


MAX 2006 Beyond Boundaries

Jason Williams
Integrating Flex Apps with
Browsers and AJAX
Adobe Systems, Inc.



2006 Adobe Systems Incorporated. All Rights Reserved. 1

Flash and Ajax

"Ajax is likely to become a mainstream tool used by web developers as an alternative to other RIA technologies such as Flash and Microsoft Windows Presentation Foundation "Everywhere" (WPF/E), but it also provides an excellent solution when integrated with other RIA technologies, in particular Flash."

- Burton Group, January 2006



"AJAX and Flash are like peanut butter and jelly; each is good in its own right, but they are best when put together"

- Richard Monson-Haefel, Burton Group, March 2006

2006 Adobe Systems Incorporated. All Rights Reserved. 2

Why Flash?

- Additional functionality, providing richer experiences, that can reach everyone.
 - Audio and Video
 - Drawing and Animation
 - Local Storage
 - Data integration
 - Backwards and Forwards Compatibility at the SWF level

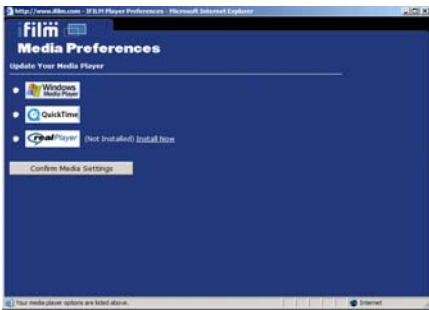
2006 Adobe Systems Incorporated. All Rights Reserved. 3

Demo

> PLACE HOLDER FOR IMAGES OF DEMO <

2006 Adobe Systems Incorporated. All Rights Reserved. 4

Experience matters



2006 Adobe Systems Incorporated. All Rights Reserved. 5

FSCCommand and SetVariable ("the hard way")

- fcommand() provides a primitive means of communication with the Flash Player host, using a single function with a "command" and "arguments" pattern.

Step 1: setup DoFSCCommand() in JavaScript

```

<html>
...
<script>
function mySwf_DoFSCCommand(command, args)
{
if (command == "displayAlert")
{
displayAlert(args);
}
else (command == "updateDisplay")
...
}
</script>
</html>

```

2006 Adobe Systems Incorporated. All Rights Reserved. 6

FSCCommand and SetVariable ("the hard way")

- To call the JavaScript function xxx_DoFSCCommand(...) use fscmmand() within the ActionScript.

Step 2: call JavaScript using fscmmand() from AS

```

package
{
import flash.display.Sprite;
...
public class FSCCommandExample extends Sprite
{
function FSCCommandExample()
{
super();
fscmmand("displayAlert", "Hello from JavaScript");
}
}

```

2004 Adobe Systems Incorporated. All Rights Reserved. 7

FSCCommand and SetVariable ("the hard way")

- SetVariable() provides a primitive means of communication with the Flash Player using JavaScript.

Step 3: Call into ActionScript using SetVariable()

```

<html>
...
<script>
function drawCircle(dia, x, y, color)
{
mySwf.SetVariable("circleDia", dia);
mySwf.SetVariable("circleX", x);
mySwf.SetVariable("circleY", y);
mySwf.SetVariable("circleColor", color);
mySwf.SetVariable("drawCircle", true);
}
</script>
</html>

```

2004 Adobe Systems Incorporated. All Rights Reserved. 8

FSCCommand and SetVariable ("the hard way")

- SetVariable will call a setter function on the root of the current swf

Step 4: Setup AS code to handle SetVariable calls

```

package
{
import flash.display.Sprite;
...
public class FSCCommandExample extends Sprite
{
...
public function set circleDia(value:int):void
{
_circleDia = value;
}
...
}

```

2004 Adobe Systems Incorporated. All Rights Reserved. 9

ExternalInterface ("the right way")

- ExternalInterface provides a programming interface that allows straight forward two-way communication between ActionScript and the Flash Player host environment.

Step 1: Setup AS code that will be invoked from JavaScript

```

package
{
import flash.display.Sprite;

public class ExternalInterfaceExample extends Sprite
{
...
public function drawCircle(dia:int, x:int, y:int, color:int):void
{
var circleShape = new Shape();
child.graphics.beginFill(color);
child.graphics.drawCircle(dia, dia, dia);
child.graphics.endFill();
addChild(circle);
}
...
}

```

2004 Adobe Systems Incorporated. All Rights Reserved. 10

ExternalInterface ("the right way")

- ActionScript can call JavaScript directly and synchronously.
- Get return values, pass any data type arguments, including strongly typed objects

Step 2: Setup JavaScript to call from ActionScript

```

<html>
...
<script>
function updateForm(name, city, state)
{
name.innerHTML = name;
city.innerHTML = city;
state.innerHTML = state;
return true;
}
</script>
...
</html>

```

Making the call

```

...
public class ExternalInterfaceExample extends Sprite
{
...
private function updateHTMLForm()
{
var result:Boolean =
ExternalInterface.call(
"Nevada", "updateForm", "Bob", "Las Vegas",
if (result)
}
...
}

```

2004 Adobe Systems Incorporated. All Rights Reserved. 11

ExternalInterface ("the right way")

- ExternalInterface is available on the following platforms:

Browser	Operating System	
Internet Explorer 5.0 and later	Windows	
Netscape 8.0 and later	Windows	Macintosh
Mozilla 1.7.5 and later	Windows	Macintosh
Firefox 1.0 and later	Windows	Macintosh
Safari 1.3 and later		Macintosh

- To check if it is available use the isAvailable() method

```

public class ExternalInterfaceExample extends Sprite
{
public function ExternalInterfaceExample()
{
if (ExternalInterface.isAvailable())
}
...
}

```

2004 Adobe Systems Incorporated. All Rights Reserved. 12

Demo

> PLACE HOLDER FOR IMAGES OF DEMO <

2006 Adobe Systems Incorporated. All Rights Reserved. 13

ExternalInterface

- Limited reentrancy (JavaScript)

2006 Adobe Systems Incorporated. All Rights Reserved. 14

ExternalInterface

- Limited reentrancy (ActionScript)

2006 Adobe Systems Incorporated. All Rights Reserved. 15

ExternalInterface

- Exceptions aren't marshaled

```

ActionScript Exception
...
child.graphics.beginFill(color);
if (dia == 0)
  throw new ArgumentError("Invalid diameter specified.");
child.graphics.drawCircle(dia, dia, dia);
...

JavaScript Exception
function updateForm(name, city, state)
{
  if (name == "")
    throw new Error("Invalid name specified.");
  name.innerHTML = name;
  ...
}

```

2006 Adobe Systems Incorporated. All Rights Reserved. 16

FlexAjaxBridge ("the easy way")

- The FlexAjaxBridge provides a thin layer of JavaScript and ActionScript that make Flex components and Flash primitives available to JavaScript using ExternalInterface.

Step 1: Setup Flex application

```

<mx:Application ...>
  <mx:LineChart id="theChart" >
  <FABridge>
  </mx:Application>

```

Step 2: Use the application within HTML

```

<html>
...
<script>
function displayDataInChart()
{
  var chart = FABridge.flash.root().theChart();
  chart.setDataProvider([x:10, y:20, z:50]);
}
</script>
</html>

```

2006 Adobe Systems Incorporated. All Rights Reserved. 17

Demo

> PLACE HOLDER FOR IMAGES OF DEMO <

2006 Adobe Systems Incorporated. All Rights Reserved. 18

FlexAjaxBridge ("the easy way")

- Exceptions aren't marshaled.
- Reentrancy is limited
- Memory management is more complex.
 - Object references marshaled across the bridge in either direction are held
 - To release references call releaseASObjects()

```

<html>
...
<script>
...
  FABridge.flash.releaseASObjects();
...
</script>
...
</html>

```

2006 Adobe Systems Incorporated. All Rights Reserved. 19

Data Integration

- Break the limitations of XML/HTTP
- Get direct access to ColdFusion, .NET, JRun, J2EE, SOAP Services, and Sockets (XML/Binary)

2006 Adobe Systems Incorporated. All Rights Reserved. 20

Flex Data Management Services (FDMS)

- FDMS provides rich client applications with rich data functionality.
 - Built on a robust messaging architecture
 - Automates data synchronization between client and server
 - Real-time data push
 - Occasionally connected clients
 - Conflict detection, notification, and resolution
 - Clustering for fail over and load balancing
 - Publish/Subscribe messaging

2006 Adobe Systems Incorporated. All Rights Reserved. 21

Ajax Client for Flex Data Services (ACFDS)

- Ajax Bridge for Data Services provides access from JavaScript for messaging and data management services.

Step 1: Load FDMSBridge into HTML page

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
  <title>Products</title>
  <link href="css/accordion.css" rel="stylesheet" type="text/css" />
  <link href="css/screen.css" rel="stylesheet" type="text/css" />
  <script type="text/javascript" src="include/FABridge.js" />
  <script type="text/javascript" src="include/FDMSLib.js" />
</head>
<body>
  <script>
    FDMSLibrary.load("ajax/Products/FDMSBridge.swf", fdmsLibraryReady);
  </script>
...

```

2006 Adobe Systems Incorporated. All Rights Reserved. 22

Ajax Client for Flex Data Services (ACFDS)

- Use the data services API similarly to how it is in Flex.

Step 2: Request data from the remote destination

```

<html xmlns="http://www.w3.org/1999/xhtml">
...
</body>
<script language="javascript">
  var productService;
  var products;
  // Once the bridge indicates that it is ready we can proceed to load the data.
  function fdmsLibraryReady()
  {
    productService = new DataService("Products");
    productService.addEventListener(DataService.RESULT, productsResult);
    productService.addEventListener(DataService.FAULT, productFault);
    products = new ArrayCollection();
    productService.fill(products);
  }

```

2006 Adobe Systems Incorporated. All Rights Reserved. 23

Ajax Client for Flex Data Services (ACFDS)

- Remote requests are asynchronous

Step 3: Update HTML elements with retrieved data

```


<html xmlns="http://www.w3.org/1999/xhtml">
...
</body>
<script language="javascript">
  function productsResult(event)
  {
    var htmlText = "<table id='productTable'>";
    var product;
    for (var i=0; i<products.length; i++)
    {
      product = products.getItemAt(i);
      htmlText += "<tr><td>"+ product.getName() +
        "</td><td>"+ product.category + "</td></tr>";
    }
    htmlText += "</table>";
    document.all["products"].innerHTML = htmlText;
  }
...

```

2006 Adobe Systems Incorporated. All Rights Reserved. 24

Demo MAX

> PLACE HOLDER FOR IMAGES OF DEMO <


2006 Adobe Systems Incorporated. All Rights Reserved. 25 

Ajax Client for Flex Data Services (ACFDS) MAX

- Integration with Spry

Step 1: Load FDMSLibrary into Spry HTML page and use the FDMSDataSet

```
<html xmlns="http://www.w3.org/1999/xhtml">
...
</body>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>2006 MAX - AJAX Songs Demo</title>
<link href="SongsDemo.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="includes/spry.js"></script>
<script type="text/javascript">
    var dsTopSongs = new FDMSDataSet("Songs");
    dsTopSongs.setColumnTypes("rank", "number");
</script>
</head>
<body>
...
</body>
</html>
```

2006 Adobe Systems Incorporated. All Rights Reserved. 26 

Better by Adobe.™

2006 Adobe Systems Incorporated. All Rights Reserved. 27 