Intro to Adobe Flex

A brief introduction to Adobe’s RIA development tool.

What is Flex?

- Flex is a a cross-platform development framework for creating Rich Internet Applications (RIAs).
- A component based tool that you can use to develop apps that run using the Flash Player or Adobe Air.
- Adobe Flex Builder™ software is an Eclipse™ based IDE for developing RIAs
- A competitor or compliment to AJAX, JavaFX, Silverlight (depends who you ask and in what context)
- Very cool

What isn’t Flex?

- It’s not a timeline based animation tool
- It’s not expensive - It’s actually available for students for free and also as an open source plug-in for Eclipse
- It’s not a server-side language
- It’s not an answer for everything

Origins of Flex

- Flex 1 and 1.5 were server based languages where MXML was compiled/executed at request time, like ColdFusion or PHP
- Built in Eclipse, compiled on the server
- It was expensive
- Adobe saw value in the Flash Platform, bought Macromedia and has embraced the framework, version 3 has just launched with 4 not too far behind.
- Opensourced in 2007. SDK/Compiler can be downloaded and snapped into Eclipse for free.
- Education version is also free

Current State of Flex

- Growing quickly:
    - Developers are embracing MXML and AS3.
    - The player has nearly a 95% penetration rate.
    - Dependable and Fast: Flex content looks like Flex content on every browser.
    - Easy to debug - Compile time errors, profiler, refactoring support
- Flex blogs, books, tutorials and conferences are exploding
- Flex related job listings are also exploding

Basics of Flex

- Runs in the Flash Player or AIR
- Tag based markup MXML serves as the basis for Flex projects
- MXML is a custom XML namespace, looks like <mx:MyTag></mx:MyTag>
- Component based development: Containers/Layout Components, UI Elements, Data Connectors, more
- Actionscript 3 can be used to add behaviors or other interactivity not covered by MXML
- Supports common OOP practices like MVC patterns, code behind, etc.

Basic of Flex Continued

- MXML components can be brought onto the canvas via the GUI similar to Dreamweaver or VisualStudio
- Markup can be written by hand with code assist, too. Again, similar to Dreamweaver or VisualStudio
- MXML components have attributes that allow for modification of appearance or behavior
- Components can be extended for additional capabilities or reuse via MXML or AS or a combo of both.
- Containers control layout (Canvas, Vertical Box, Horizontal Box)
- UI components display data, control interactivity or display states or other visual feedback (Datagrids, lists, buttons, selects menus, radio buttons, etc.)
- Data components load or send data (Services, XML, Remoting, etc)
- Other tags include: Script, Style, etc.
- Tags can be nested similar to HTML/XML

Designing Interfaces in Flex

- CSS can be used to style the components
- Flash, PNGs, and JPGs can be used to skin the components, too
- Illustrator, Photoshop, Fireworks, Flash all have component design sets to assist in UI skinning
- Constraint based layouts and percentage based placements make polished, uniform layouts much easier

My First Flex App: Hello World

```
    <?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">
    <mx:Label x="10" y="10" text="Output" id="output"/>
    <mx:Button x="10" y="36" label="Click Me" click="{output.text = 'Hello World'}"/>
</mx:Application>
```

My First Okay Flex App: Hello World from XML

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">
    <mx:Script>
        <![CDATA[
            //my favorite little random function - seed with floor and ceiling and it brings back a random integer from the range
            public function randRange(min:Number, max:Number):Number {
            var randomNum:Number = Math.floor(Math.random() * (max - min + 1)) + min;
                return randomNum;
            }
        ]]>
    </mx:Script>
    <mx:XML id="myXML" xmlns="">
    <greetings>
        <greeting>Hola Mundo</greeting>
        <greeting>Hello World</greeting>
        <greeting>Ciao Mondo</greeting>
        <greeting>Hallo welt</greeting>
        <greeting>ハローワールド</greeting>
    </greetings>
    </mx:XML>
    <mx:Label x="10" y="10" text="Output" id="output"/>
    <mx:Button x="10" y="36" label="Click Me" click="{output.text = myXML.greeting[randRange(0, 4)].valueOf()}"/>
</mx:Application>
```

My First More Okay Flex App: Hello World from External XML

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">
    <mx:Script>
        <![CDATA[
            //my favorite little random function - seed with floor and ceiling and it brings back a random integer from the range
            public function randRange(min:Number, max:Number):Number {
            var randomNum:Number = Math.floor(Math.random() * (max - min + 1)) + min;
                return randomNum;
            }
        ]]>
    </mx:Script>
    <mx:XML id="myXML" source="externalXML.xml" xmlns="" />
    <mx:Label x="10" y="10" text="Output" id="output"/>
    <mx:Button x="10" y="36" label="Click Me" click="{output.text = myXML.greeting[randRange(0, 4)].valueOf()}"/>
</mx:Application>
```

More Flex Info

- adobe.com/flex
- flex.org
- weblogs.macromedia.com/mxna
- opensource.adobe.com
- visualrinse.com
- 30onair.com

Enough yakking, let’s code

- Cross your fingers
- Stick around to heckle and get some freebies