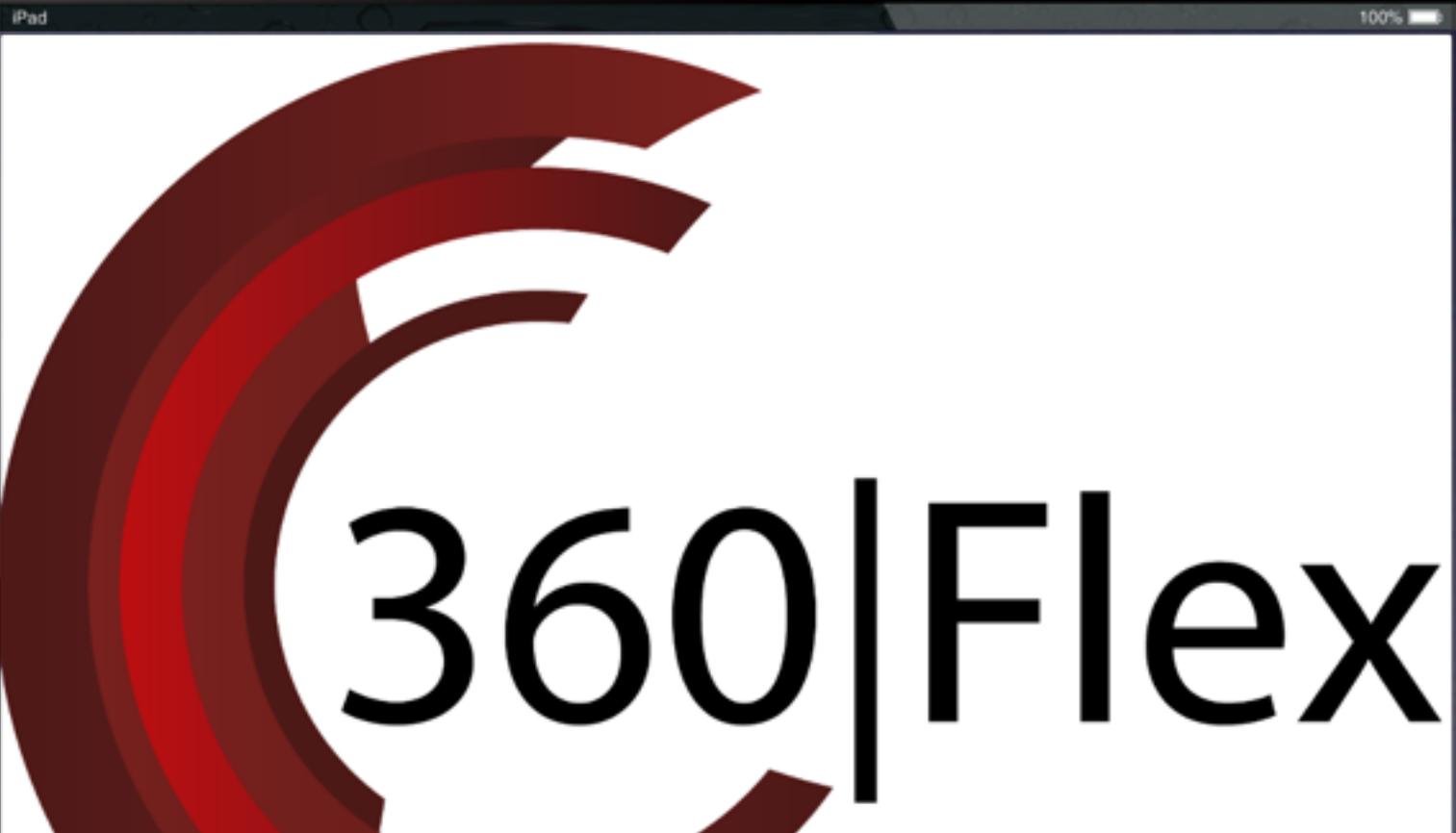


Building (iPad) Apps with Flex



@danielwanja
<http://n-so.com>

Agenda

- Flex SDK 4.6
- Views and View Navigation
- Components

Me

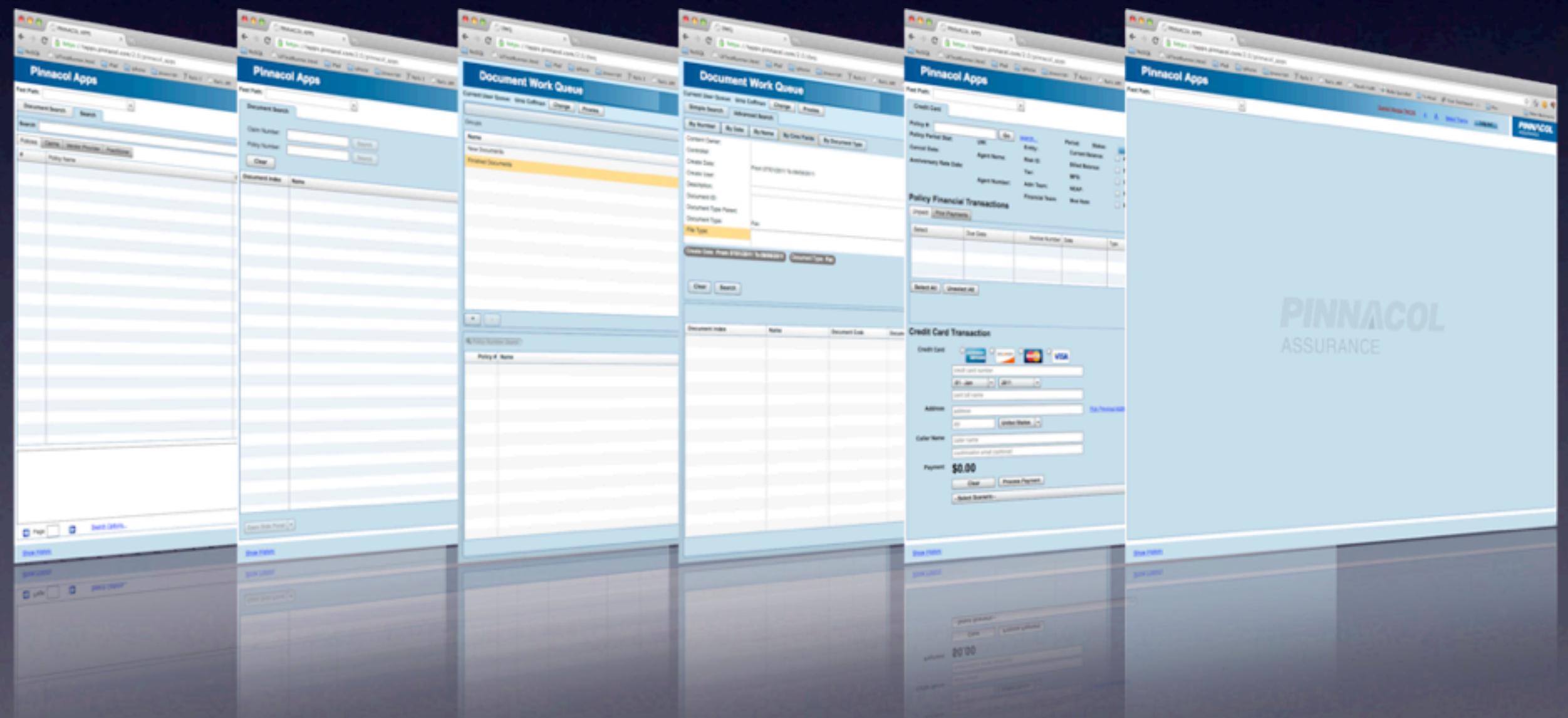
- d@n-so.com
- [@danielwanja](https://twitter.com/danielwanja)
- n-so.com/blog
- onrails.org
- appsden.com
- flexonrails.com
- github.com/danielwanja



=



Pinnacol -> Flex + Rails



PINNACOL
ASSURANCE

Portfolio



SEARCH

Men Women

murr



Reset

MATCHES

<Back



D. Ferrer v A. Murray

Ferrer, who had defeated world No.1 Rafael Nadal in the quarterfinals, continued his goo...



R. Federer v A. Murray

Although he lagged in their head-to-head, Federer held the edge in Grand Slam play, an...



A. Murray v N. Djokovic

Djokovic was the form player coming into the final, and from the first point, the result never...



M. Cilic v A. Murray

Playing in his first major semifinal, Cilic began

HIGHLIGHTS

<Back



D. Ferrer v A. Murray (Semi Finals)

Andy Murray gets the better of Ferrer on a fantastic point by both players. 40-15



D. Ferrer v A. Murray (Semi Finals)

Andy Murray riffs a 211 km/h service ace down the center line. 30-15

D. FERRER



VERSUS

6	6	1	6
4	7	6	7
1	2	3	4
			5

A. MURRAY



WON

2011 MEN'S SEMI FINALS



Share this moment



DOWNLOAD
THE OFFICIAL
AUSTRALIAN
OPEN 2012
app for iPad®

proudly presented to you by ANZ



STAY IN TOUCH

australian open tv

australian open.com



tennis.com.au

My Own Stuff

The collage displays five different applications or interfaces:

- time.onrails.org - Time tracking made simple.**: A web application for time tracking, showing a sidebar with tasks like "NCAA | February" and "Pinnacle | January [via harvest]".
- RailsLogVisualizer v0.5**: A desktop application for visualizing log files, showing a tree view of log entries and a pie chart labeled "TimeController".
- Usage Report v0.2 for Amazon Web Services™**: A web-based usage report for AWS, displaying a table of costs per day and a pie chart of usage type breakdown.
- Mifi**: A mobile interface for monitoring Mifi usage, showing IN (287.78 KB) and OUT (52.98 KB) traffic, connected status (00:05:30), and IP addresses (75.253.107.78, 255.255.255.255).
- WebSnapshot**: A desktop application for taking screenshots of web pages, showing a thumbnail of the Ruby on Rails homepage.
- Pick Your Clock**: A mobile application for setting alarms, showing a grid of clock icons and a digital display showing "1001 10110".

And a book



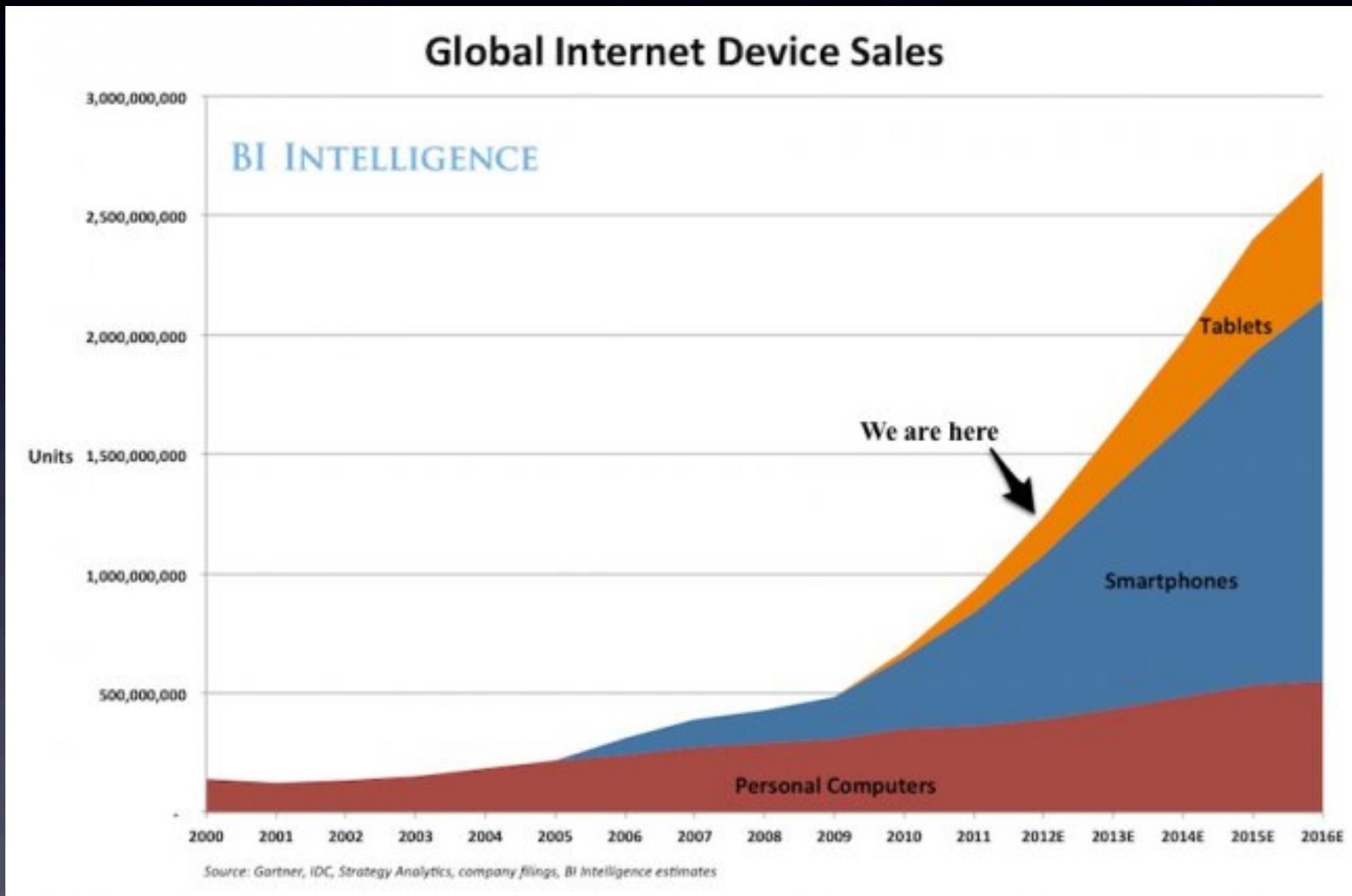
github/danielwanja

- `activeresource` - Flex to Ruby on Rails
- `talks/iPadAppsWithFlex` - This talks and apps source code
- `TourDeMobileFlex` - A demo of the Flex SDK
- `UndoManager` - An experiment!

Flex and mobile?

- Really?
- Yea, Flex for Mobile rocks! Let's check it out.
- ...the good, the bad and the ugly!

Why should you listen?



Ways to build iOS apps

- Native App - xCode
- Hybrid App - PhoneGap, Titanium, ... others ... and Flex SDK
- Mobile Web - HTML5, JavaScript, CSS

Why should you listen?

- If you know Flex...it's easy to get going with mobile development.
- Flex = iOS, Android and more...

Tour De Mobile Flex Demo



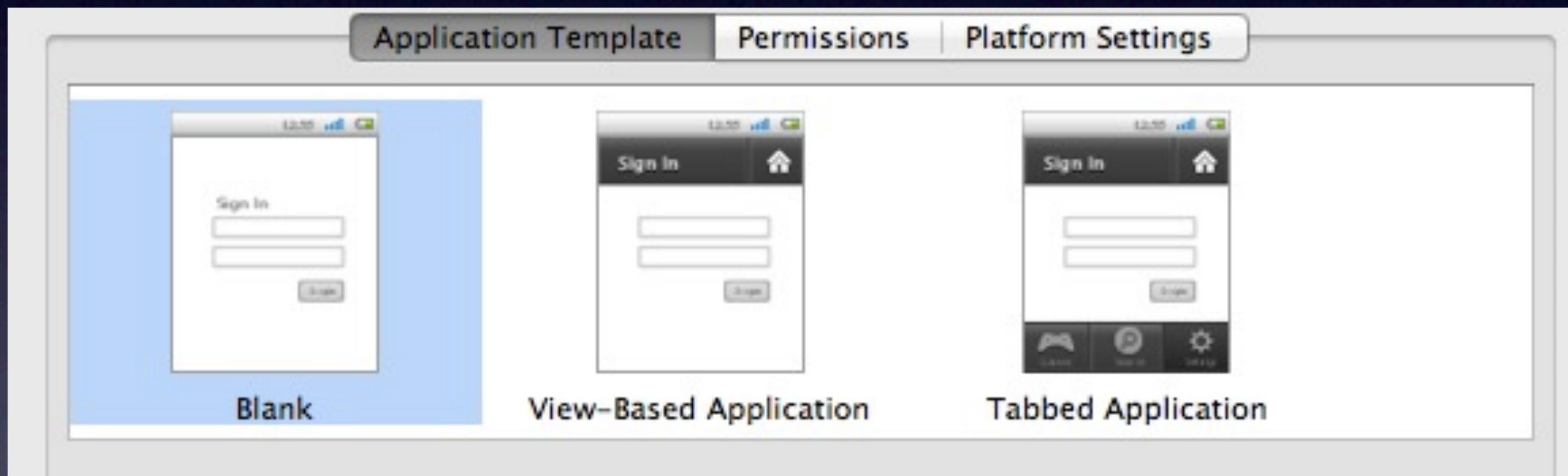
Development Workflow

- Desktop Emulator is fast
- Nothing beats using the real thing. I use an Android Tablet to develop..
- Unless you use the app on the Tablet...you won't know if it's right.

Development Workflow

- FlashBuilder -> Debug/Run on Simulator
-> Debug/Run on Device (iOS no USB)
- ADT -> Debug/Run on Simulator
-> Debug/Run in iOS Emulator (XCode)
-> Debug/Run on Device (wifi)
-> Debug/Run on Device (usb)
- Workflow is simpler with Android

FB: 3 templates (for now)



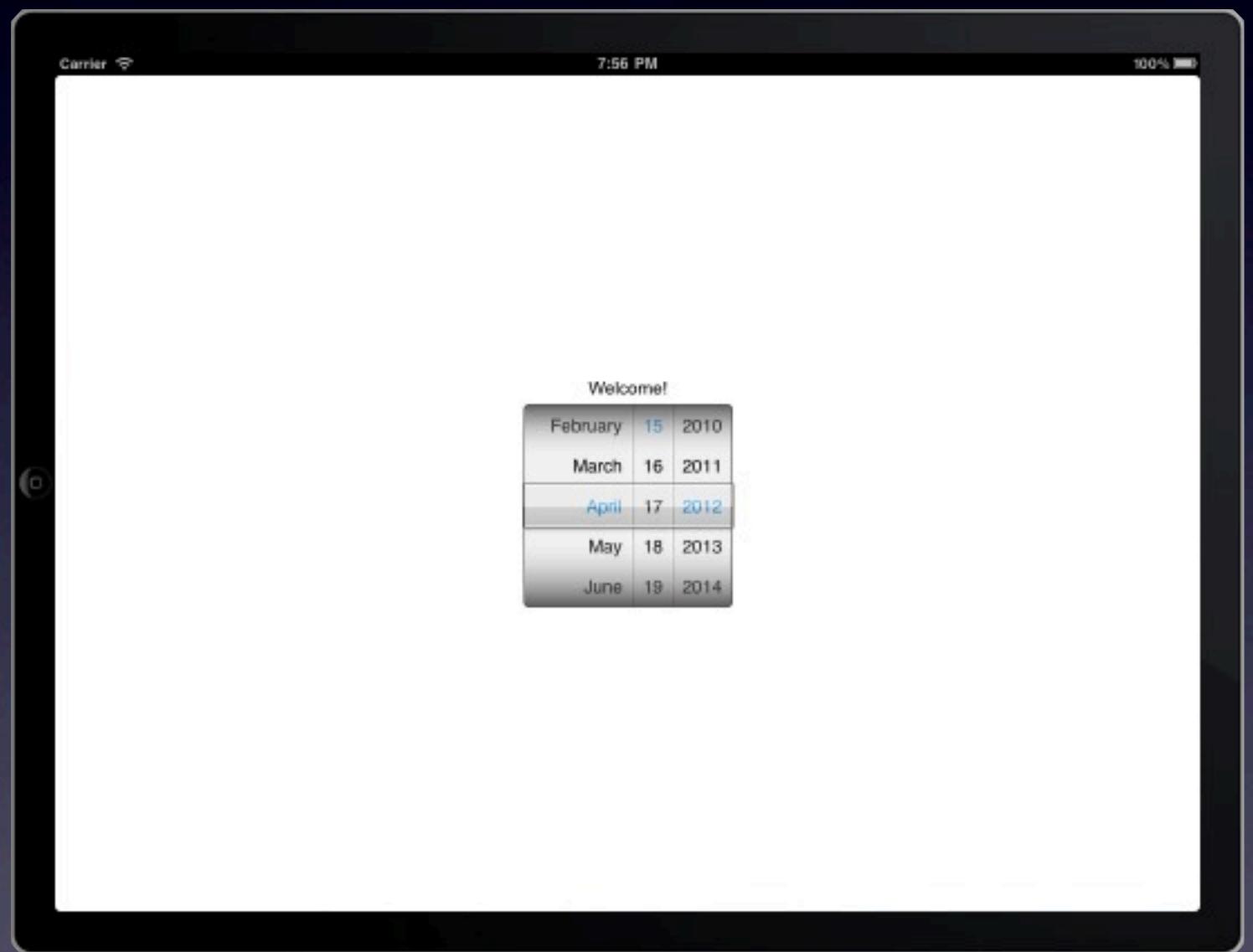
Blank App

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark" applicationDPI="160">
    <fx:Declarations>
        <!-- Place non-visual elements (e.g., services, value objects) here -->
    </fx:Declarations>
</s:Application>
```

Blank App

```
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark" applicationDPI="160">
    <s:layout>
        <s:VerticalLayout horizontalAlign="center" verticalAlign="middle"/>
    </s:layout>
    <s:Label text="Welcome!" />
    <s:DateSpinner />
</s:Application>
```

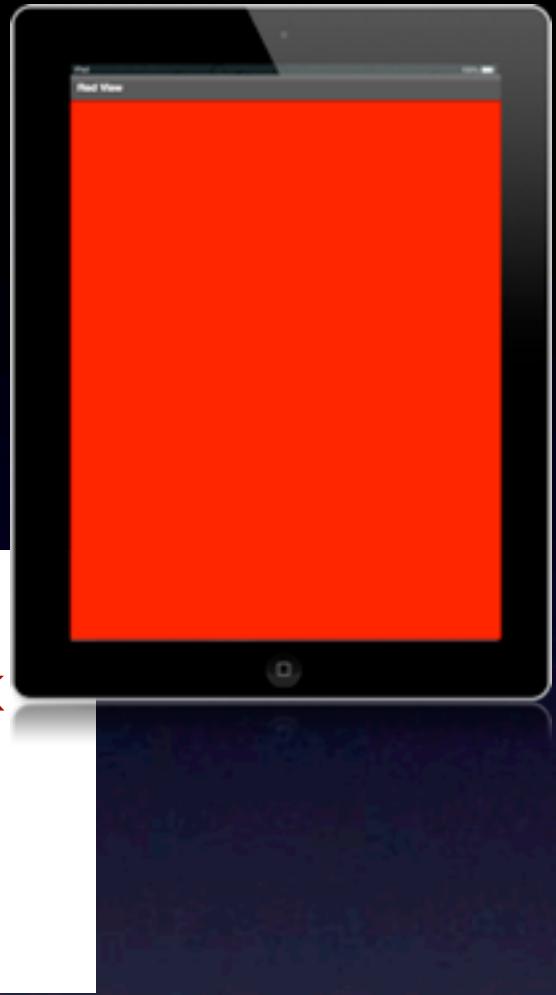
Blank APP



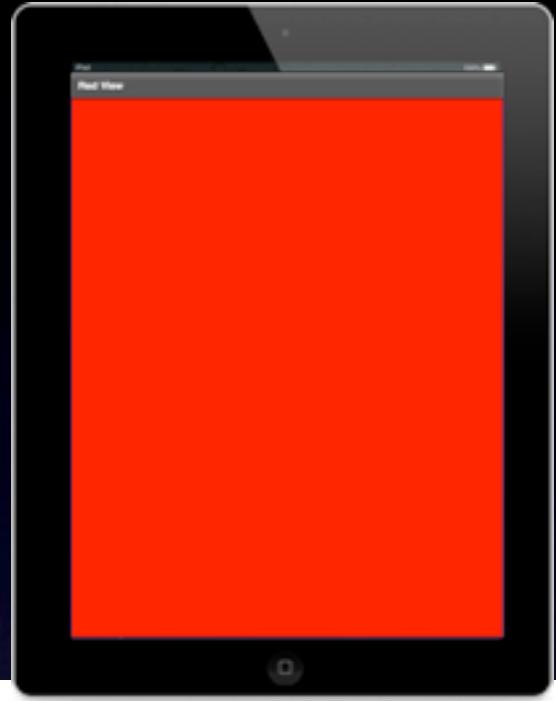
ViewNavigatorApplication

```
<s:ViewNavigatorApplication xmlns:fx="http://ns.adobe.com/mxml/2009"  
    xmlns:s="library://ns.adobe.com/flex/spark"  
    firstView="views.RedView"  
    applicationDPI="160">  
</s:ViewNavigatorApplication>
```

```
<s:View xmlns:fx="http://ns.adobe.com/mxml/2009"  
    xmlns:s="library://ns.adobe.com/flex/spark"  
    title="Red View"  
    backgroundColor="red">  
</s:View>
```



navigator.pushView()



```
protected function showBlueClickHandler(event:MouseEvent):void
{
    var data:Object = null;
    var context:Object = null;
    var viewTransition:ViewTransitionBase = null;
    navigator.pushView(BlueView, data, context, viewTransition);
}
```

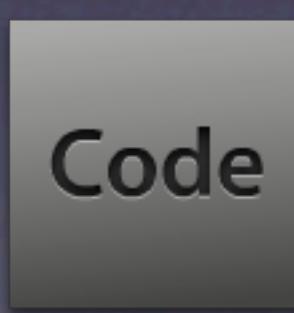
navigator.popView()



```
protected function goBackClickHandler(event:MouseEvent):void
{
    var viewTransition:ViewTransitionBase = null;
    navigator.popView(viewTransition);
}
```

ViewNavigator

- pushView()
- popView()
- popToFirstView()
- popAll()



app01_ViewNavigatorNavigation

TabbedViewNavigatorApplication

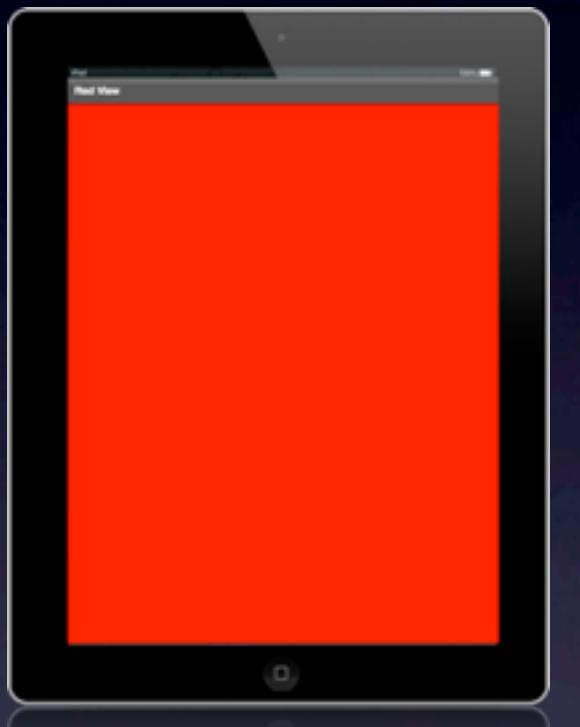
```
<s:TabbedViewNavigatorApplication>
    <s:ViewNavigator label="Red" width="100%" height="100%"
        firstView="views.RedView"/>
    <s:ViewNavigator label="Green" width="100%" height="100%"
        firstView="views.GreenView"/>
    <s:ViewNavigator label="Blue" width="100%" height="100%"
        firstView="views.BlueView"/>
</s:TabbedViewNavigatorApplication>
```

Red

Green

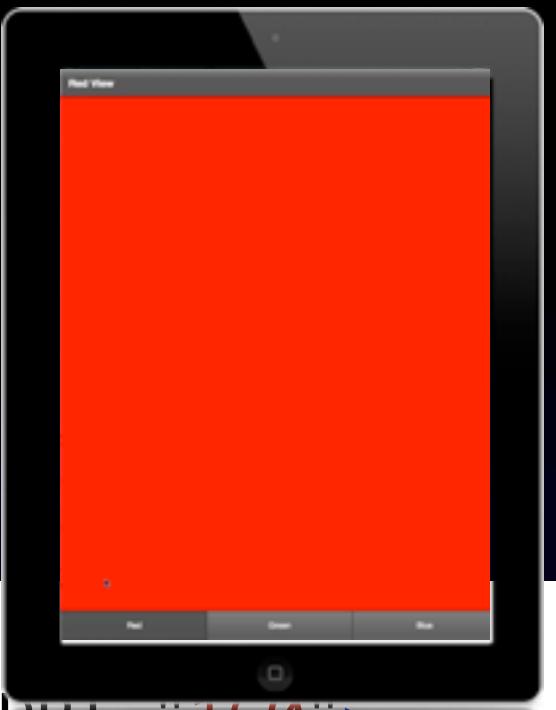
Blue

ViewNavigator



```
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark" applicationDPI="160">
    <s:ViewNavigator    firstView="views.RedView"    width="100%" height="100%"/>
</s:Application>
```

TabbedViewNavigator



```
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark" applicationDPI="160">
    <s:TabbedViewNavigator width="100%" height="100%">
        <s:ViewNavigator label="Red" width="100%" height="100%"
            firstView="views.RedView"/>
        <s:ViewNavigator label="Green" width="100%" height="100%"
            firstView="views.GreenView"/>
        <s:ViewNavigator label="Blue" width="100%" height="100%"
            firstView="views.BlueView"/>
    </s:TabbedViewNavigator>
</s:Application>
```

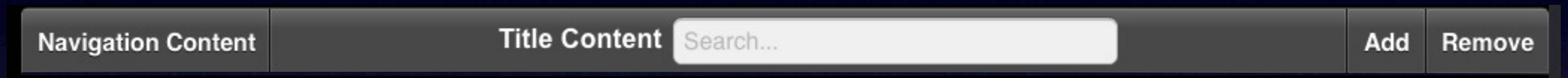
Anatomy of a Flex Mobile View

ActionBar

View Body



ActionBar



Navigation Content

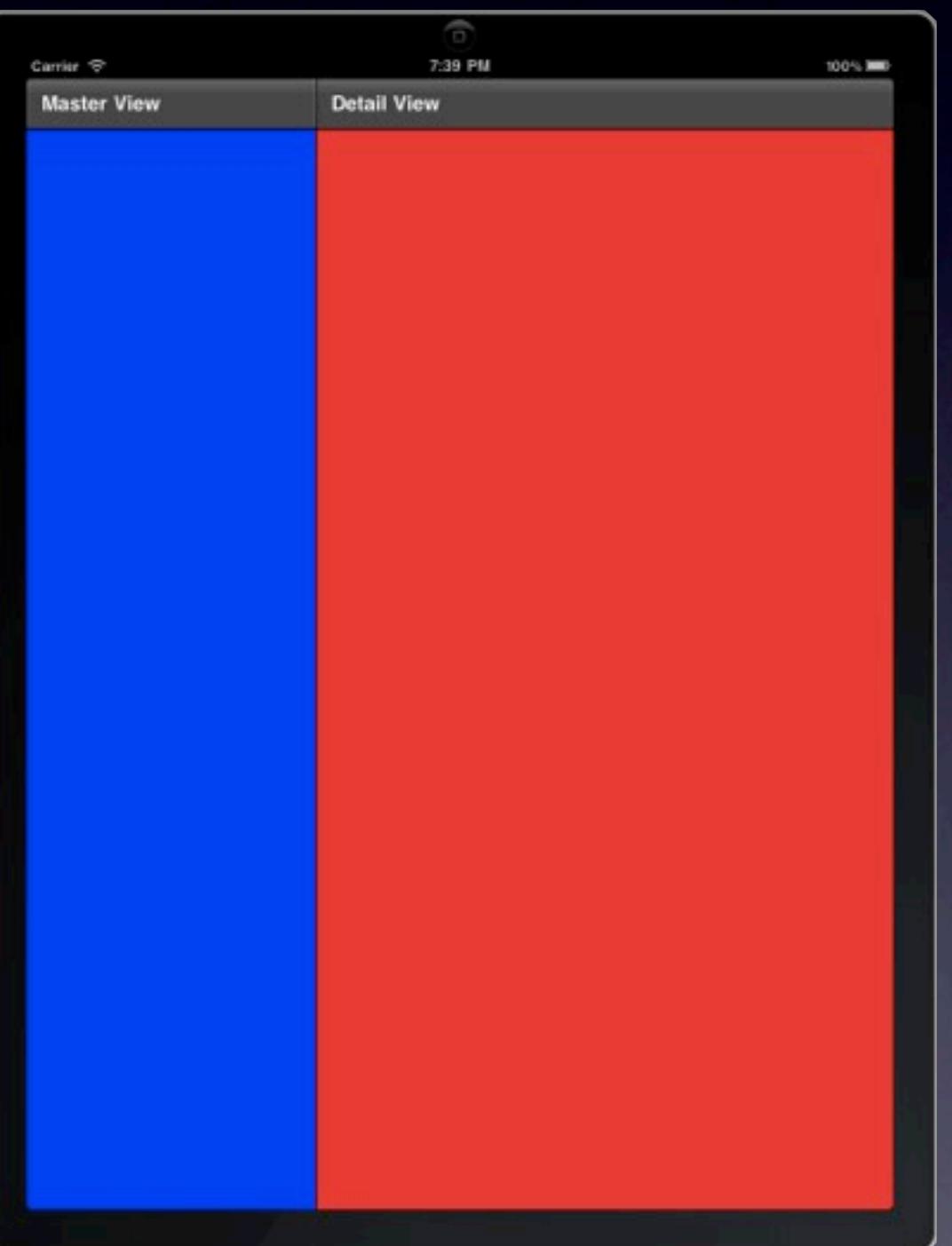
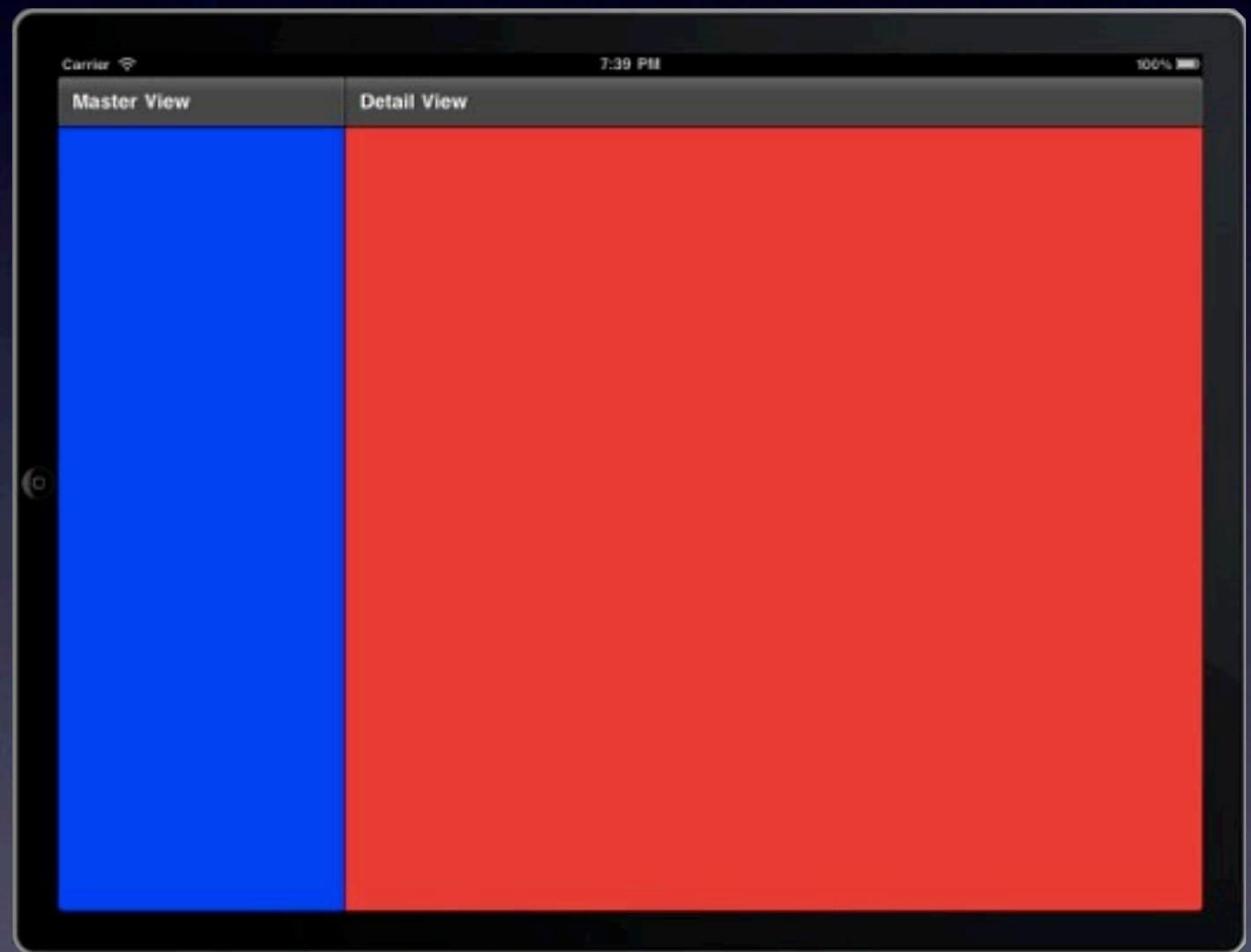
Title Content

Action Content

Navigation, Title, Action

- actionBarVisible
- actionContent
- actionLayout
- title
- titleContent
- titleLayout
- navigationContent
- navigationLayout
- overlayControls
- viewMenuItems

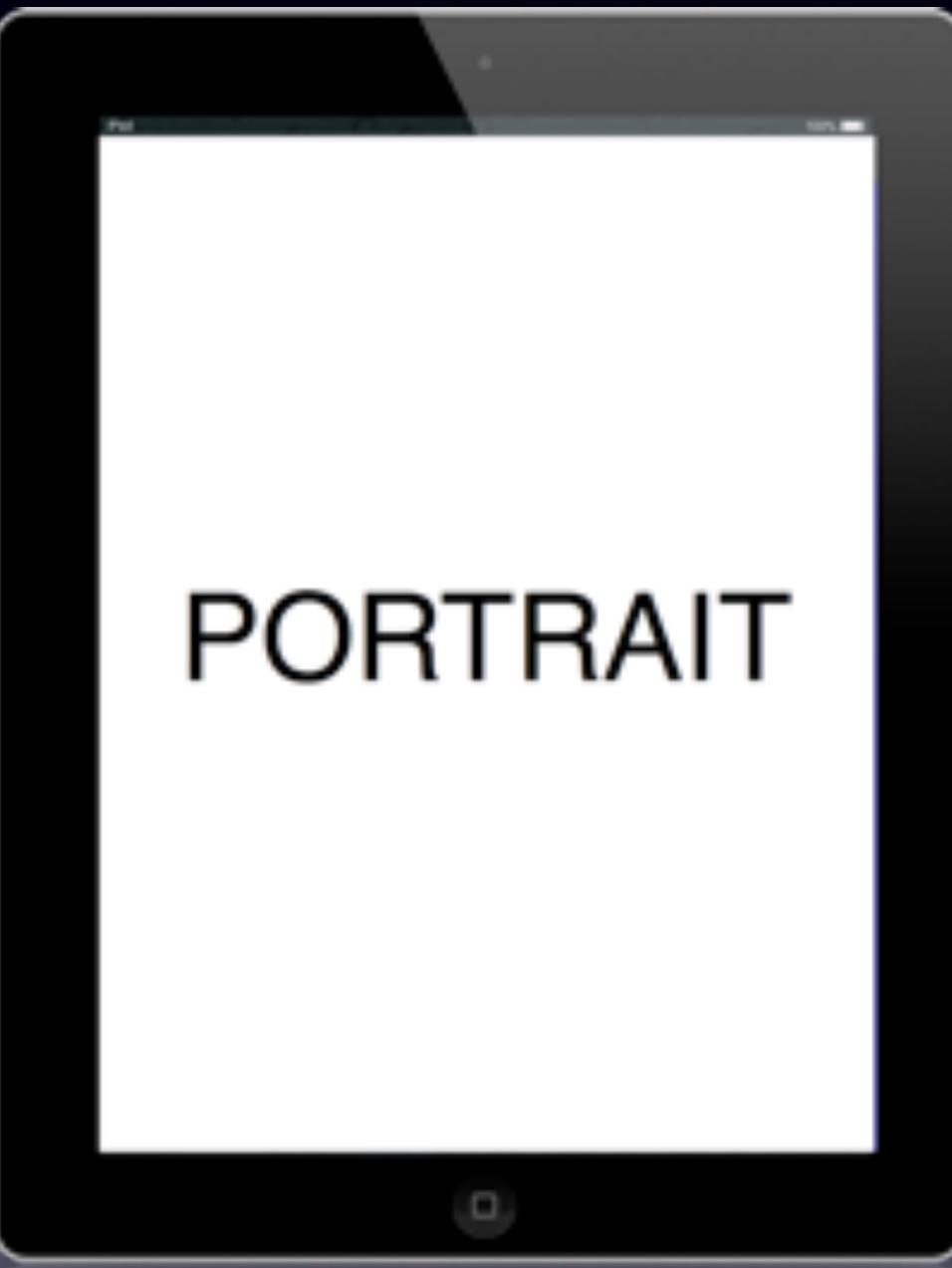
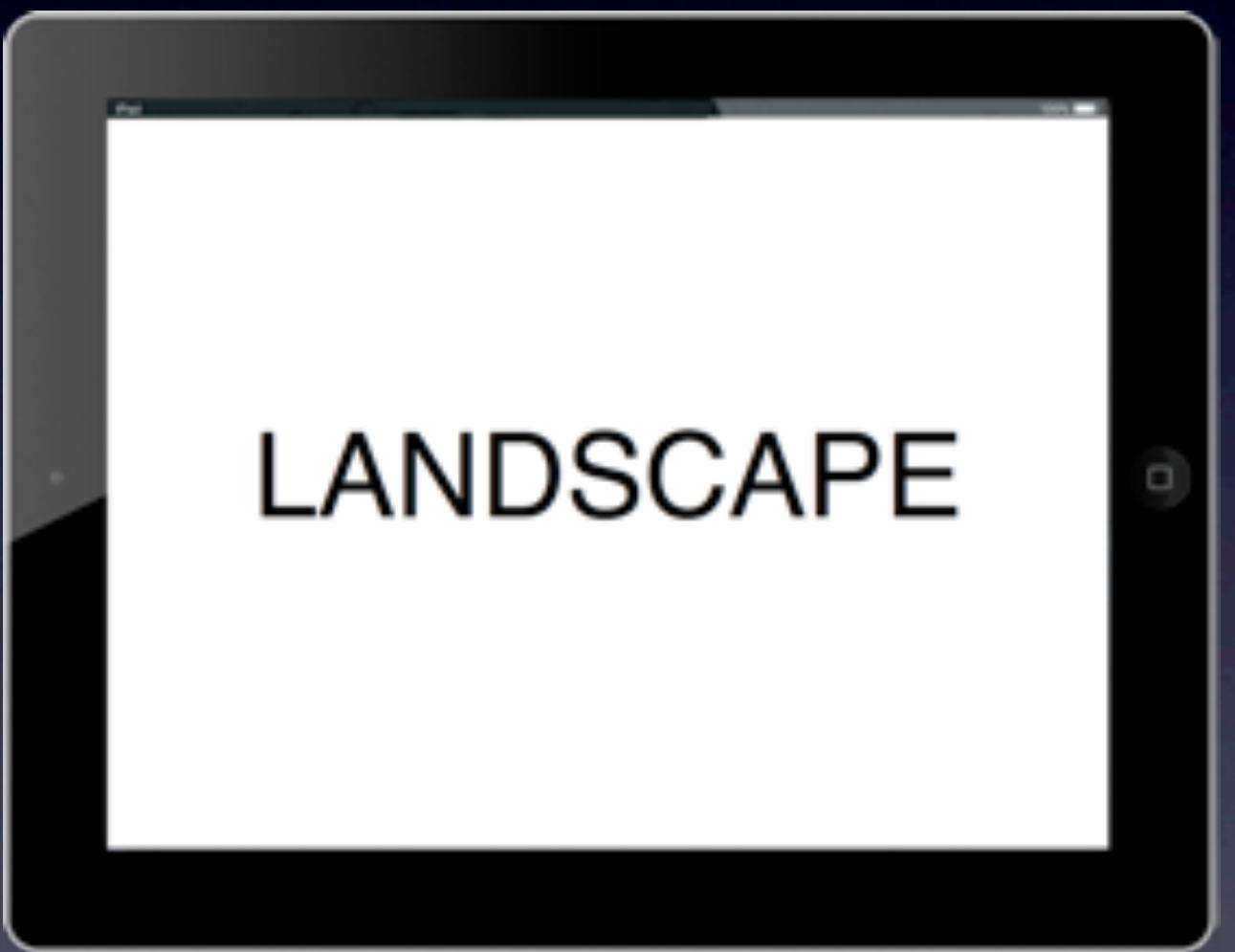
SplitViewNavigator



SplitViewNavigator

```
<s:SplitViewNavigator width="100%" height="100%">
    <s:ViewNavigator width="256" height="100%"
        firstView="views.v03.MasterView" />
    <s:ViewNavigator id="mainNavigator" width="100%" height="100%"
        firstView="views.v03.DetailView" />
</s:SplitViewNavigator>
```

Orientation Change



demo

app04_OrientationChange.mxml

Orientation Change

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
                 xmlns:s="library://ns.adobe.com/flex/spark" applicationDPI="160"
                 resizeMode="currentState = FlexGlobals.topLevelApplication.aspectRatio">
    <fx:Script>
        import mx.core.FlexGlobals;
    </fx:Script>
    <s:states>
        <s:State name="portrait" />
        <s:State name="landscape" />
    </s:states>
    <s:layout>
        <s:HorizontalLayout verticalAlign="middle" horizontalAlign="center" />
    </s:layout>
    <s:Label text.landscape="LANDSCAPE" text.portrait="PORTRAIT"    fontSize="120"/>
</s:Application>
```

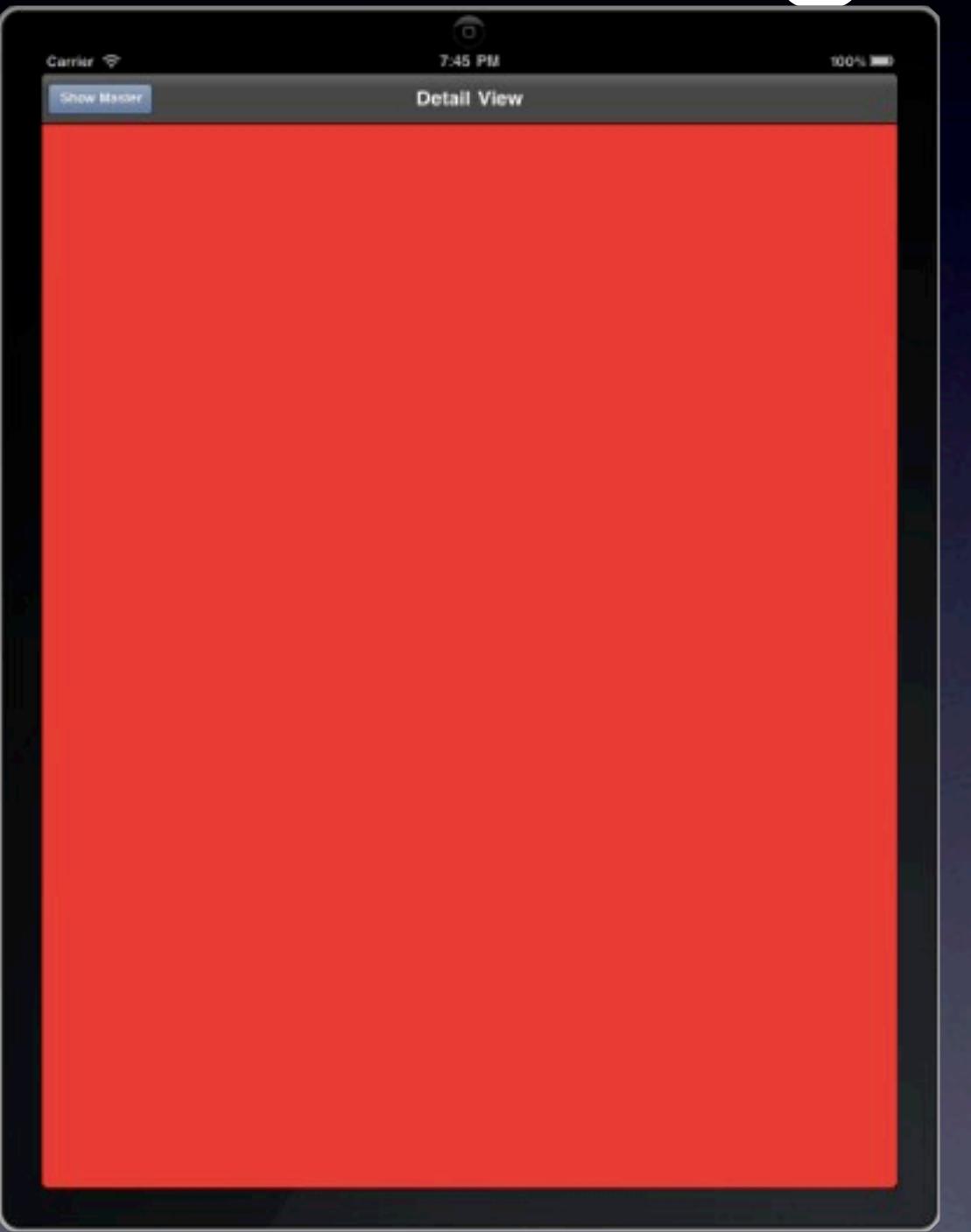
Orientation Change

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark" applicationDPI="160"
    resizeMode="currentState = FlexGlobals.topLevelApplication.aspectRatio">
    <fx:Script>
        import mx.core.FlexGlobals;
    </fx:Script>
    <s:states>
        <s:State name="portrait" />
        <s:State name="landscape" />
    </s:states>
    <s:layout>
        <s:HorizontalLayout verticalAlign="middle" horizontalAlign="center" />
    </s:layout>
    <s:Label text.landscape="LANDSCAPE" text.portrait="PORTRAIT"    fontSize="120"/>
</s:Application>
```

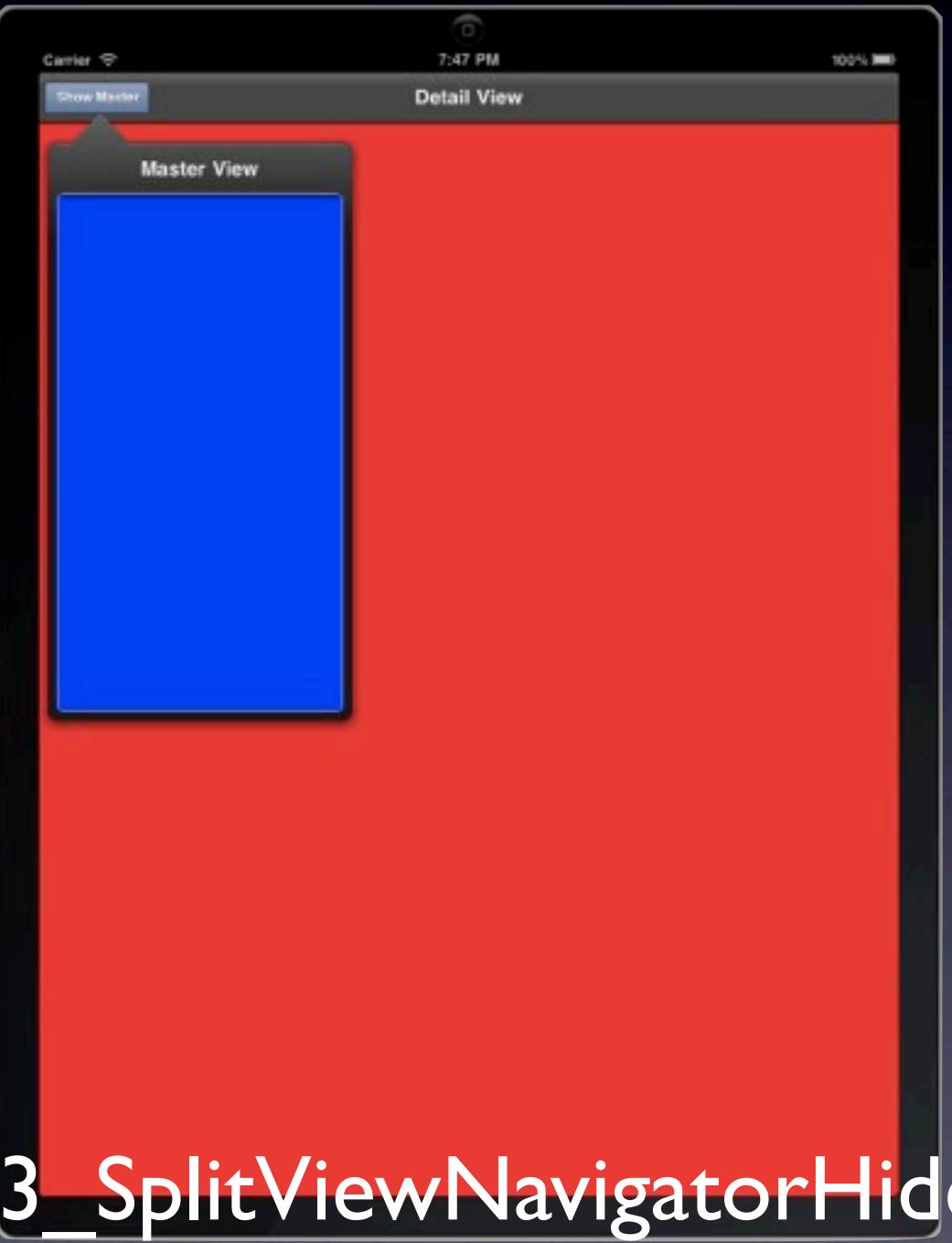
SplitViewNavigator



SplitViewNavigator



SplitViewNavigator



demo

Code

app03_SplitViewNavigatorHideLeftView

SplitViewNavigator

```
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark"
    xmlns:views="views.*"
    applicationDPI="160" >
    <views:SplitView width="100%" height="100%" />
</s:Application>
```

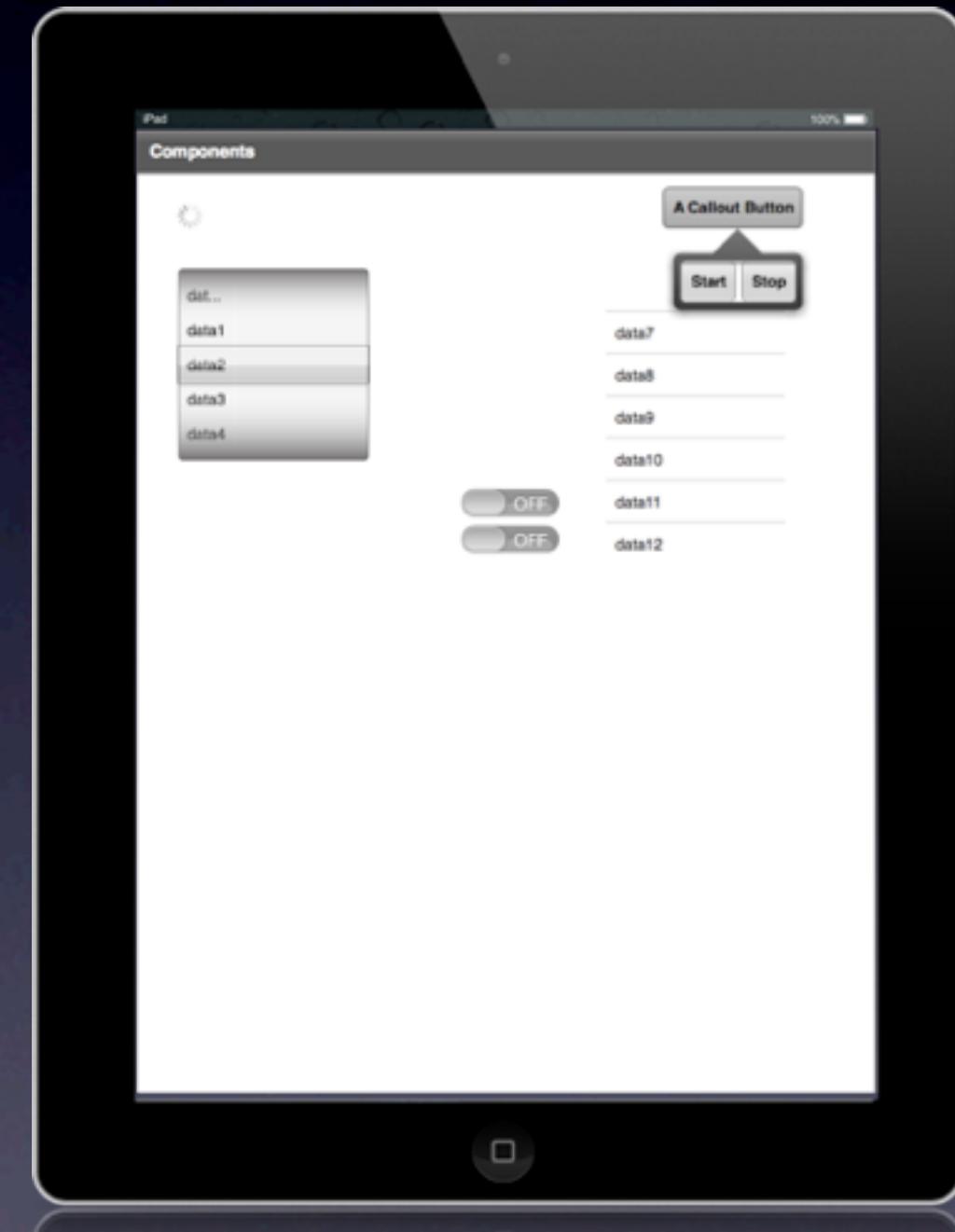
```
<s:SplitViewNavigator
    xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark"
    width="100%" height="100%"
    autoHideFirstViewNavigator="true"
    resize="currentState = FlexGlobals.topLevelApplication.aspectRatio">
    <s:states>
        <s:State name="portrait" />
        <s:State name="landscape" />
    </s:states>
    <s:ViewNavigator width="256" height="100%" height.portrait="500"
        firstView="views.RedView" />
        <s:ViewNavigator id="mainNavigator" width="100%" height="100%" firstView="views.BlueView"
    >
        <s:navigationContent.portrait>
            <s:Button id="navigatorButton" label="Show Red"
                click="showFirstViewNavigatorInPopUp(navigatorButton)" />
        </s:navigationContent.portrait>
    </s:ViewNavigator>
    <fx:Script>
        import mx.core.FlexGlobals;
    </fx:Script>
</s:SplitViewNavigator>
```

```
<s:SplitViewNavigator
    xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark"
    width="100%" height="100%"
    autoHideFirstViewNavigator="true"
    resize="currentState = FlexGlobals.topLevelApplication.aspectRatio">
    <s:states>
        <s:State name="portrait" />
        <s:State name="landscape" />
    </s:states>
    <s:ViewNavigator width="256" height="100%" height.portrait="500"
        firstView="views.RedView" />
    <s:ViewNavigator id="mainNavigator" width="100%" height="100%" firstView="views.BlueView"
    >
        <s:navigationContent.portrait>
            <s:Button id="navigatorButton" label="Show Red"
                click="showFirstViewNavigatorInPopUp(navigatorButton)" />
        </s:navigationContent.portrait>
    </s:ViewNavigator>
    <fx:Script>
        import mx.core.FlexGlobals;
    </fx:Script>
</s:SplitViewNavigator>
```

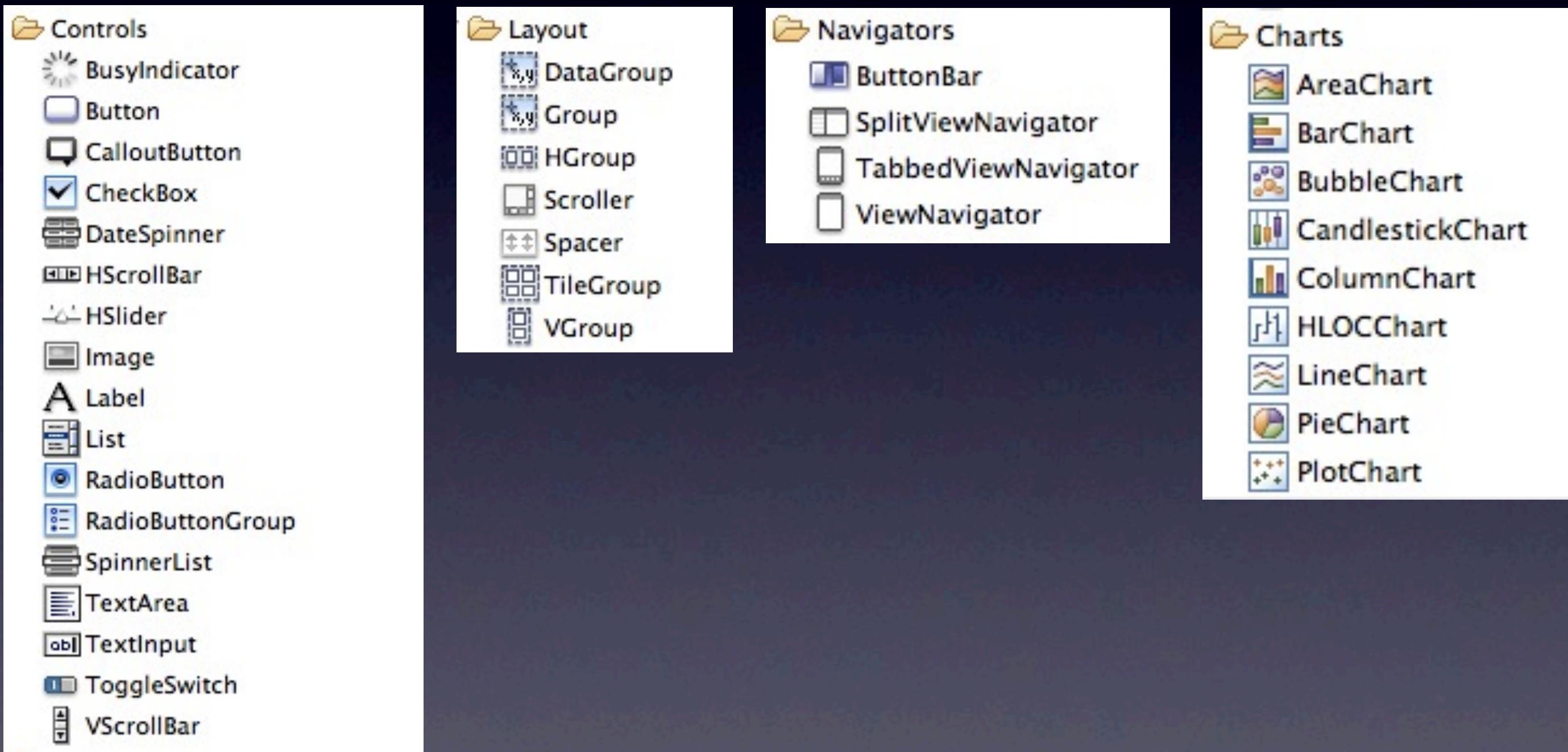
```
<s:SplitViewNavigator
    xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark"
    width="100%" height="100%"
    autoHideFirstViewNavigator="true"
    resize="currentState = FlexGlobals.topLevelApplication.aspectRatio">
    <s:states>
        <s:State name="portrait" />
        <s:State name="landscape" />
    </s:states>
    <s:ViewNavigator width="256" height="100%" height.portrait="500"
        firstView="views.RedView" />
    <s:ViewNavigator id="mainNavigator" width="100%" height="100%" firstView="views.BlueView"
    >
        <s:navigationContent.portrait>
            <s:Button id="navigatorButton" label="Show Red"
                click="showFirstViewNavigatorInPopUp(navigatorButton)" />
        </s:navigationContent.portrait>
    </s:ViewNavigator>
    <fx:Script>
        import mx.core.FlexGlobals;
    </fx:Script>
</s:SplitViewNavigator>
```

Components

- CalloutButton
- DateSpinner
- SpinnerList
- ToggleSwitch
- BusyIndicator



Flex SDK 4.6 Component



Recommend Components

Spark ActionBar

Spark BusyIndicator

Spark TabbedViewNavigator

Spark

TabbedViewNavigatorApplication

Spark View

Spark ViewMenu

Spark ViewNavigator

Spark ViewNavigatorApplication

Spark Button

Spark CheckBox

Spark DataGroup

Spark Group/HGroup/VGroup/TileGroup

Spark Image/BitmapImage

Spark Label

Spark List

Spark RadioButton/RadioButtonGroup

Spark SkinnableContainer Spark Scroller

Spark TextArea

Spark TextInput

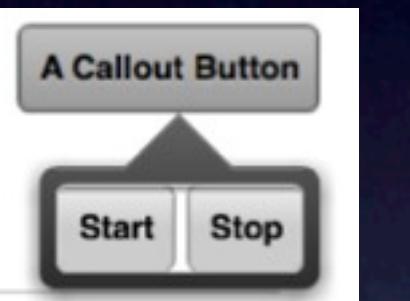
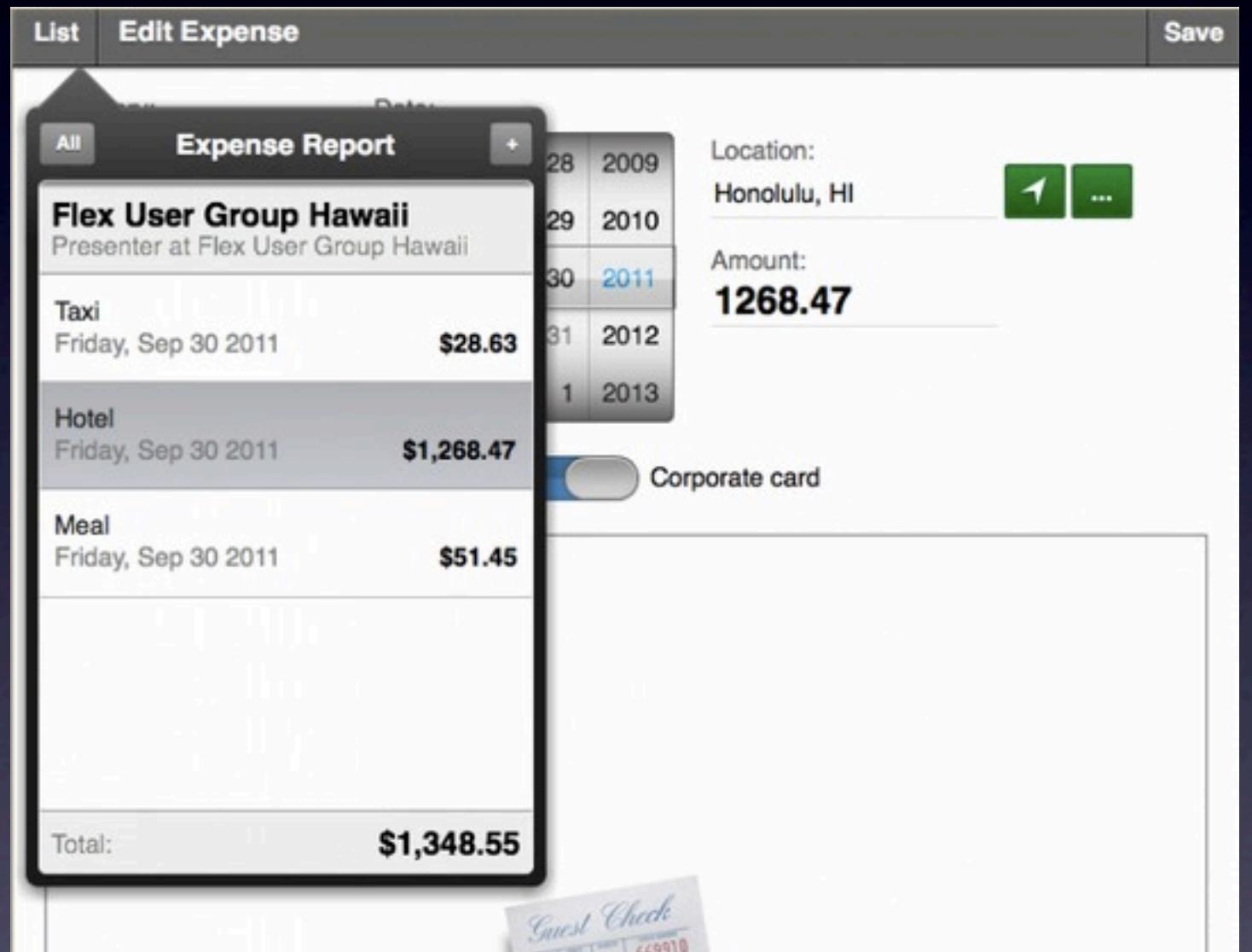
Discouraged Components

Spark DataGrid

Spark RichEditableText

Spark RichTex

CallOutButton



CalloutButton

```
<s:CalloutButton id="callout" x="547" y="15" label="A Callout Button"  
    horizontalPosition="end" verticalPosition="after">  
    <s:calloutLayout>  
        <s:HorizontalLayout/>  
    </s:calloutLayout>  
    <s:Button label="Start" click="busy.visible=true; callout.closeDropDown();"/>  
    <s:Button label="Stop" click="busy.visible=false;callout.closeDropDown();"/>  
</s:CalloutButton>
```

DateSpinner

February	11	2010
March	12	2011
April	13	2012
May	14	2013
June	15	2014

7	22	
8	23	AM
9	24	PM
10	25	
11	26	

Wed, Apr 11	7	22	
Thu, Apr 12	8	23	AM
Fri, Apr 13	9	24	PM
Sat, Apr 14	10	25	
Sun, Apr 15	11	26	

DATE (default)

TIME

DATE_AND_TIME

DateSelectorDisplayMode

DateSpinner

```
<s:DateSpinner displayMode="{dateDisplayMode.selectedItem.data}" />  
  
<s:DateSpinner displayMode="{DateSelectorDisplayMode.TIME}" />  
  
<s:DateSpinner displayMode="{DateSelectorDisplayMode.DATE_AND_TIME}" />
```



app20_DateSpinner

SpinnerList



SpinnerList

```
<s:SpinnerListContainer x="42" y="100" width="200" height="200">
  <s:SpinnerList id="spinnerList" height="100%" labelField="data">
    <s:ArrayList>
      <fx:Object data="data1"></fx:Object>
      <fx:Object data="data2"></fx:Object>
      <fx:Object data="data3"></fx:Object>
      <fx:Object data="data4"></fx:Object>
      <fx:Object data="data5"></fx:Object>
      <fx:Object data="data6"></fx:Object>
    </s:ArrayList>
  </s:SpinnerList>
</s:SpinnerListContainer>
```

SpinnerList

23	56	
0	57	
1	58	AM
2	59	PM
3	0	

SpinnerList

```
<s:SpinnerListContainer top="350" left="100">
    <s:SpinnerList typicalItem="100">
        <s:dataProvider>
            <s:NumericDataProvider minimum="0" maximum="23" stepSize="1"/>
        </s:dataProvider>
    </s:SpinnerList>
    <s:SpinnerList typicalItem="100">
        <s:dataProvider>
            <s:NumericDataProvider minimum="0" maximum="59" stepSize="1"/>
        </s:dataProvider>
    </s:SpinnerList>
    <s:SpinnerList typicalItem="100"
        dataProvider="{new ArrayList(['AM', 'PM'])}"
        wrapElements="false"/>
</s:SpinnerListContainer>
```

SpinnerList + IconItemRenderer



app2I_IconSpinnerList

SpinnerList + IconItemRenderer

```
<fx:Declarations>
  <s:ArrayCollection id="iconList">
    <fx:Object icon="@Embed('/assets/icons/spinner/flex_50x50.gif')"/>
    <fx:Object icon="@Embed('/assets/icons/spinner/acrobat_50x50.gif')"/>
    <fx:Object icon="@Embed('/assets/icons/spinner/flash-builder-48x48.png')"/>
    <fx:Object icon="@Embed('/assets/icons/spinner/flash_50x50.gif')"/>
    <fx:Object icon="@Embed('/assets/icons/spinner/flash_player_50x50.gif')"/>
    <fx:Object icon="@Embed('/assets/icons/spinner/photoshop_50x50.gif')"/>
  </s:ArrayCollection>
</fx:Declarations>
```

SpinnerList + IconItemRenderer

```
<fx:Declarations>
    <fx:Component className="CustomIconItemRenderer">
        <s:IconItemRenderer labelField="" iconField="icon"/>
    </fx:Component>
</fx:Declarations>
<s:SpinnerListContainer>
    <s:SpinnerList width="90" dataProvider="{iconList}" selectedIndex="0"
        itemRenderer="CustomIconItemRenderer" />
    <s:SpinnerList width="90" dataProvider="{iconList}" selectedIndex="2"
        itemRenderer="CustomIconItemRenderer" />
    <s:SpinnerList width="90" dataProvider="{iconList}" selectedIndex="1"
        itemRenderer="CustomIconItemRenderer" />
</s:SpinnerListContainer>
```

Keyboard



demo

app10_keyboard

Keyboard

```
<s:TextInput prompt="contact" softKeyboardType="contact"/>  
<s:TextInput prompt="default" softKeyboardType="default" />  
<s:TextInput prompt="email" softKeyboardType="email"/>  
<s:TextInput prompt="number" softKeyboardType="number"/>  
<s:TextInput prompt="punctuation" softKeyboardType="punctuation"/>
```

default



Keyboard

```
<s:TextInput prompt="contact"  
<s:TextInput prompt="default"  
<s:TextInput prompt="email"  
<s:TextInput prompt="number"  
<s:TextInput prompt="punctuation"
```

```
softKeyboardType="contact"/>  
softKeyboardType="default" />  
softKeyboardType="email"/>  
softKeyboardType="number"/>  
softKeyboardType="punctuation"/>
```

email



Keyboard

```
<s:TextInput prompt="contact" softKeyboardType="contact"/>  
<s:TextInput prompt="default" softKeyboardType="default" />  
<s:TextInput prompt="email" softKeyboardType="email"/>  
<s:TextInput prompt="number" softKeyboardType="number"/>  
<s:TextInput prompt="punctuation" softKeyboardType="punctuation"/>
```

number



Keyboard

```
<s:Application resizeForSoftKeyboard="true">
```

```
<s:Callout moveForSoftKeyboard="true">
```

demo

Code

app10_keyboardAndPopup.mxml
InputCallout.mxml

Keyboard Events

- softKeyboardActivating
- softKeyboardActivate
- softKeyboardDeactivate

flash.ui.Multitouch.inputMode

- MultitouchInputMode.NONE
→ Mouse Events Only
- MultitouchInputMode.TOUCH_POINT
→ Mouse and Touch Events
- MultitouchInputMode.GESTURE
→ Mouse and Gesture Events

Touch Events

- touchBegin
- touchDelay
- touchEnd
- touchInteractionEnd
- touchInteractionStart
- touchInteractionStarting
- touchMove
- touchOver
- touchRollOut
- touchRollOver
- touchTap

demo

Code

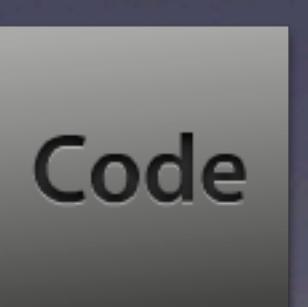
app08_Touch.mxml

Touch Events

```
<s:Group width="100%" height="100%">
    <s:touchBegin>
        var id:int = event.touchPointID; // to track multiple touches at once
        circle.x = event.localX - 70;
        circle.y = event.localY - 70;
        circle.visible = true;
    </s:touchBegin>
    <s:touchMove>
        circle.x = event.localX - 70;
        circle.y = event.localY - 70;
    </s:touchMove>
    <s:touchEnd>
        circle.visible = false;
    </s:touchEnd>
</s:Group>
```

Gestures

- gesturePan
- gesturePressAndTap
- gestureRotate
- gestureSwipe
- gestureTwoFingerTap
- gestureZoom



app09_Gestures.mxml

Gestures

```
<s:Group width="100%" height="100%">
    <s:gesturePan>
        img.x += event.offsetX;
        img.y += event.offsetY;
    </s:gesturePan>
    <s:gestureZoom>
        img.transformAround(new Vector3D(event.localX, event.localY, 0),
                            new Vector3D(img.scaleX * event.scaleX,
                                         img.scaleY * event.scaleY, 0)
        );
    </s:gestureZoom>
    <s:Image id="img" source="@Embed('/assets/apacheflex_fc.jpg')"/>
</s:Group>
```

Passing Data Around Views

- ViewNavigator has build in mechanism to pass data to views
- Each view has a *data* attribute

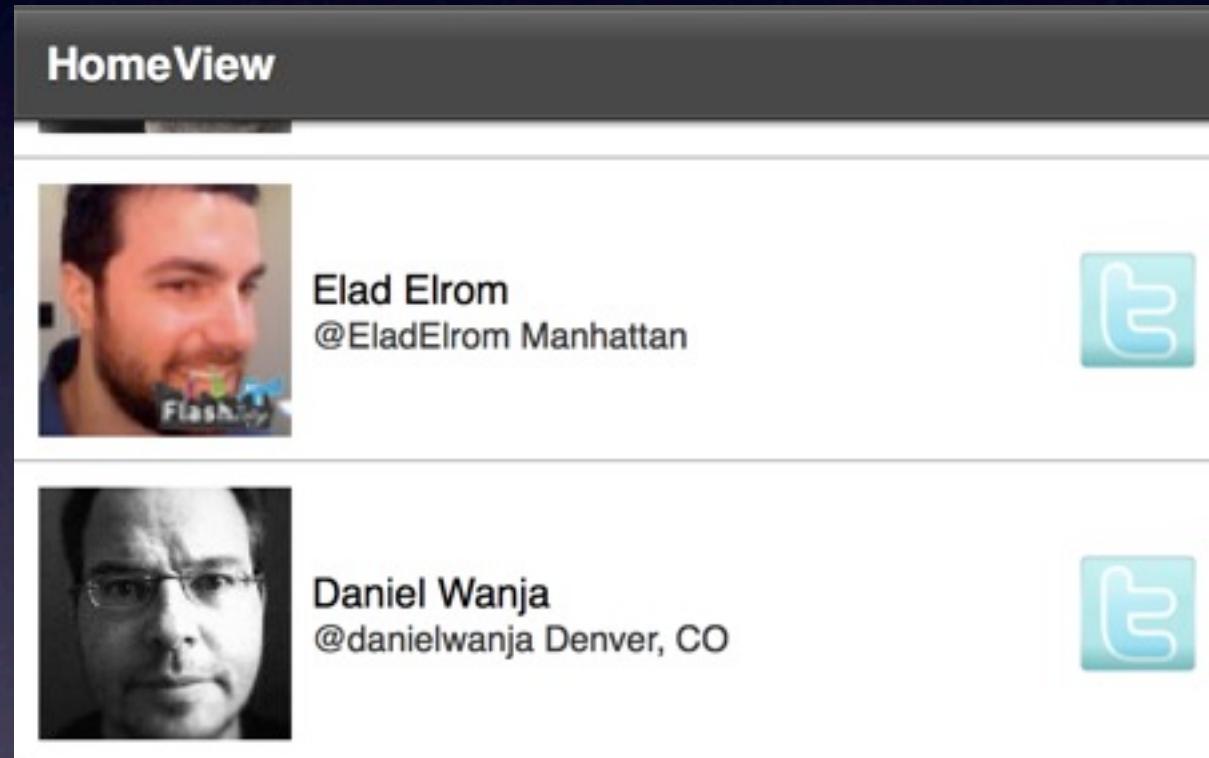


demo

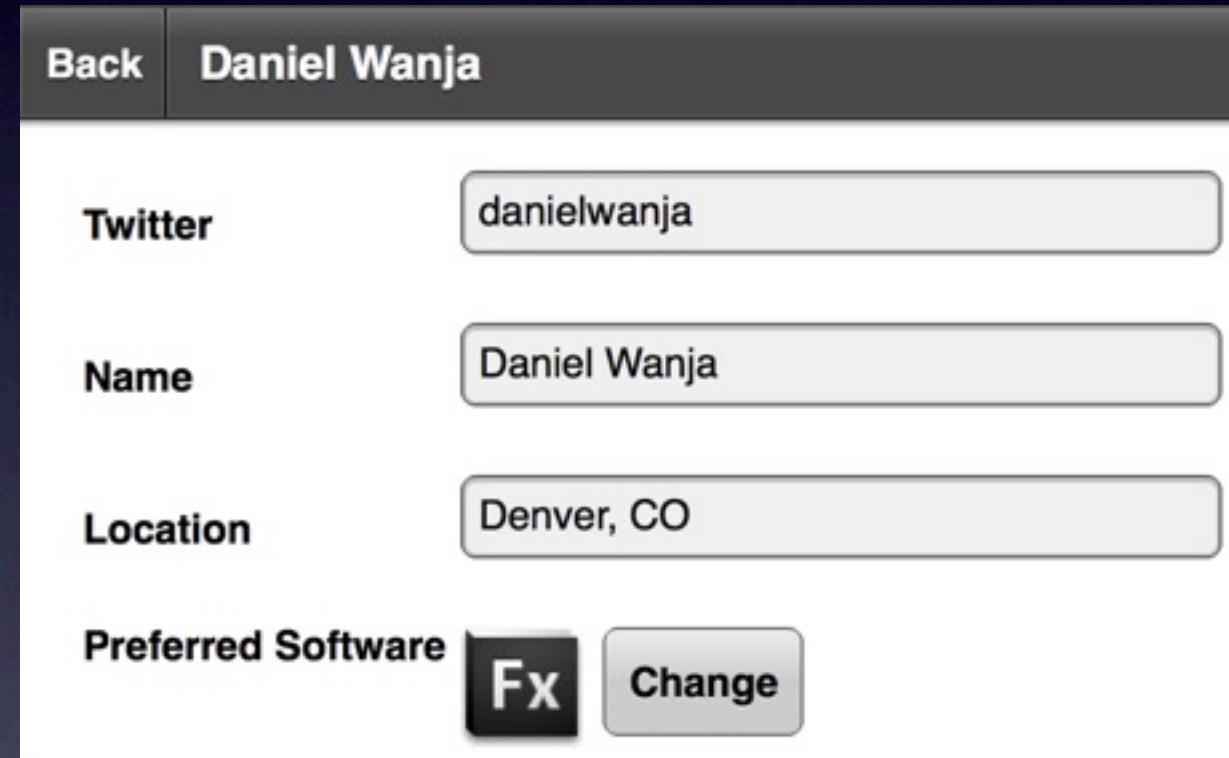
app || PassingData

Passing Data to a View

ListView



DetailView



`navigator.pushView(DetailView, list.getSelectedItem) ➔`

Returning Data

DetailView

Back Daniel Wanja

Twitter	danielwanja
Name	Daniel Wanja
Location	Denver, CO
Preferred Software	 

SelectView

Cancel SelectView Save

 Flash Player
 Photoshop
 Flex
 Acrobat

pushView() →

← popView()

Returning Data

SelectView

popView() →

```
override public function createReturnObject():Object
{
    return selectedProduct;
}
```

DetailView



```
<s:add>
var returnedObject:ViewReturnObject = navigator.poppedViewReturnedObject;
if (returnedObject&&returnedObject.object) {
    data.software = returnedObject.object;
    img.source = data.software.icon;
}
</s:add>
```

ItemRenderers

- Performance issues with large list
- Don't use binding (for large list)

demo

Code

app12_iconRenderer

ItemRenderers

```
<s>List width="100%" dataProvider="{data}" height="100%">
  <s:itemRenderer>
    <fx:Component>
      <s:IconItemRenderer height="120" labelField="name"
        iconField="photo" iconHeight="100" iconWidth="100"
        messageFunction="getMessage"
        decorator="@Embed('/assets/icons/twitter_icon_50.png')">
        <fx:Script>
          protected function getMessage(o:Object):String { return "@" +
o.thandle + " " + o.location; }
        </fx:Script>
      </s:IconItemRenderer>
    </fx:Component>
  </s:itemRenderer>
</s>List>
```

Scroller

```
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark" applicationDPI="160">
    <s:Scroller width="100%" height="100%">
        <s:Group>
            <s:Image source="@Embed('/assets/the_last_photo_of_the_eiffel_tower.jpg')"
                width="5480" height="3596" />
        </s:Group>
    </s:Scroller>
</s:Application>
```



appl4_Scroller

StagedWebView

Creation Complete

```
webView = new StageWebView();
webView.stage = this.stage;
resizeWebView();
webView.loadURL("http://google.com");
```

Resize

```
var p:Point = new Point(0, 0);
p = this.localToGlobal(p);
webView.viewPort = new Rectangle(p.x, p.y, this.width,
this.height);
```

Remove

```
var webView:StageWebView = this.webView;
this.webView = null;
webView.dispose();
```

demo

app13_StageWebView

StageWebView

Additional Methods

```
webView.historyBack()  
webView.historyForward()
```

Events

```
webView.addEventListener(Event.COMPLETE,handleLoad);  
webView.addEventListener( LocationChangeEvent.LOCATION_CHANGEING,  
                           handleLocationChanging );
```

StageWebView

ActionScript to JavaScript Communication

```
webView.loadURL("javascript:alert('Flex talks to Javascript')");
```

JavaScript to ActionScript Communication

- less clean
- JavaScript sets `document.location` to pass some info as string
- Action use `LocationChangeEvent`, `preventDefault`
- The reads the that info from `event.location`

Maps

- Google Maps with StagedWebView
- MapQuest Flex Components



app22_MapQuest

MapQuest

```
<tilemap:TilemapComponent id="map" width="100%"  
height="100%" key="This%7IsCluu2n1uSecret-hw70u"  
zoom="4"/>
```

```
geocoder = new Geocoder(map.tileMap);  
  
map.addControl(new SMLargeZoomControl());  
map.addControl(new SMViewControl());  
map.addControl(new MouseWheelZoomControl());
```

<http://developer.mapquest.com/web/products/featured/as3-flex-flash-mobile>

MapQuest

```
geocoder.addEventListener(GeocoderEvent.GEOCODE_RESPONSE, onGeocodeResponse);  
geocoder.addEventListener(GeocoderEvent.GEOCODE_ERROR_EVENT, onGeocodeError);  
geocoder.addEventListener(GeocoderEvent.HTTP_ERROR_EVENT, this.onHttpError);
```

```
geocoder.geocode("Apple Store Aspen Grove, Littleton, CO");  
geocoder.geocode("10345 Park Meadows Drive, Lone Tree, CO");
```

mx:Chart



mx:Chart

```
[Bindable] private var serie:ArrayCollection = new ArrayCollection;  
serie.addItem({rpm:rpm, response_time:run.last_response_time});
```

mx:Chart

```
<mx:LineChart id="chart" dataProvider="{serie}" showDataTips="true" >
  <mx:verticalAxis>
    <mx:LinearAxis id="vAxis"/>
  </mx:verticalAxis>
  <mx:series>
    <mx:LineSeries yField="rpm">
      <mx:fill>
        <s:SolidColor color="#FF0000"/>
      </mx:fill>
    </mx:LineSeries>
  </mx:series>
</mx:LineChart>
```

mx:Chart

```
<mx:ColumnChart id="chart2" dataProvider="{serie}" showDataTips="true">
    <mx:verticalAxis>
        <mx:LinearAxis id="vAxis2"/>
    </mx:verticalAxis>
    <mx:series>
        <mx:ColumnSeries yField="response_time" />
    </mx:series>
</mx:ColumnChart>
```

References

- DEVELOPING MOBILE APPLICATIONS WITH FLEX AND FLASH BUILDER 4.6
[http://help.adobe.com/en_US/flex/mobileapps/
developing_mobile_apps_flex_4.6.pdf](http://help.adobe.com/en_US/flex/mobileapps/developing_mobile_apps_flex_4.6.pdf)

Where we go from here?

- 2D
- 3D
- Suite of Native Extensions

Go build stuff!



Swiz: initialize

```
<swiz:Swiz beanProviders=" {[Config]}>
  <swiz:config>
    <swiz:SwizConfig strict="true"
      eventPackages="events,flash.events"
      viewPackages="views"/>
  </swiz:config>
</swiz:Swiz>
```

Swiz: config

```
<swiz:BeanProvider
    xmlns:swiz="http://swiz.swizframework.org"
    xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:models="models.*"
    xmlns:controllers="controllers.*"
    >
    <models:Model id="model" />
    <controllers:Controller id="controller" />
</swiz:BeanProvider>
```

Swiz: event

```
dispatchEvent(new SnapshotEvent(SnapshotEvent.TAKE, model.url,{width:300,  
height:300}));
```

Swiz: controller

```
public class Controller
{
    [Inject] public var model:Model;

    [EventHandler(event='SnapshotEvent.TAKE', properties="url,data")]
    public function takeSnapshot(url:String, data:Object):void {
        // do stuff
    }
}
```



MORE STUFF...

ViewTransitionBase

- CrossFadeViewTransition
- FlipViewTransition
- SlideViewTransition
- ZoomViewTransition

Native Extension

- For whatever need that is not fulfilled by the Flex SDK
- Can include native Objective-C code with your App.
- I.e. Vibration, Twitter integration, Game Center integration

Anatomic/Ergonomic/Physical Considerations

- Hands are in the way
- Looking down on table (viewing area)
- Keyboard hides the bottom half of the screen...so don't put input fields there (i.e. a filter at the bottom of a list)
- ...

DPI: 160, 240, 360

- Downscale .vs. Upscale conflicting approach (jpg/png .vs. vector graphics)
- AIR 3.3, iOS SDK 5.1
- override RuntimeDPIProvider
- Set runtimeDPIProvider on your Application

Deploying APPS

- iOS...arg! it's slow...but works
- Android...Yea!

Native Extension

- Beyond the scope of this talk...It's there and useful if you want to have functionality that's not provided by the SDK

Building On The Command Line

- compile (mxml to swf)
- package (swf to ipa)
- install app on iOS Simulator
- launch app on iOS Simulator

Compile

```
$ mxmclc +configname=airmobile -compiler.library-path+=..libs -swf-version=16  
myapp.mxml
```

Package for Simulator

```
adt -package -target ipa-test-interpreter-simulator -storetype pkcs12 -keystore  
cert.p12 -storepass secret myapp myapp.xml myapp.swf
```

→ myapp.ipa

Package for iPad

```
adt -package -target ipa-test -storetype pkcs12 -keystore cert.p12 -storepass a  
-provision.mobileprovision adt -package -target ipa-test -storetype pkcs12 -  
keystore cert.p12 -storepass a -provisioning-profile provision.mobileprovision  
myapp myapp-app.xml myapp.swf
```

→ myapp.ipa

Move to iOS Simulator

```
adt -installApp -platform ios -platformsdk /Developer/Platforms/  
iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator4.3.sdk -device ios-  
simulator -package myapp.ipa
```

```
adt -launchApp -platform ios -platformsdk /Developer/Platforms/  
iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator4.3.sdk -device ios-  
simulator -appid com.n-so.myapp
```

Google Map

- Use StageWebView
 - or use the e-skimo library

<http://e-skimo.com/>

Google Map

```
<pia:GMap id="map" width="100%" height="100%"  
zoom="{zoomSlider.value}"  
complete="map_completeHandler(event)"  
error="SkinnableAlert.show(event.toString(), 'Loading  
Error')"/>
```

```
map.setCenter(39.545529, -104.87031);  
map.zoom = 15;
```

```
map.addMarker(parseFloat(latMarker.text),parseFloat(lngMarker.text),  
'360Flex','Custom description', true)
```