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
- Apache Flex PMC, Incubator PMC, Apache member
- Release manager for Apache Flex, FlexUnit, Tour De Flex, Squiggly
- Run IoT meetup in Sydney Australia

History

- 2004 First release by Macromedia
- 2007-2011 Flex 3.0, Flex 4.0, Flex 4.5 by Adobe
- 2011 Incubation in Apache
- 2012 4.8 release and top level Apache project
- 2013 Flex 4.9, 4.10 and 4.11 released
- 2014 Flex 4.12 and 4.13 released
- 2015 Flex 4.14 released

What is Apache Flex?

- Application framework
- Developer and designer friendly
- Targets a wide range of platforms
- Currently uses the Flash or AIR runtimes
- WIP JS cross compiler and framework
- Adobe continues to provide resources to Apache Flex

Apache Flex Framework

- Provides application architecture
- Set of common UI components
- Rapid cross platform application development
- Easy Integration with back end services
- Easy to create mobile applications
- Enterprise style applications

Recent News

- New mobile skins
- Flat Spark theme
- Support for Promises
- Upcoming Flex JS 0.5 release
- IntelliJ open source plugin
- Maven support

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 (but types optional)
- Both compile time and run time type checking
- Event handling based on DOM events

ActionScript Snippet

```
public class People extends EventDispatcher {
   [Bindable] public var people:ArrayCollection
            = new ArrayCollection();

   public function People(autoLoad:Boolean = true) {
      if (autoLoad) {
         loadXML();
      }
   }
   …
}
```

Compared to Java

- Flexibility when you need it
- No difference between functions and variables
- Setters and getters show as properties
- No method overloading
- Functions arguments can have defaults
- 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/or 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

- 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 { } set up binding
- Commonly used to update UI
- Commonly 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 bubble
- Can create / dispatch your own custom events
- One way to reduce dependancies / loosely couple components

Event Handlers

- Method called when an event occurs
- Take an event (or subclass of event) 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;

   dispatchEvent(new PersonEvent
      (PersonEvent.SELECT_PERSON, person, true));
}

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

MVC for free!

- Custom components are your view
- Have your view bind to simple ActionScript classes (your model)
- Dispatch custom events for loose coupling to tell application (your controller) to update model
- Update model and views automatically change (no code required) via binding

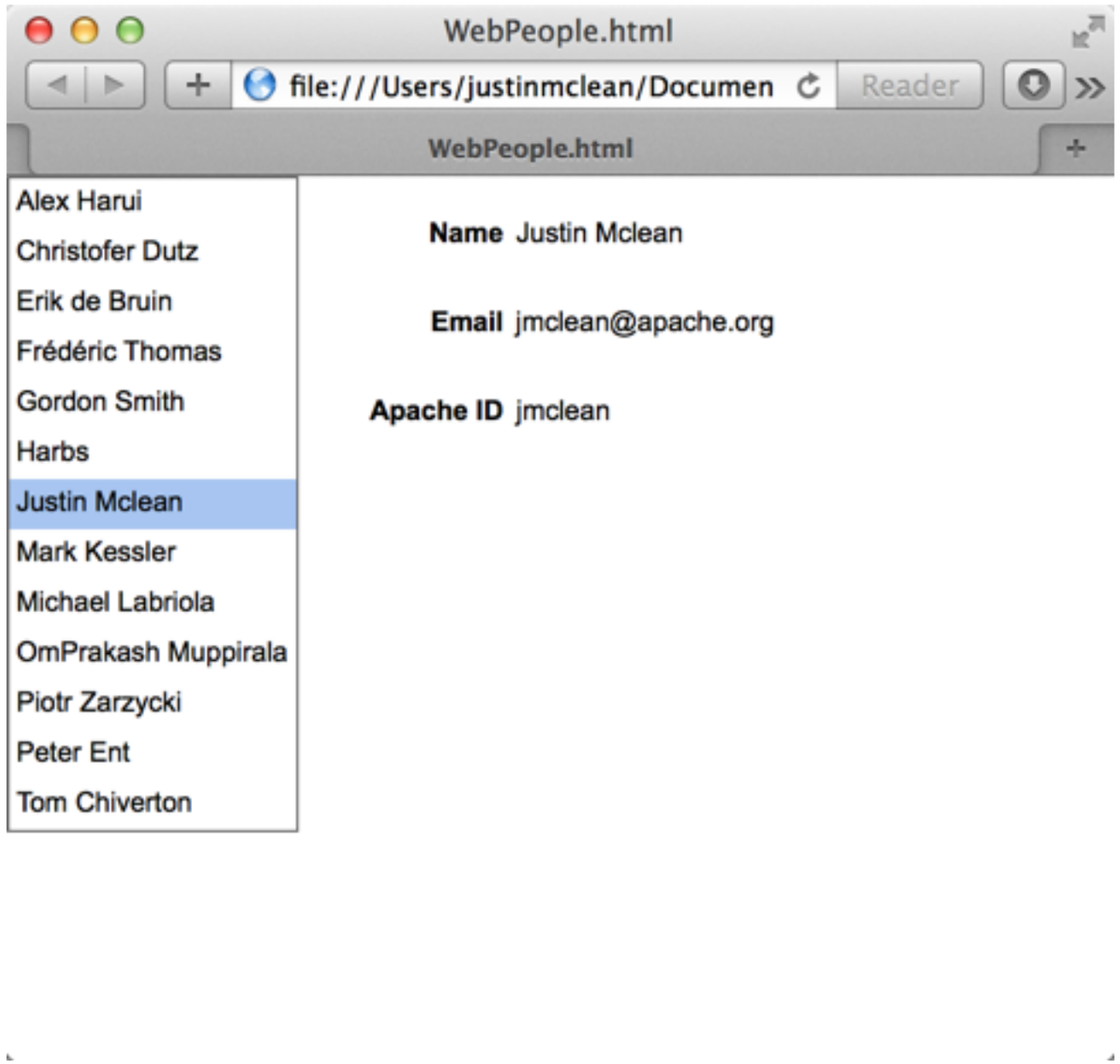
Simple Model

```
public class Person
{
   public var name:String;
   public var email:String;
   public var apacheID:String;

   public function Person(name:String,
                          email:String, apacheID:String) {
      this.name = name;
      this.email = email;
      this.apacheID = apacheID;
   }
}
```

Browser Applications

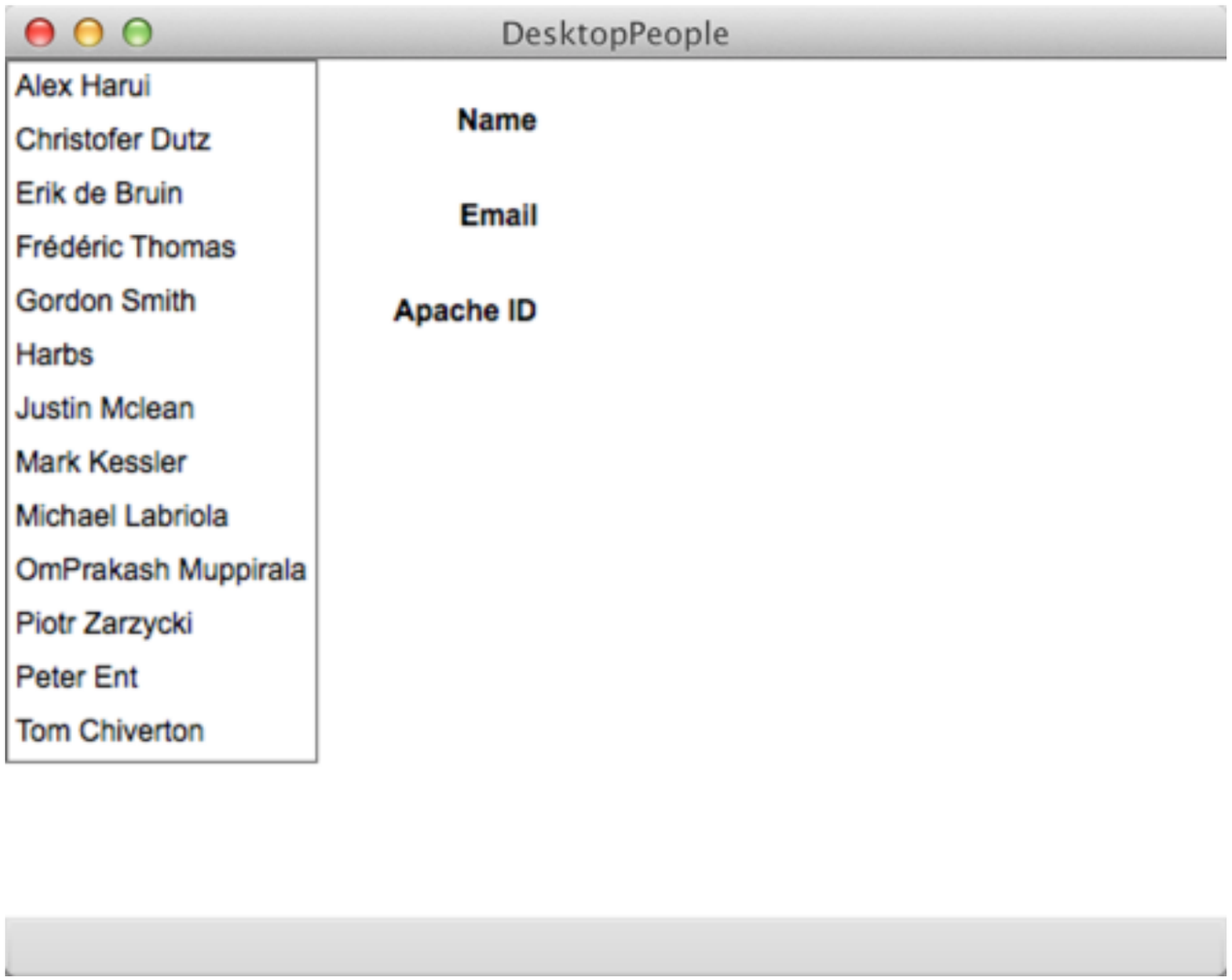
- Runs in Flash Player virtual machine in the browser
- JavaScript framework is being actively worked on and you can download beta version


WebPeople.html
file:///Users/justinmclean/Documen
Reader
WebPeople.html
Alex Harui
Christofer Dutz
Erik de Bruin
Frédéric Thomas
Gordon Smith
Harbs
Justin Mclean
Mark Kessler
Michael Labriola
OmPrakash Muppirala
Piotr Zarzycki
Peter Ent
Tom Chiverton
Name Justin Mclean
Email jmclean@apache.org
Apache ID jmclean

Browser

Desktop Applications

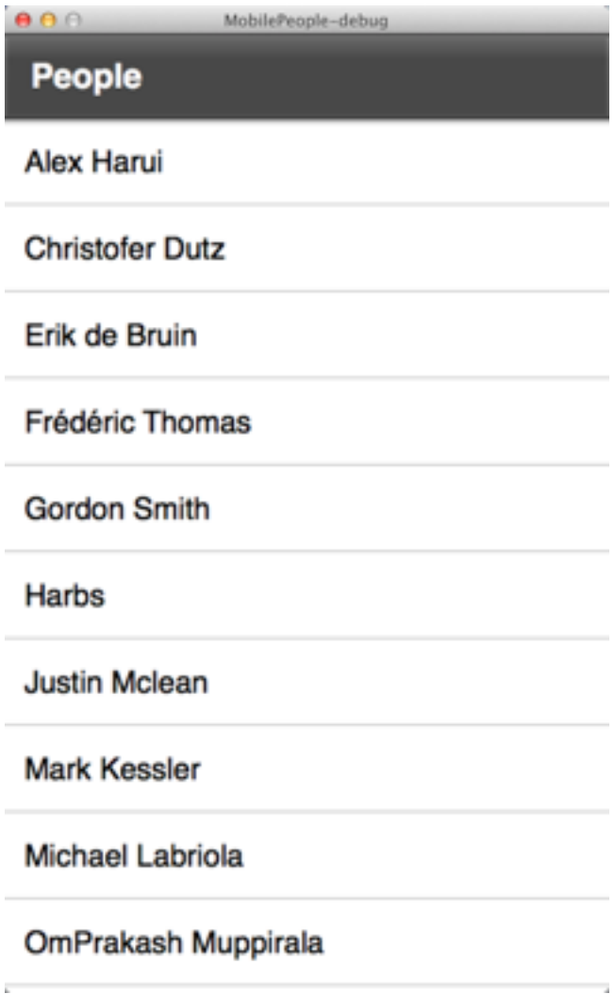
- Runs in the AIR runtime
- AIR runtime can be packaged with application
- Browser to Desktop just change Application to WindowsApplication
- Also supports application updates, file access, web view, SQLLite, native windows, menus and lots more


DesktopPeople
Alex Harui
Christofer Dutz
Erik de Bruin
Frédéric Thomas
Gordon Smith
Harbs
Justin Mclean
Mark Kessler
Michael Labriola
OmPrakash Muppirala
Piotr Zarzycki
Peter Ent
Tom Chiverton
Name
Email
Apache ID

Desktop

Mobile Applications

- Runs in AIR runtime but compiled into native applications for IOS and Android
- Can deploy applications in app stores
- Application structure slightly different - 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


MobilePeople-debug
People
Alex Harui
Christofer Dutz
Erik de Bruin
Frédéric Thomas
Gordon Smith
Harbs
Justin Mclean
Mark Kessler
Michael Labriola
OmPrakash Muppirala

Mobile

Many More Features

- Skinning and styling
- Advanced debugger and profiler
- FlexUnit testing framework
- Datagrid UI control
- Item editors and item renderers
- 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

Links

- Apache Flex site
  http://flex.apache.org
- Mailing list sign up
  http://flex.apache.org/community-mailinglists.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 GitHub mirror
  https://github.com/apache/flex-sdk

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 Conference Site.

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