An Introduction To Apache Flex

Justin Mclean
Class Software

Email: justin@classsoftware.com
Twitter: @justinmclean
Blog: http://blog.classsoftware.com

Who am I?

- Programming for 25 years
- Developing and creating web applications for 15 years
- Certified developer and trainer in Flex
- Adobe Community Professional
- Apache Flex Podling PMC member and committer

Apache Flex

History of Flex

- 2004 Macromedia closed source server solution Flex 1.0 + 1.5 $15K/CPU
- 2005 Adobe acquisition of Macromedia
- 2006 open source Flex 2.0
- 2007 Flex 3.0 2010 Flex 4.0 2011 Flex 4.5
- End 2011 donation to Apache

What is Apache Flex?

- Application Framework
- Developer friendly
- Target a wide range of platforms
- Currently uses the Flash or AIR runtimes
- Adobe continues to provide support and resources to Apache Flex

Apache Flex Framework

- Application architecture
- Common UI components
- Rapid cross platform application development
- Easy Integration with back end services
- Create mobile applications
- Aimed at enterprise style applications

What was Donated?

- Flex SDK (including automation and data visualisation)
- ActionScript and MXML compilers
- New spark UI components
- BlazeDS
- Flex testing tools (Mustella)
- New generation compiler (Falcon)

Why Apache?

- iOS would never support Flash in a browser
- Adobe unable to keep up with growing number of hardware platforms
- Flex was open source but model was broken
- Adobe was unable to work out a way to make money out of Flex

Current Status

- Flex SDK 4.6 framework donated
- Apache Flex 4.8 Parity release
- IDE packager (10 release candidates!)
- Infrastructure set up (few issues)
- Several new committers
- Moving from SVN to Git

Apache Flex Development

ActionScript Overview

- Based on JavaScript with Java/C features
- Open source specification, compiler and VM
- Class based not prototype based
- Package and namespace support
- Strongly typed (mostly)
- Both compile time and runtime type checking
- Event handling based on DOM events

Compared to Java

- Dynamic - flexibility when you need it
- No difference between functions and variables
- Setters and getters show as properties
- No method overloading
- Function parameters default parameters
- Primitive types and references
- XML is a native type
- Fully featured arrays, collections and vectors

MXML

- XML UI markup language
- Used to create flexible and simple layout
- Can nest files / reuse components
- Supports binding and event handlers
- Include optional code blocks
- Treated as an ActionScript class

MXML Snippet

```
<s:FormItem label="Name">
  <s:Label text="{person.name}" />
</s:FormItem>
<s:FormItem label="Email">
  <s:Label text="{person.email}" />
</s:FormItem>
<s:FormItem label="Apache ID">
  <s:Label text="{person.apacheID}" />
</s:FormItem>
```

Components

- Breaks up complex UI into bite size pieces
- Components can be reused
- Components can be MXML or ActionScript
- Components can be used in MXML or ActionScript
- Components communicate with each other via variable binding and dispatching/listening for events

Using MXML Components

```
<s:Application
xmlns:components="components.*">

  <components:SelectPerson
people="{people.people}"
selectPerson="changePerson(event)"/>
  <components:PersonDetails
person="{currentPerson}" />

</s:Application>
```

Binding

- Group of classes to allow variable binding
- Watch for changes of value to a variable and updates anything bound to that variable
- Can be implemented in MXML or ActionScript
- [Bindable] metadata and { } indicate binding
- Used to update UI
- Used to bind component properties

MXML Snippet

```
[Bindable] public var person:Person;

<s:FormItem label="Name">
  <s:Label text="{person.name}" />
</s:FormItem>
<s:FormItem label="Email">
  <s:Label text="{person.email}" />
</s:FormItem>
```

Events

- Dispatched via user interaction with your application OR when something occurs in your application
- Register interest in an event by either writing an event handler OR by listening for an event
- Can dispatch your own custom events
- Use to reduce dependancies / loosely couple components

Event Handlers

- Method called when an event occurs
- Take an event (of some type) and return void
- Can cancel, prevent default behaviour or stop events propagation

Event Handler

```
protected function
onChangePerson(event:IndexChangeEvent):void
{
   var person:Person =
   (event.target as List).selectedItem;

   var personEvent:PersonEvent = new
   PersonEvent(PersonEvent.SELECT_PERSON, person, true);

   dispatchEvent(personEvent);
}

<s:List dataProvider="{people}"
   change="onChangePerson(event)" labelField="name" />
```

Event Bubbling

- 2 main phases in events - target and bubbling
- Target events occurred on component eg a button is clicked
- Event then bubbles to parent of the target eg a button inside you was clicked
- Event bubbles all the way up to base application

MVC for free!

- Custom components are your view
- Have your view bind to simple ActionScript classes (your model)
- Dispatch custom event for loose coupling to tell application (your controller) to update model
- Update model and views automatically change because of binding

Browser Application

- Runs in Flash Player virtual machine in the browser

Desktop Application

- Runs in the AIR runtime
- AIR runtime can be packaged with application
- Browser to Desktop just change Application to WindowsApplication
- Also supports application updates, file access, web views, SQLLite, native windows, menus and more

Mobile

- Runs in AIR runtime but compiled natively for IOS and Android
- Can deploy applications in app stores
- Application structure slightly different need to use View's and Navigator to move between them
- UI components optimised for mobile and touch
- Support gestures, notifications and other mobile specific functionality
- Native extensions

Many More features

- Advanced debugger and profiler
- Unit testing framework
- Datagrid UI control
- Item editors and item renderers
- RSLs/modules
- CSS support
- I18n and L10n support

Get Involved

- Download and have a play give us feedback
- Sign up and contribute to the mailing list
- Look through JIRA there's fair amount of simple issues to fix
- We're new to this so need help with things not Flex related

Links

- Apache Flex site
  http://incubator.apache.org/flex/
- Mailing list sign up
  http://incubator.apache.org/flex/mailing-lists.html
- Apache Flex mail archives
  http://markmail.org/search/+list:org.apache.incubator.flex-dev
- Apache Flex JIRA
  https://issues.apache.org/jira/browse/FLEX
- Apache Flex SVN viewer
  http://svn.apache.org/viewvc/incubator/flex/trunk/

Adobe White Papers

- Adobe Flex white paper: http://www.adobe.com/devnet/flex/whitepapers/roadmap.html
- Adobe Flash white paper and roadmap: http://www.adobe.com/devnet/flashplatform/whitepapers/roadmap.html

Questions?

Ask now, see me after the session,
follow me on twitter @justinmclean
or email me at justin@classsoftware.com.

Slides can be found at Google Docs
https://docs.google.com/open?id=0B3cTQYHN73CEZXBoaE1VaWFLRWs

Code can be found at GitHub
https://github.com/justinmclean/ApacheConFlexExample