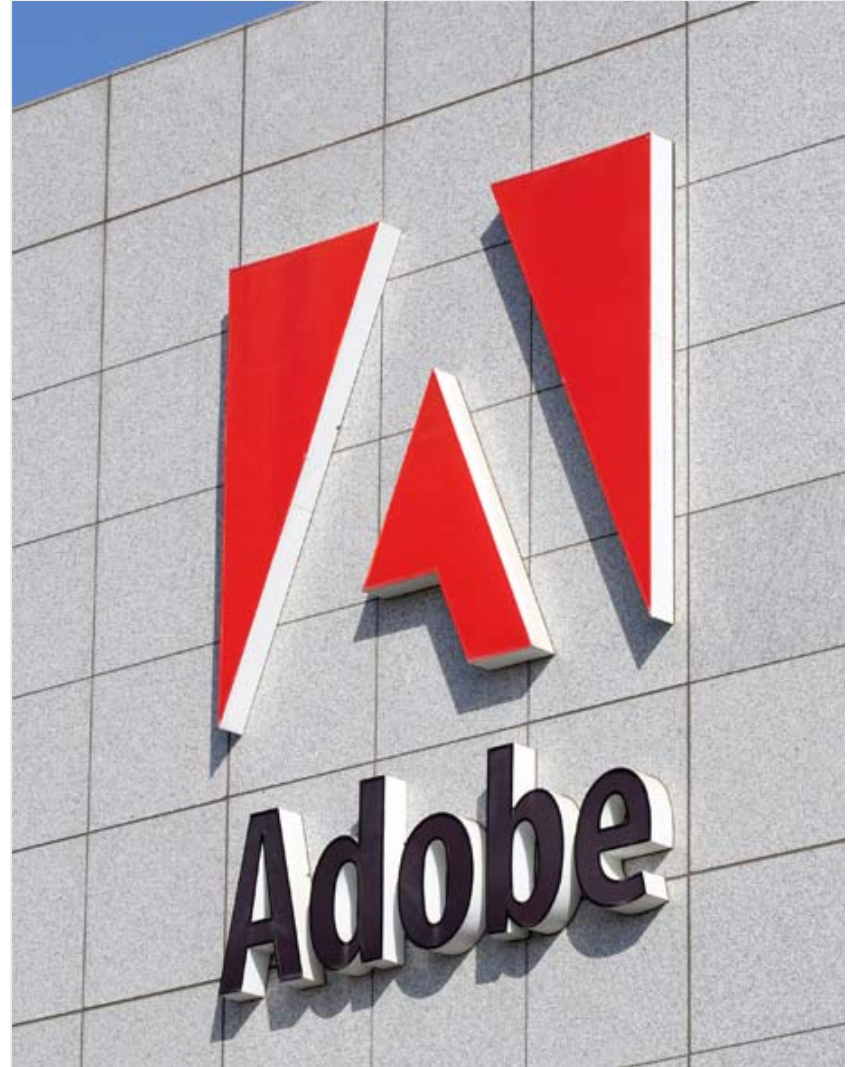


Flex 2 Overview

Matt Chotin

Adobe Systems



Flex 2 Product Line



Flex SDK 2

Flex Languages

MXML and ActionScript 3.0

Flex Framework

Components, Containers, Layout Managers, Behaviors, and Effects

Flex Utilities

Compiler, Debugger, Automated Testing



Flex Data Services 2

Message Service

Data Management Service

RPC Services



Flex Builder 2

Visual Layout

Code Hinting

Debugging

Skinning and Styling



Flash Player 9



Flash Player 9

ActionScript 1/2

ActionScript 3

- New Virtual Machine
- Actionscript 3
- Seamless Deployment

Flex Appeal

- A focus on expressiveness
- Rich functionality included throughout the framework
 - Drag and Drop
 - Error indicators
- Everything is customizable
- Steady learning curve for developers
 - Leverage basic functionality in a matter of minutes/hours
 - Advanced features in days
 - Component developer after a single summit!
- The power of the dual nature of AS3
 - Strongly typed
 - Ability to add dynamic behavior

Actionscript 1

```
PushButton = function() { // this is a constructor

    var saveWidth = this._width;

    var saveHeight = this._height; // typical flash voodoo

    this._xscale = this._yscale = 100;

    this.setSize(this.saveWidth, this.saveHeight);

    this.createEmptyTextField("fLabel",1,100,0,0,100,100);
}

PushButton.prototype = new MovieClip(); // this set up inheritance

PushButton.prototype.setSize(w,h) { // this is a method

    this.fWidth = w; this.fHeight = h;

    this.draw();

}

PushButton.prototype.draw = function() {

    this.moveTo(0,0);

    this.beginFill(0x474747,100); // this is pretty much the

    this.lineTo(this.fWidth,0); // entire drawing API

    this.lineTo(this.fWidth,this.fHeight);

    // etc.

}
```

Actionscript Evolved

Actionscript 1

- Based on ECMAScript (javascript)
- Functions, Prototypes, Delegates
- No types – everything is 'Object'
- Language of the past

Actionscript 3

- ES3 Compliant, ES4 implementation
- Classes, Packages, Interfaces
- Dynamic typing for 'script feel', strong static typing for larger projects.
- E4X makes working with XML easy
- Language of the future

Actionscript 3

```
package my.custom.components {           // package code organization
import mx.core.UIComponent;             // imports for aliasing
[Bindable]                               // class and field metadata
public class PushButton extends UIComponent // true classes and inheritance
{
    private var fLabel:TextField;         // member variables
    private var fToggled:Boolean         // strong typing, access control
    public var icon:*;                    // dynamic typing still legal

    public function PushButton() {        // true constructors
        var tf:TextField = new TextField();
        tf.text = "Hello, World";
        addChild(tf);
    }
    public function get toggled():Boolean { // get/set properties
        return fToggled;
    }

    // explicit overrides
    override protected function draw(width:Number, height:Number) {
    }
}
}
```

Flex Framework and Flash Player Building Blocks



Effects	Validators	Formatters	Charts
Layouts	Controls	List Based Controls	Collections

Cursor	View States	Effects / Animation	Formatting/Validation
Drag/Drop	ToolTips	Automation	RPC
History	Procedural Rendering	Dynamic Layout	Component Model
Application Focus	Bootstrap	CSS / Styling	Component Accessibility



Graphics / DOM	Text / Fonts	External	Code Access
Networking / IO	XML / Binary Data	Accessibility	Security
Keyboard / Focus	Mouse	Audio / Video	Printing

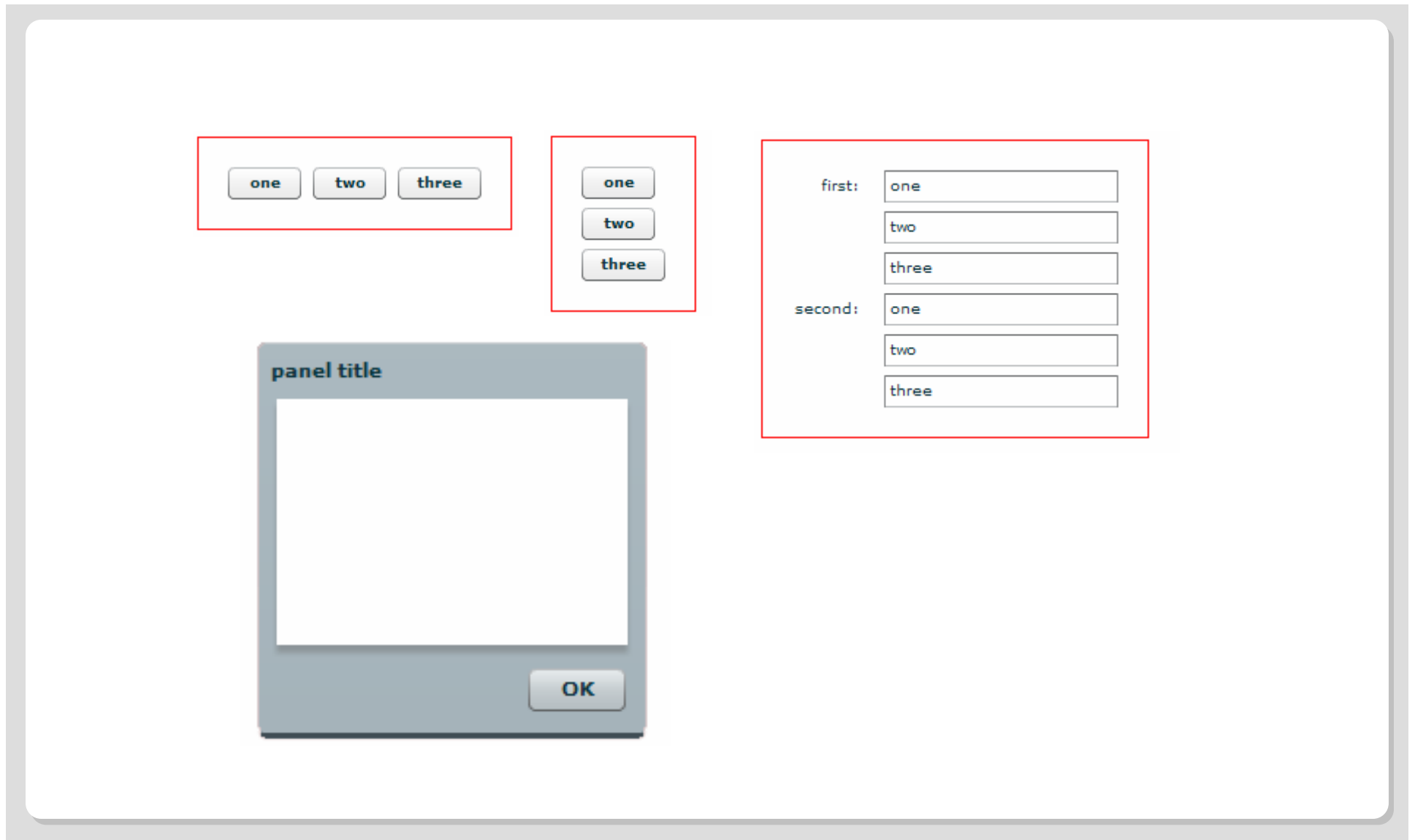
Components: Controls

The image displays a variety of standard UI controls:

- Button:** A button labeled "Hello".
- Checkbox:** A checkbox labeled "Enabled".
- Slider:** A horizontal slider with a range from 0 to 100.
- Text Input:** A text input field containing the text "edit me".
- Calendar:** A calendar for January 2006, with the 31st highlighted.
- Color Picker:** A color picker showing a grid of colors and a hex code "000000".
- Tree View:** A tree view showing a hierarchy of nodes: <XMLLiteral>, <node label="root">, <node label="child1">, <node label="child 3">.
- List Box:** A list box containing items "item 1" through "item 7", with "item 5" selected.
- Table:** A table with three columns (A, B, C) and multiple rows of data.

A	B	C
47	23	59
34	23	59
37	23	3423
74	920	4403
44	232	488
77	344	59
83	233	23359
34	344	494
2	2358	59
47	23	59
34	23	59
37	23	3423
74	920	4403

Components: Containers and Layouts



MXML

- MXML is a compile time source format
- No MXML appears in the SWF
 - SWF has bytecode
- Declarative format:
 - More readable
 - More maintainable
 - More toolable
- *MXMLC* generates Actionscript classes from markup
- *Generative* approach saves the developer busywork

```
var a:Application = new Application();
a.percentWidth=100;
a.percentHeight=100;

var p:Panel = new Panel();
a.addChild(p);
p.percentWidth=100;
p.percentHeight=100;
p.title="summit app";

var h:HBox = new HBox();
p.addChild(h);
h.percentWidth=100;
h.percentHeight=100;

var l:List = new List();
h.addChild(l);
l.width=300;
l.percentHeight=100;

var dataSvc:HTTPService = new HTTPService();
dataSvc.url = "http://localhost/userList/";
dataSvc.addEventListener("result",function(event:Event)
    {l.dataProvider = dataSvc.result});

var details:DetailPane = new DetailPane();
h.addChild(details);
l.addEventListener("change",function(event:Event)
    {details.user = l.selectedItem} );
```

Who wants to write this?

More MXML

- MXML language defines very few tags
- Any class discoverable by compiler is fair game
- Tags, attributes, events, and styles all driven by the class definition
 - Events and Styles exposed via metadata
- No special treatment for the framework
 - Except for some legacy RPC service support, but we don't like to talk about that
- Namespaces used like packages to organize code
 - You can create your own!
- Special logic to parse data binding expressions and generate appropriate watchers
 - Developers expose properties as sources for binding

MXML to Actionscript

```
<Application creationComplete="srv.send();" >
  <HTTPService url="..." id="srv" />
  <Panel title="demo">
    <List id="userList"
      dataProvider = "{srv.result}" />
    <HRule />
    <Label text="name:" />
    <TextInput id="firstName"
      text="{userList.selectedItem.firstName}" />
    <ControlBar>
      <Button label="Apply"
        click="changeUserName(firstName.text);" />
    </ControlBar>
  </Panel>
</Application>
```

MXML to Actionscript

```
public class MyApp extends mx.core.Application {

public function MyApp() {
    srv = new HTTPService();
    srv.url = "...";
    var panel0:Panel = new Panel();
    panel0.title = "demo";
    addChild(panel0);
    userList = new List();
    watchExpression("srv.result",__executeBinding0);
    panel0.addChild(userList);
    ...
    var button0:Button = new Button();
    button0.addEventListener("click",__handleEvent0);
}

public var srv:HttpService;
public var userList:List;

private function __executeBinding0():void {
    userList.dataProvider = srv.result;
}
private function _handleEvent0(e:Event):void {
    changeUserName(firstName.text);
}
}
```

Classes to MXML

```
package adobe.web {  
    [Event(name="selectionChanged", type="flash.events.Event")]  
    public class AssetBrowser extends mx.core.UIComponent {  
        public var displayMode:String;  
        [Bindable] public var selectedAsset:String;  
    }  
}
```

```
<adb:AssetBrowser xmlns:adb="adobe.web.*"  
    displayMode="details"  
    selectionChanged="loadImageDetails();"  
    id="browser"  
>
```

```
<adb:PreviewPane  
    imageSource="{browser.selectedAsset}" />
```

Delivering Components

- *COMPC* generates SWCs which can contain your components
- Multiple packages can be combined using manifest file
- Developers simply drop the SWC into their library path and away they go
- Post them to the Flex Exchange!

Miscellaneous

- Flex provides localization support through locale-specific compilation and resource bundles
- Exposing our internal documentation tool, ASDoc, is in the works
- Unit testing frameworks are available from labs (FlexUnit) and we're investigating additional testing solutions geared towards components

Better by Adobe.™