features
  - Static typing

- Works the same across multiple browsers
  - Thanks to the Flash Player runtime
Simple ActionScript Example:
The Code

```actionscript
public function showHmi():void
{
    const mainPanel:Panel = new Panel();
    mainPanel.title = "Colorado Software Summit 2008 ActionScript Example";
    const mainLabel:Label = new Label();
    mainLabel.text = "This example is very simple.";
    mainLabel.width = 200;
    mainLabel.height = 25;
    mainPanel.addChild( mainLabel );
    const anotherLabel:Label = new Label();
    anotherLabel.text = "It is great being in Keystone!";
    mainPanel.addChild( anotherLabel );
    this.addChild(mainPanel);
}
```
width="750" height="500"
applicationComplete="showHmi()">

<mx:Script>
<![CDATA[
import mx.containers.Panel;
import mx.controls.Label;

... ActionScript 3 code from previous slide ...
]]>
</mx:Script>
</mx:Application>
Simple ActionScript Example:
The Flash Presentation

This example is very simple.

It is great being in Keystone!
ActionScript 3.0
Syntax Characteristics

- Class-based Object-oriented
  - Packages, interfaces, inheritance, objects, and methods
- Extensive XML Support
  - E4X (ECMAScript for XML) Support
- Large Class Library
- Access to the Flash Player
- Many Java-like features and syntax
ActionScript Class Library

- Flex provides feature-rich ActionScript class library similar to the Java SE class library
  - Core ECMAScript ActionScript Classes and Global Functions
    - Top-level package
    - Core language features, data types, functions
  - Flash Player 9 Action Script Classes
    - In flash.* packages
    - Allow direct application of Flash-specific APIs
  - Flex-specific Action Script Classes
    - In mx.* packages
    - ActionScript classes that implement MXML tags
Some Core ActionScript Classes and Global Functions

Classes
- Array
- Boolean
- Date
- Error
- Function
- Math
- Number
- String
- XML
- XMLList

Global Functions
- int
- isFinite
- isNaN
- Number
- parseFloat
- parseInt
- String
- trace
- XML
- XMLList
Flex: MXML and ActionScript

- MXML typically used for presentation layout
- ActionScript typically used for event handling and non-presentation business logic
- Most MXML Tags are implemented as ActionScript
Integrating MXML and ActionScript

- ActionScript can be embedded within MXML
  - Must specify `CDATA` when ActionScript syntax includes characters that will break XML parsing
  - ActionScript can be embedded in MXML within the `<mx:Script>` tags or within MXML tag event handler attributes

- ActionScript can be referenced from MXML file using `import` and `include` statements
Flex Components: Some of Flex’s Standard Components

- **General Controls**
  - Alert
  - ColorPicker
  - ComboBox
  - DataGrid
  - ProgressBar
  - Spacer

- **Button Controls**
  - Button
  - Checkbox
  - LinkBar
  - PopUpButton
  - RadioButton

- **Date Controls**
  - DateChooser
  - DateField

- **Text Controls**
  - Label
  - RichTextEditor
  - TextArea
  - TextInput

- **Containers**
  - Accordion
  - HDividedBox
  - VDividedBox
  - TitleWindow
Flex Component Example: RichTextEditor Used in Code

```xml
    width="900" height="500">
  <mx:RichTextEditor
      id="richTextEditor"
      title="Colorado Software Summit 2008 Rich Text Editor Example"
      height="85%" width="75%"
      text="Type whatever you like here."
      toolTip="Rich Text Editor Tool Tip"
      label="Rich Text Editor Label" />

  <mx:TextArea id="rteText" width="90%" height="15%" />

  <mx:HBox>
    <mx:Button label="Show Plain Text"
        click="rteText.text=richTextEditor.text;" />
    <mx:Button label="Show HTML Markup"
        click="rteText.text=richTextEditor.htmlText;" />
  </mx:HBox>
</mx:Application>
```
Flex Component Example: RichTextEditor Displayed
Flex Component Example: DataGrid Source Code

```xml
<mx:DataGrid id="dataGrid"
    width="100%" height="100%"
    rowCount="5" dataProvider="{presenters}">
    <mx:columns>
        <mx:DataGridColumn dataField="lastname"
            headerText="Last Name" width="75" />
        <mx:DataGridColumn dataField="firstname"
            headerText="First Name" width="75" />
        <mx:DataGridColumn dataField="organization"
            headerText="Organization" width="100" />
        <mx:DataGridColumn dataField="url"
            headerText="CSS URL" width="200" />
    </mx:columns>
</mx:DataGrid>
```

The dataProvider "presenters" is set up elsewhere in the MXML file.
Flex Component Example: DataGrid on Display

- Flex DataGrid is powerful and flexible
- Easily sortable/editable/selectable
- Populated via data binding with XML-based datasource
Third-Party Flex Components

- More third-party Flex components seem to appear each day
  - flexlib: Open Source Flex Component Library
  - ASDIA: ActionScript Diagram Library
  - Many more ...
    - FlexBox lists many Flex components
- Create your own Flex components!
Flex Property Binding

- A favorite Flex feature
  - Its simplicity belies its power
- Data in one object tied to data in another object
  - Updates to an object are reflected in other object
- Specify Flex property binding:
  - Use curly braces ({{}}) [MXML]
  - `<mx:Binding>` tag [MXML]
  - `BindingUtils` object [ActionScript]
- Use Flex metadata annotation [Bindable] to denote bindable objects
Flex Property Binding
Example: ColorPicker/Source

<!-- This example demonstrates the Flex ColorPicker component and Flex's property data binding mechanism. The selectedColor from the ColorPicker component impacts the background color. -->

<mx:VBox id="mainPanel" width="100%" height="100%"
  backgroundColor="{colorBackground.selectedColor}">
  <mx:Label id="mainAppTitle"
    textAlign="center"
    fontWeight="bold"
    text="Colorado Software Summit 2008 Flex ColorPicker Example" />
  <mx:HBox>
    <mx:Label text="Background Color" />
    <mx:ColorPicker id="colorBackground"
      showTextField="true"
      selectedColor="#DDDDDD" />
  </mx:HBox>
</mx:VBox>
Flex Property Binding
Example: Output
Flex Events and Event Handling

- Flex event handling is similar to that in JavaScript/DOM
- Event handlers specified as MXML tag attributes
- Add event handlers in ActionScript with `addEventListener()`
Compiling Flex Applications

- **mxmlc** from command line
  - Approach I’m using in this presentation
- Ant using `<exec>` task
- Ant using Flex Ant Tasks
- Can also invoke mxmlc compiler from Java
  - `java -jar ..;/lib/mxmlc.jar` ...
- FlexBuilder IDE
Flex SDK Debugger

- Flex Debugger (**fdb**) is freely available with Flex SDK
  - Use `trace()` function in code to log info to debugger
  - Compile Flex code with `-debug=true`
  - Run `fdb` from console
  - Use Flash 9 Debug Player in web browser
- FlexBuilder includes debugger support
FlexBuilder / Flex Charting

- FlexBuilder is an Eclipse-based IDE for Flex development lifecycle
  - Not included with free Flex SDK
  - Many of Flex 3’s enhancements focus on Flex Builder 3
  - Two versions: Eclipse Plug-in (works with Java) or Standalone Eclipse-based IDE

- Flex Charting provides dynamic and appealing charts based on data
  - Not included with free Flex SDK
Agenda

- Advantages of Flash Player
- Introduction to Flex
- Flex and Java
  - HTTPService
  - WebService
  - BlazeDS
- Introduction to OpenLaszlo
- OpenLaszlo and Java
- Flex and OpenLaszlo - Conclusion
Flex/Server Communication

- Two predominant out-of-the-box methods for Flex to communicate with Java EE back-ends are:
  - **HttpService** (traditional HTTP request-response)
  - **WebService** (SOAP-based Web Services)
- BlazeDS-provided additional options:
  - Web Messaging (HTTP publish-subscribe with JMS, ColdFusion, and/or other Flash/Ajax client)
  - Remoting with AMF (access server-side objects from Flash client-tier as if they were hosted there)
Proxy-less Flash Remote Communication

- Flash application can either access remote sites or the client’s machine, but not both
  - `-use-network` in Flex (`true` for remote; `false` for local)
- Use `crossdomain.xml` file on server to allow remote Flash clients to connect
- Applies to Flex-generated Flash applications talking to server without proxy
Flex/Server Communication: General Mechanisms

- **HTTPService** and **WebService**
  - Can be used with any supporting server-side technology
  - Tend to use XML-formatted data

- **HTTPService** might connect to Java HttpServlet or other server mechanism supporting HTTP request/response

- **WebService** might connect to Java-based Web Service or other type of SOAP-based Web Service
<mx:HTTPService id="requestPresenters"
url="http://localhost:8080/CssExamples/javawebserver?action=listPresenters"
method="POST"
resultFormat="e4x"
result="handleResult(event)"
fault="handleFault(event)"/>
Flex WebService Syntax Example

```xml
<mx:WebService
    id="someServiceId"
    wsdl="http://aHost:aPort/aSrvc/endpnt?WSDL"
    fault="handleFault(event)"
    result="handleWsResult(event)" />
```
Flex/Server Communication: BlazeDS

- Java Remoting and Web Messaging
  - Enhances and provides alternatives to HTTPService and WebService for communication with Java-based server
  - Subset of Adobe LiveCycle Data Services ES
- AMF binary data transfer format
  - Better performance than uncompressed text/XML
- Freely available open source (LGPL v3)
More BlazeDS

- Flex proxy server support for HTTPService and WebService
  - Set `useProxy` attribute to `true`
  - Authentication, logging, and more
- New RPC mechanism: `RemoteObject`
  - ActionScript/JEE transparent integration
- Adds publish/subscribe messaging
  - JMS adapter available
- Ajax support
- More complete REST support
GraniteDS

- Granite Data Services
  - Free, Open Source (LPGL)
  - Alternative to Adobe LiveCycle Product
  - Similar to Adobe-provided BlazeDS
    - GraniteDS is not provided by Adobe
  - COMET-like functionality (Gravity)
  - Supports Spring Framework (Acegi Security), EJB3, JBoss SEAM, Guice, etc.
Flex Frameworks

- Cairngorm
  - Adobe-provided (Adobe Consulting)
  - Design pattern oriented and open source
- PureMVC
  - ActionScript 3 version is Reference Implementation
  - Proposed or in-work implementations for Java, Ruby, C#, Perl, Python, PHP
- Mate, Flest, and several others
Flex and AIR

- AIR is Adobe Integrated Runtime
  - Combines advantages of web and desktop
    - Works across multiple operating systems
      - But Linux support still considered alpha
    - Local data access
  - Build with web languages and technologies
    - Flex / Tamarin / Flash
    - Ajax / HTML / JavaScript / XML
  - Runs on desktop
    - Flash technologies for the desktop
    - RIA for the desktop
What’s Coming in Flex 4?

- Flex 4 / Gumbo
  - Improvements for Designers
    - Easier customization
    - Better tool support
  - Improvements for Developers
    - Faster compiler (up to 5 times faster?)
    - Two-way data binding
  - Leverage New Features of Flash 10
Agenda

- Advantages of Flash Player
- Introduction to Flex
- Flex and Java
- Introduction to OpenLaszlo
  - What is OpenLaszlo?
  - LZX and JavaScript
  - Components, Compiling, Debugging
- OpenLaszlo and Java
- Flex / OpenLaszlo – Common and Conclusion
OpenLaszlo: Based on Standards

- OpenLaszlo 4 is XML-based
  - LZX is OpenLaszlo-specific XML grammar
  - LZX files are well-formed XML
  - LZX files typically have .lzx extension
  - XML data access with XPath syntax

- OpenLaszlo 4 is ECMAScript-based
  - OpenLaszlo uses subset of JavaScript, an ECMAScript Edition 3 implementation
  - JavaScript typically used for event handling and non-view business logic
OpenLaszlo: Flash 9 Not Required

- Flex 2 and Flex 3 require Flash Player 9
- OpenLaszlo architected to deploy on different runtime environments:
  - Flash Player 7
  - Flash Player 8
  - Flash Player 9 (under work)
  - DHTML
  - Some classes cannot compile into all runtimes
- Intriguing Runtime Deployment Possibilities:
  - What if OpenLaszlo source code could be compiled into Silverlight or JavaFX?
LZX

- LZX is Laszlo XML
- LZX is OpenLaszlo’s XML-based presentation layout language
- LZX can be edited with any text editor
  - XML tools or Integrated Development Tools (IDEs) with XML support preferred
- LZX is to OpenLaszlo JavaScript as JavaServer Pages are to Java
  - LZX provides layout via tags and JavaScript provides dynamic support
Simple LZX Code Example

```xml
<canvas width="750" height="500">
  <window id="MainWindow"
title="Colorado Software Summit 2008 OpenLaszlo Example">
    <simplelayout axis="y" spacing="10" />
    <statictext id="mainLabel"
      width="200"
      height="25">This is a very simple example.
    </statictext>
    <statictext>A week in Keystone, Colorado!
    </statictext>
  </window>
</canvas>

LZX does not have namespaces!
Simple LZX Example Display

Flash Player 8 Display

This is a very simple example.

A week in Keystone, Colorado!
Simple LZX Example Display

DHTML Display

This is a very simple example.

A week in Keystone, Colorado!
OpenLaszlo’s JavaScript

- OpenLaszlo’s scripting language based on ECMAScript (ECMA-262)
  - Browser-specific objects not available
  - Works same across web browsers
- Use for non-presentation business logic, event handling, etc.
  - Can be used for presentation
OpenLaszlo Constraints

- One attribute’s value depending on one or more other attributes’ values
- Specify constraints multiple ways:
  - `${}` in LZX tags
  - `applyConstraint()` method in script
  - `LzDelegate` in script
OpenLaszlo Events

- OpenLaszlo event handling is similar to Flex and JavaScript event handling
- All OpenLaszlo attributes have implicit “on” event
  - Any change to attribute’s value triggers this event
- Event Handlers
  - CSS-style event handlers in LZX nodes
  - `<handler></handler>`
LZX and JavaScript

- Some LZX XML elements have underlying scripting classes (prefix \texttt{Lz})
- Script can be embedded in LZX XML:
  - Inside event attributes
  - Within \texttt{<script>}/\texttt{<script>} tags
  - Within \texttt{<method>}/\texttt{<method>} tags
  - Within \texttt{<handler>}/\texttt{<handler>} tags
  - Separate file
    - \texttt{<script src="someFile.js"/>}
Select Examples of OpenLaszlo Components

- alert
- button
- checkbox
- combobox
- datepicker
- edittext
- form
- grid
- menubar
- radiobutton
- scrollbar
- slider
- statictext
- tab
- tree
- vscrollbar
- window
- hbox
- vbox
OpenLaszlo Component
Example: DatePicker

```xml
<datepicker showingdate="new Date(2008, 9, 19)"
 earliestdate="new Date(2008, 0, 1)"
 latestdate="new Date(2010, 11, 31)"
 selecteddate="new Date(2008, 9, 24)">

<handler name="onselecteddate">
  if( this.selecteddate != null ) {
    dateDisplay.month.setText(        this.selecteddate.getMonth() + 1 );
    dateDisplay.date.setText(        this.selecteddate.getDate() );
    dateDisplay.year.setText(        this.selecteddate.getFullYear() );
  }
</handler>

</datepicker>
```
OpenLaszlo Component

Example: DatePicker

Flash Player 8
OpenLaszlo Component
Example: DatePicker

DHTML
<grid datapath="presentersData:/presenters"
    contentdatapath="presenter"
    shownitems="5" width="${parent.width}">
    <gridcolumn width="${parent.width*.15}">Last Name
    <text datapath="lastname/text()" />
    </gridcolumn>
    <gridcolumn width="${parent.width*.15}">First Name
    <text datapath="firstname/text()" />
    </gridcolumn>
    <gridcolumn width="${parent.width*.2}">Organization
    <text datapath="organization/text()" />
    </gridcolumn>
    <gridcolumn width="${parent.width*.5}">CSS URL
    <text datapath="url/text()" />
    </gridcolumn>
</grid>
### OpenLaszlo Component: Grid on Display

#### SWF8 Output

<table>
<thead>
<tr>
<th>Last Name</th>
<th>First Name</th>
<th>Organization</th>
<th>CSS URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Johnsson</td>
<td>Dan</td>
<td>OmegaPoint AB</td>
<td><a href="http://softwaresummit.org/2008/speakers/johnsson.htm">http://softwaresummit.org/2008/speakers/johnsson.htm</a></td>
</tr>
<tr>
<td>Kaplan-Moss</td>
<td>Jacob</td>
<td>Whiskey Media</td>
<td><a href="http://softwaresummit.org/2008/speakers/kaplan-moss.htm">http://softwaresummit.org/2008/speakers/kaplan-moss.htm</a></td>
</tr>
<tr>
<td>Lan</td>
<td>Ikai</td>
<td>LinkedIn</td>
<td><a href="http://softwaresummit.org/2008/speakers/lan.htm">http://softwaresummit.org/2008/speakers/lan.htm</a></td>
</tr>
<tr>
<td>Marx</td>
<td>Dustin</td>
<td>Raytheon Company</td>
<td><a href="http://softwaresummit.org/2008/speakers/marx.htm">http://softwaresummit.org/2008/speakers/marx.htm</a></td>
</tr>
<tr>
<td>Morrill</td>
<td>Dan</td>
<td>Google</td>
<td><a href="http://softwaresummit.org/2008/speakers/morrill.htm">http://softwaresummit.org/2008/speakers/morrill.htm</a></td>
</tr>
</tbody>
</table>

#### DHTML Output

<table>
<thead>
<tr>
<th>Last Name</th>
<th>First Name</th>
<th>Organization</th>
<th>CSS URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Johnsson</td>
<td>Dan</td>
<td>OmegaPoint AB</td>
<td><a href="http://softwaresummit.org/2008/speakers/johnsson.htm">http://softwaresummit.org/2008/speakers/johnsson.htm</a></td>
</tr>
<tr>
<td>Kaplan-Moss</td>
<td>Jacob</td>
<td>Whiskey Media</td>
<td><a href="http://softwaresummit.org/2008/speakers/kaplan-moss.htm">http://softwaresummit.org/2008/speakers/kaplan-moss.htm</a></td>
</tr>
<tr>
<td>Lan</td>
<td>Ikai</td>
<td>LinkedIn</td>
<td><a href="http://softwaresummit.org/2008/speakers/lan.htm">http://softwaresummit.org/2008/speakers/lan.htm</a></td>
</tr>
<tr>
<td>Marx</td>
<td>Dustin</td>
<td>Raytheon Company</td>
<td><a href="http://softwaresummit.org/2008/speakers/marx.htm">http://softwaresummit.org/2008/speakers/marx.htm</a></td>
</tr>
<tr>
<td>Morrill</td>
<td>Dan</td>
<td>Google</td>
<td><a href="http://softwaresummit.org/2008/speakers/morrill.htm">http://softwaresummit.org/2008/speakers/morrill.htm</a></td>
</tr>
</tbody>
</table>
OpenLaszlo Incubator

- Components not yet integrated into standard framework
- Contributed by community
- Often very useful
- Lack testing, review, or documentation of standard components

Select Components

- colorpicker
- draglib
- formlayout
- inspector
- richtext
- stepper
- tooltipmanager
- validators
- gradientview

*OpenLaszlo Incubator items covered explicitly in my blog.
Compiling OpenLaszlo

- Two steps in OpenLaszlo compilation
  - Compile LZX tags into JavaScript
  - Compile JavaScript into SWF7, SWF8, or DHTML
  - Developer sees both steps as single step
Compiling OpenLaszlo

- Three Compilation Approaches
  - Developer’s Console (in Web Browser)
    - Easily switch between SWF7, SWF8, and DHTML
    - Easily turn debugging on and off
    - Easily view source
  - URLs (in Web Browser)
    - Can copy the URL parameters applied by Developer’s Console directly to URL approach
  - Command-line lzc Compiler
Deployment Modes

- **Proxied**
  - Source compiled as needed and binary sent to web client
  - Required mode to use OpenLaszlo RPC

- **SOLO (Standalone OpenLaszlo Output)**
  - Precompiled code
  - AKA “Serverless”
  - Most common in production
  - Often has better performance
OpenLaszlo’s Debugger

- OpenLaszlo’s debugger is one of its most useful and underrated features
  - Easy-to-use with any of the following approaches
  - Add `&debug=true` to end of request URL
  - Check “Debug” checkbox in “Compile Options” window at bottom of web browser hosting OpenLaszlo application
  - Set `debug` attribute of `canvas` element to `true`
More on OpenLaszlo Debugger

- Debug is displayed in special window within OpenLaszlo application output in browser
- `Debug.write()` writes text to debugger
OpenLaszlo Request Types

- Handy during development
- Turned off during production
- Easy to use
  - Add appropriate request type to end of OpenLaszlo URL
OpenLaszlo Request Types

- **Examples:**
  - `lzt=source` displays source code
  - `debug=true` displays debug information
  - `lzt=serverinfo` displays OpenLaszlo server configuration
  - `lzt=stat` monitors server statistics
  - `lzt=log` displays current log file

- Simply apply at end of OpenLaszlo URL
- Do not apply in OpenLaszlo SOLO
Agenda

- Advantages of Flash Player
- Introduction to Flex
- Flex and Java
- Introduction to OpenLaszlo
- OpenLaszlo and Java
  - HTTP Data with dataset
  - Ajax with XMLHttpRequest
  - Proxied RPC communication modes
- Flex / OpenLaszlo – Common and Conclusion
OpenLaszlo and Server Communication: All

- **HTTP Data**
  - Dataset element with `type` attribute set to `http`
    - `<dataset type="http" src= ...`

- **Ajax API**
  - `XMLHttpRequest()`

- These methods work in SOLO and proxied modes
OpenLaszlo/Server
Communication: Proxied

- OpenLaszlo RPC only works in Proxied Mode
  - JavaRPC – Access server-side Java objects from client
  - XML-RPC – Remote operations using XML over HTTP
  - SOAP – Access SOAP-based web services
HTTP Data (dataset)

```xml
<dataset
    name="allEmployees"
    request="true"
>
```
Notes About OpenLaszlo

- Change Laszlo-provided `web.xml` from servlet 2.2 DTD to 2.5 XML Schema if using Java EE 5 resource injection
- Must use `crossdomain.xml` file with Flash for remote connectivity
  - SOLO mode
More OpenLaszlo Notes

- LZX elements can have both a **name** and an **id** attribute
  - **id** attribute is globally unique
  - **name** is locally referenced (relative)

- OpenLaszlo’s data lookup XPath-based

- OpenLaszlo development work with DHTML has helped Ajax/DHTML community
Flex in OpenLaszlo

- OpenLaszlo 4.2 includes SWF9 with Flex compiler
- ActionScript will be supported for OpenLaszlo SWF9 deployments
- Enables OpenLaszlo to benefit from Flash Player 9
OpenLaszlo and DHTML

- OpenLaszlo applications can be executed in SWF7, SWF8, and DHTML
- Not all source code features work in all environments
  - Differentiate code based on runtime with `<when runtime=""">` tag
- Flash files (.swf) may run directly
- DHTML (.js) files require wrapper
OpenLaszlo Features: Flash But Not DHTML

- Several features for Flash runtimes that are not for DHTML runtime
  - Many asset-related features
  - Embedded fonts
  - richinputtext component
  - Flash Player-specific features
    - Flash Player ActionScript libraries
    - Example: Flash Cookies

- Can mix DHTML and Flash for best of both worlds
OpenLaszlo’s Future?

- Full SWF 9 (Flash Player 9) Support
- Laszlo-sponsored IDE for OpenLaszlo
- OpenLaszlo for Java ME
  - Project Orbit
- Potentially New Runtimes? (lots of speculation here)
  - Silverlight?
  - JavaFX?
Agenda

- Advantages of Flash Player
- Introduction to Flex
- Flex and Java
- Introduction to OpenLaszlo
- OpenLaszlo and Java
- Flex / OpenLaszlo – Common and Conclusion
  - Flash Application Integration with HTML
  - IDE / Editor Support and Charting
  - Differences and Similarities
- Conclusion
Integrating Flash with HTML

- Flex-generated and OpenLaszlo-generated SWF often hosted in HTML “wrapper” pages
  - Flash components part of larger HTML-based application
  - Enhances searchability
  - Improved interaction with browser via JavaScript (or JScript)
Methods for Generating SWF Wrappers

- Custom-written HTML/Java­Script
- Framework-provided templates
- On-the-fly Generation
  - FlexBuilder/IIS-generated for Flex
  - OpenLaszlo-Generated for OpenLaszlo
- Flex html­-wrapper Ant task
- swfobject 2
Integrating SWF/HTML: Custom Wrapper

- Use `<object>` tag for Microsoft Internet Explorer / ActiveX
- Use `<embed>` tag for non-IE browsers

Simplest Example

```xml
<object width="1000" height="800">
<param name="FlashApp" value="VeryBasicMXML.swf">
<embed src="VeryBasicMXML.swf" width="1000" height="800"></embed>
</object>
```
Integrating SWF/HTML: Flex-Provided Templates

- Templates provided with Flex SDK
  - `flex_install_dir/templates`
- Several different templates provided
  - Two versions of each (with and without deep linking/history)
    - client-side-detection
    - express-installation
    - no-player-detection
  - Customize by hand and with `{}` tokens
Integrating SWF/HTML: html-wrapper Ant Task

- Requires Flex and Ant
- Generates artifacts for deployment of Flex applications
  - HTML, JavaScript, CSS, etc.
  - Generates same types of templates available in Flex template directory
    - But with settings provided in Ant task
swfobject

- Popular mechanism for including Flash in HTML
  - Simple (no alteration required)
  - Standards-oriented
    - Likely to become Flash embedding standard
  - Small (JavaScript) footprint
  - Best of breed
    - SWFObject 2 based on SWFObject 1.5 (FlashObject/) and UFO
  - Formerly known as SWFFix
**Example: swfobject and SWF-to-SWF Communication**

**Step 1**: Access SWFObject *swfobject.js* File and Register SWFs with it.

```html
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
  <!-- Template adapted from http://code.google.com/p/swfobject/wiki/documentation. -->
  <head>
    <title>CSS 2008 Flex-to-Flex Example: Driver</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <script type="text/javascript" src="../../swfobject.js"></script>
    <script type="text/javascript">
      swfobject.registerObject(          "sourceSWF", "9.0.0", "expressInstall.swf");
      swfobject.registerObject(          "targetSWF", "9.0.0", "expressInstall.swf");
    </script>
  </head>
</html>
```
Example: swfobject and SWF-to-SWF Communication

**Step 2**: Embed SWF in HTML (shown for sender, but similar for receiver).

```html
<div>
  <object id="sourceSWF"
    classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
    width="100%" height="250">
    <param name="movie" value="DriverDataGrid.swf" />
    <!--[if !IE]-->
    <object type="application/x-shockwave-flash"
      data="DriverDataGrid.swf"
      width="100%" height="250">
    <!--<![endif]-->
    </object>
    <!--[if !IE]-->
    <p>Unable to find or render the DriverDataGrid.swf Flash Application</p>
    <!--[if !IE]-->
  </object>
  <!--[endif]-->
</div>
```
Example: swfobject and SWF-to-SWF Communication

**Step 3:** Key pieces of “sending” MXML code.

```javascript
private const connection:LocalConnection =
    new LocalConnection();

public function setUpSenderConnection():void {
    connection.addEventListener(
        StatusEvent.STATUS, onConnectionSendStatus);  }

private const selectedPresenterName:String =
    dataGrid.selectedItem.firstname +
    dataGrid.selectedItem.lastname +
    " of " + dataGrid.selectedItem.organization;

function callOtherFlashApp():void {
    connection.send("cssConnection", "displaySelectedPresenter",
        selectedPresenterName);  }
```
Example: swfobject and SWF-to-SWF Communication

**Step 4:** Key pieces of “receiving” MXML code.

```actionscript
private const connection:LocalConnection = new LocalConnection();

public function setUpReceiverConnection():void {
    connection.client = this;
    connection.connect("cssConnection");
}

public function displaySelectedPresenter(
    selectedPresenterString:String):void {
    try
    {
        targetText.text = selectedPresenterString;
    }
    catch (error:ArgumentError)
    {
        Alert.show("Error trying to connect to LocalConnection.");
    }
}
```
Example: swfobject and SWF-to-SWF Communication

CSS 2008 Flash-to-Flash Example: Driver Page

<table>
<thead>
<tr>
<th>Last Name</th>
<th>First Name</th>
<th>Organization</th>
<th>CSS URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Johnson</td>
<td>Dan Bergh</td>
<td>OmegaPoint AB</td>
<td><a href="http://softwaresummit.org/2008/speakers/johnson.htm">http://softwaresummit.org/2008/speakers/johnson.htm</a></td>
</tr>
<tr>
<td>Kaplan-Moss</td>
<td>Jacob</td>
<td>Whiskey Media</td>
<td><a href="http://softwaresummit.org/2008/speakers/kaplan-moss.htm">http://softwaresummit.org/2008/speakers/kaplan-moss.htm</a></td>
</tr>
<tr>
<td>Lan</td>
<td>Ikai</td>
<td>LinkedIn</td>
<td><a href="http://softwaresummit.org/2008/speakers/lan.htm">http://softwaresummit.org/2008/speakers/lan.htm</a></td>
</tr>
<tr>
<td>Marx</td>
<td>Dustin</td>
<td>Raytheon Company</td>
<td><a href="http://softwaresummit.org/2008/speakers/marc.htm">http://softwaresummit.org/2008/speakers/marc.htm</a></td>
</tr>
<tr>
<td>Morrill</td>
<td>Dan</td>
<td>Google</td>
<td><a href="http://softwaresummit.org/2008/speakers/morrill.htm">http://softwaresummit.org/2008/speakers/morrill.htm</a></td>
</tr>
</tbody>
</table>

Selected Presenter: Dustin Marx of Raytheon Company
Flex and OpenLaszlo: IDE Support

### Flex
- **FlexBuilder**
  - Not part of free SDK
  - Eclipse-based
- **Flash Develop**
- **FlexBean**
  - Netscape 6.1 plugin

### OpenLaszlo
- **IDE for Laszlo**
  - Eclipse Plug-in
  - “Archived”
- **NetBeans 6.x Plug-in**
  - Relatively new
- **Laszlo IDE**
  - Laszlo Systems
  - Will be Eclipse-based
  - Unlikely to be open source
Flex and OpenLaszlo Editors

- Any text editor or IDE can be used with Flex and OpenLaszlo
  - XML and JavaScript editors are especially effective
  - Several useful choices:
    - NetBeans / Eclipse / JDeveloper / IDEA
    - Spket IDE
    - Aptana
    - JEdit
    - Many more ...
Flex and OpenLaszlo: Charting

- Charts
  - Flex Charts are sophisticated but not free for most realistic production applications
  - OpenLaszlo Charts are still in beta
  - Alternatives (third-party charting):
    - JFreeChart / Eastwood Chart Servlet
    - Open Flash Chart
    - amCharts
    - Fusion Charts Free
    - Google Charts
## Flex 3 and OpenLaszlo 4: Commonalities/Differences

<table>
<thead>
<tr>
<th>Framework Characteristic</th>
<th>Flex 2 / Flex 3</th>
<th>OpenLaszlo 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Sponsor</td>
<td>Adobe</td>
<td>Laszlo Systems</td>
</tr>
<tr>
<td>Runtime</td>
<td>Flash 9 only</td>
<td>Flash 7/8/9* &amp; DHTML</td>
</tr>
<tr>
<td>XML Grammar</td>
<td>MXML</td>
<td>LZX</td>
</tr>
<tr>
<td>Scripting</td>
<td>ActionScript 3.0 (ECMA Edition 4)</td>
<td>JavaScript (ECMA Edition 3)</td>
</tr>
<tr>
<td>Charting</td>
<td>Not in free SDK</td>
<td>Still in beta</td>
</tr>
<tr>
<td>Documentation</td>
<td>Solid (blogs useful)</td>
<td>Solid (blogs useful)</td>
</tr>
<tr>
<td>Books</td>
<td>10+</td>
<td>2</td>
</tr>
</tbody>
</table>

*Flash 9 still in alpha for OpenLaszlo.*
What I Like Most About Flex and OpenLaszlo

<table>
<thead>
<tr>
<th>Flex 3</th>
<th>OpenLaszlo 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>No more browser-specific frustrations.</td>
<td></td>
</tr>
<tr>
<td>Simplified development of interactive apps.</td>
<td></td>
</tr>
<tr>
<td>Clean separation (loose coupling) with server.</td>
<td></td>
</tr>
<tr>
<td>Rich component sets (provided and 3\textsuperscript{rd} party).</td>
<td></td>
</tr>
<tr>
<td>Data Binding</td>
<td>Constraints</td>
</tr>
<tr>
<td>E4X Support</td>
<td>XPath-based</td>
</tr>
<tr>
<td>Large Community</td>
<td>Debugger</td>
</tr>
<tr>
<td>Leverages Latest Flash</td>
<td>Multiple Runtimes</td>
</tr>
</tbody>
</table>
Alternative to Flex and OpenLaszlo: Java Applets

- Is the Java Applet Back?
  - Flash-based development has some similarities to applet-based development
    - Web Browser Runtime Plug-ins
    - Extremely Rich User Experience
    - Easy Development Experience without Browser Specific Frustrations

- Java SE 6 Update 10 now released
  - Significantly improves applet experience for user
Alternatives to Flex and OpenLaszlo

- Extensible Ajax Platform (XAP)
  - XML Declarative Framework like Flex and OpenLaszlo
- Google Web Toolkit (GWT)
- JavaServer Faces (JSF)
- Curl
  - Own Runtime Environment similar to Flash
- Pivot
More Alternatives: JavaScript Libraries

- jQuery
- Dojo
- Prototype
- SproutCore
  - Ruby
- Spry (Adobe)

See http://javascriptlibraries.com/ for many, many, many, many more
Flex and OpenLaszlo offer many advantages for the RIA web developer

- Ubiquitous runtime in Flash Player
  - DHTML option as well for OpenLaszlo
- Write once for all web browsers
  - Flash hides browser idiosyncrasies
  - OpenLaszlo handles browser idiosyncrasies for DHTML
- Easy access to rich and highly fluid component sets
Flex and OpenLaszlo for the Java Developer

- Flex and OpenLaszlo offer the Java developer many advantages:
  - Easy front-end integration with JEE back-ends via HTTP
  - Java-inspired syntax and constructs
  - Both frameworks built with Java in mind
    - OpenLaszlo server is Java-based
    - Flex BlazeDS is Java-centric
    - Support other languages, but Java was first thought
Java EE + Flex/OpenLaszlo: A Compelling Combination

- Flex or OpenLaszlo provide:
  - Users with rich, highly interactive clients in a web browser
  - Developers with a single API for their web development

- Java EE provides:
  - Enterprise-ready back-ends
    - Management with JMX, Full and current web services support, JEE web server advantages
  - Large base of existing back-end services
Applying Flash to Java: Flex and OpenLaszlo

Background Material
Flex 3 Documentation

- Flex Documentation

- Flex 3 Language Reference
  - Visual examples in Javadoc-like API HTML pages
  - Includes source code examples in API comments
Flex 3 Component Explorer

- Adobe Flex 3 Component Explorer
  - “Wow” your management and clients
  - Examples of built-in component library and how to use each component
  - Taste of the Flash effects available with Flex 3
More Flex Documentation

- Flex Developer’s Guide
- Programming ActionScript 3.0
- Building and Deploying Flex 3 Applications
Still More Flex Documents

- Flex 3 Quick Starts
  - [http://www.adobe.com/devnet/flex/?navID=gettingstarted](http://www.adobe.com/devnet/flex/?navID=gettingstarted)

- Adobe Technology Platform ActionScript Reference: Rich Internet Application Development
OpenLaszlo 4

Documentation

- OpenLaszlo documentation
  - http://www.openlaszlo.org/documentation

- LZX Reference Manual
  - Visual examples in Javadoc-like API HTML pages
  - Includes source code examples in API comments
  - http://www.openlaszlo.org/lps/docs/reference/
OpenLaszlo Components Examples

- OpenLaszlo Components Examples
  - “Wow” your management and clients
  - Examples of built-in component library and how to use each component
  - Taste of Flash/DHTML effects available
Flash, Flex, and OpenLaszlo: Dustin’s Blog Entries

- Dustin’s Blog Entries on ActionScript
  ➢ http://marxsoftware.blogspot.com/search/label/ActionScript

- Dustin’s Blog Entries on Flex
  ➢ http://marxsoftware.blogspot.com/search/label/Flex

- Dustin’s Blog Entries on OpenLaszlo
  ➢ http://marxsoftware.blogspot.com/search/label/OpenLaszlo
Flash, Flex, and OpenLaszlo: Additional Resources

- Adobe Flash Player
- Adobe Flex
- Laszlo Systems OpenLaszlo
  - [http://www.openlaszlo.org/](http://www.openlaszlo.org/)
Check These Out!

- OpenLaszlo Demos
  - http://www.openlaszlo.org/demos
- Flex Showcase
  - http://flex.org/showcase/

The demonstrations in this presentation are intentionally simple.

See the demonstrations at the links above for what is really possible with Flex, OpenLaszlo, Flash, and a little imagination.
Contrarian Opinions

- Flex Breaks the Web

- Seven Reasons Not to Use Flex
  - [http://visualajax.blogspot.com/2008/04/7-reasons-not-to-use-flex.html](http://visualajax.blogspot.com/2008/04/7-reasons-not-to-use-flex.html)
Q: What happens when an OpenLaszlo application runs on Flash Player 9 (does it work at all because of OpenLaszlo’s SWF7/SWF8 focus)?

A: OpenLaszlo applications will run on Flash Player 9 because Flash Player 9 is backwards compatible with Flash 7 and Flash 8 applications. However, the OpenLaszlo application cannot take advantage of Flash 9 performance and other features. OpenLaszlo currently has beta support for a Flash 9 runtime.
Some participants in the audience prefer using Flex Builder plug-in over using full out-of-the-box Flex Builder. Both are Eclipse-based.

One participant noted that Flex and Open Laszlo really encourage traditional client-server architecture with their clean separation and loose coupling between client (Flex or Open Laszlo in web browser) and server (such as Java EE application).
Questions / Comments

- One participant noted that he has used Flash Player socket connection capabilities to use more "pure" REST including functions such as PUT and DELETE.
  - This was in response to the point made in the presentation that the Flash Player only supports POST and GET HTTP functions.
  - Participant is submitting his socket-based, REST-supporting library to Adobe.
Questions / Comments

Q: Are there more advanced features for Flex DataGrid or OpenLaszlo Grid such as pagination rather than scrollbar for more entries than displayed at one time?

A: The standard DataGrid in Flex and Grid in OpenLaszlo do not do this, but there are third party components that do

- Flex Example:
  
  http://develop.gurufaction.com/App.swf