

From Java to Flex

Peter Paugh

Software Engineer

Chariot Solutions

Agenda

- What are Flex and ActionScript
- Going from Java to Flex
- How will Eclipse help
- When to Choose Flex
- Demos

What are Flex 3.0 and ActionScript 3.0

- *ActionScript* is the language of Flash
- *Flex* is a declarative UI markup in XML
- Anything that can be done in *Flex* can be done in *ActionScript*
- *Flex* is built on *ActionScript* with *ActionScript* and gets compiled to *ActionScript*

Flex Is Open Source

- *opensource.adobe.com*
- SDK
- Compilers
- Debugger
- Core libraries
- Blaze DS (formerly part of LiveCycle DS)
- NOT Open Source
 - FlexBuilder IDE, Flash plug-in, AIR

Going from Java to 'Flex'

- More appropriate to compare ActionScript to Java
- Compare Flex to JSF w/ Facelets
- ActionScript is
 - ECMAScript 4 compliant
 - statically typed... sorta.
 - Retains full dynamic behaviour

Thinking in ActionScript...

(thanks Bruce Eckel;-)

- Start with *Java*, then...
- Local variable promoting
- Constructors are always public
- No overloading
- Everything is an object

Still different...

- Default parameter values
- Formal Set and Get methods
`object.foo = 9;` actually calls
`object.setFoo(9);`
- Override is a keyword
- No notion of abstract

And now for something completely different...

Closures...

- Global functions, e.g. `trace()`, `setInterval()`
- Package-Level functions
- Nested functions
- Source-File-Level functions – compare to private static methods
- Function literals

Arrays

- Think ArrayList that also has [] syntax
- Sparse
- Automatically grow
- Contain mixed types
- Set Filter and Sort functions; refresh()
- push, pop, shift, unshift, splice, concat, toString
- delete()

Data Binding

- Built-in
- @Bindable for POJOs
- “{}” syntax
 - `<mx:Label text=“ { username.text } ”`
- `<mx:Binding>`
- Validators

Native XML, E4X

```
var bookXml:XML = <books>
    <book isbn="059652689">
        <title>Programming Flex 2</title>
        <author>Chafic Kazoun</author>
    </book>
    <book isbn="0596526946">
        <title>Essential ActionScript
            3.0</title>
        <author>Colin Moock</author>
    </book>
</books>

for each (var child:XML in bookXml.*) {
    trace(child.title);
    child.@isbn = 2;
    child.foo = "bar";
}
```

Events & Event Handling

...

```
<mx:Button id="filePickerButton" click="onBrowse()"
           label="Browse" />
<mx:Text id="localFileNameLabel" />
```

...

```
<mx:Script>
  <![CDATA[
    private var localFile:FileReference = new FileReference();

    public function onBrowse():void{
      localFile.addEventListener(Event.SELECT, onFileSelected);
      localFile.browse();
    }

    public function onFileSelected(event:Event):void{
      localFileNameLabel.text = localFile.name;
    }
  ]]>
</mx:Script>
```

...

Garbage Collection & Threads

- GC is very Java-like
- Incremental Mark & Sweep
- No explicit threads

No Direct DB Access

- RPC
 - HTTP
 - Web
 - RemoteObject – requires Blaze DS
- Messaging – requires Blaze DS
- Exception AIR

Cairngorm Framework

- Client-side MVC
- Event driven
- View class per page
- Singleton Model
- Controller to manage asynchronous communication
- Delegate to make the server call

JavaScript Interface

- Flex can call JavaScript
 - `ExternalInterface.call("javascript fn");`
- JavaScript can call Flex
 - `ExternalInterface.addCallback("javascript callable fn", flexFn);`

How Eclipse will Help

- FlexBuilder or Eclipse plug-in for \$
- Some auto-complete/suggest
- Basic debugging
 - No inspect, evaluate, conditional breakpoints
- No JavaDoc, double-click from stack trace to source,
- No Optimize imports for Flex files

When to Choose Flex

- Constraint based
 - Rich client
 - Support wide variety of browsers
 - ‘Near zero’ install is okay
 - Fat client is okay
- GWT
- JSF
- Applet

Demos

- eBay Desktop
- HP Print Studio
- Harley-davidson.com
- onreflexion.blogspot.com

Resources

- Flex.org
- Adobe.com