Patterns and Best Practices for Building Applications with ArcGIS API for Flex

Antony Jayaprakash

Mansour Raad

Introductions

- Who are we?
  - Members of the ArcGIS API for Flex team
    - Antony – Product Engineer
    - Mansour (non-recovering Flex addict)
- Who are you?
  - Interested in developing applications with ArcGIS API for Flex
  - In “need” of a Vulcan Mind Meld

Schedule

- Today we will cover
  - Data preparation
  - User experience considerations
  - Implementation paths

- We will answer questions at the end of the session?

Please complete the session survey!

Great Web Applications

Characteristics

- User experience
  - Great cartography
  - Multi-scale
  - Fast
  - Informative
  - Easy to use
  - Meet users expectation
- Implementation
  - Modular
  - Testable

Web Applications

Development Path

- Prepare data
- Publish services
- Design user experience
- Implement design
- Test application
- Deploy application

Data Preparation

Web Applications

Data Preparation

- Applications are composed of layers.
- Layer is based on a map service.
- Map service is published using a map document.
- Map documents are authored in ArcMap

Base maps

- Different types
- Use ArcMap to create
- Pay attention to
  - Image types
  - Tiling scheme
  - Coordinate system

Operational layers

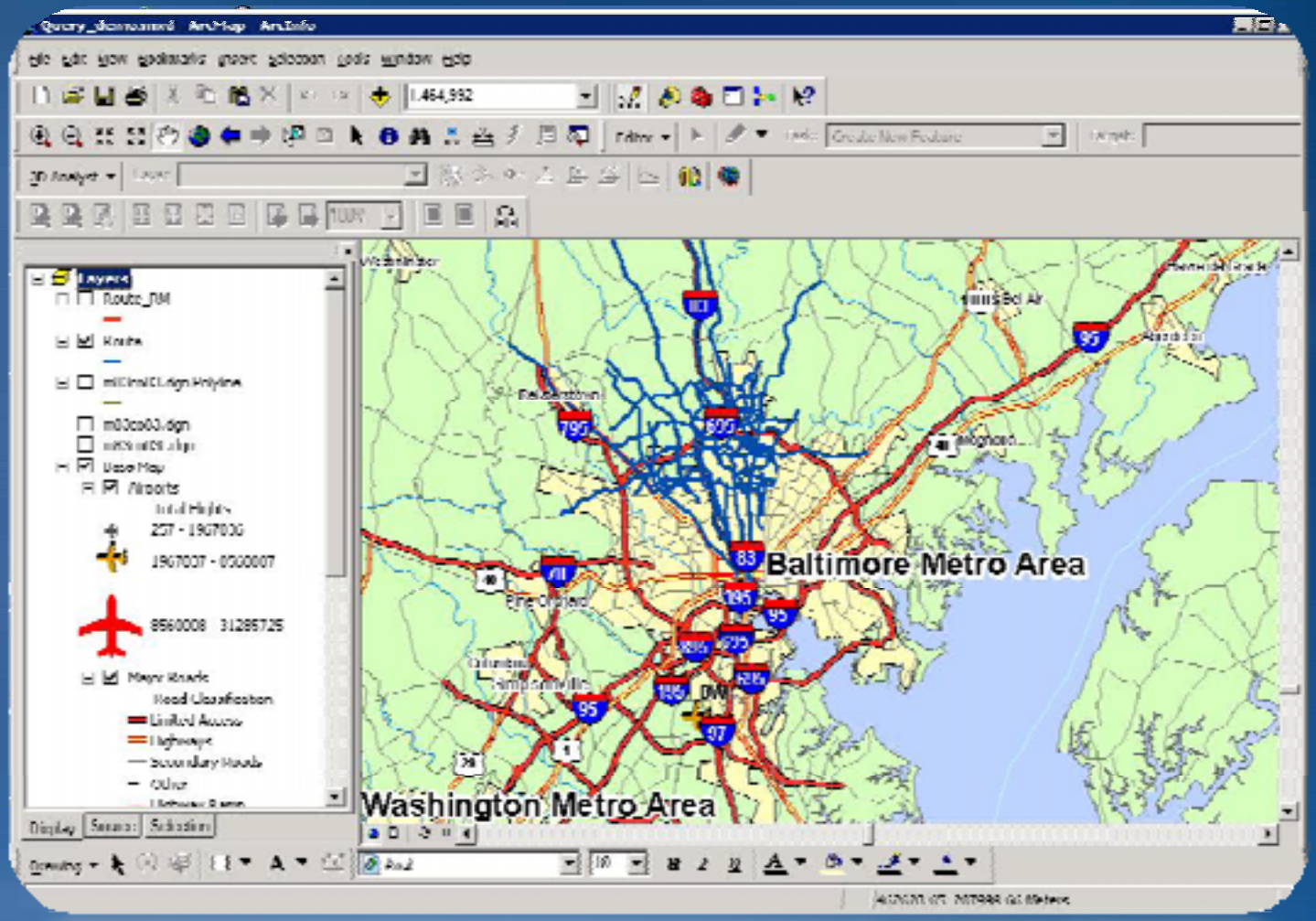
- Results of operations
- Graphics
- Dynamic images
- Cached tiles
- Scale dependent

User Experience

We Applications

User Experience

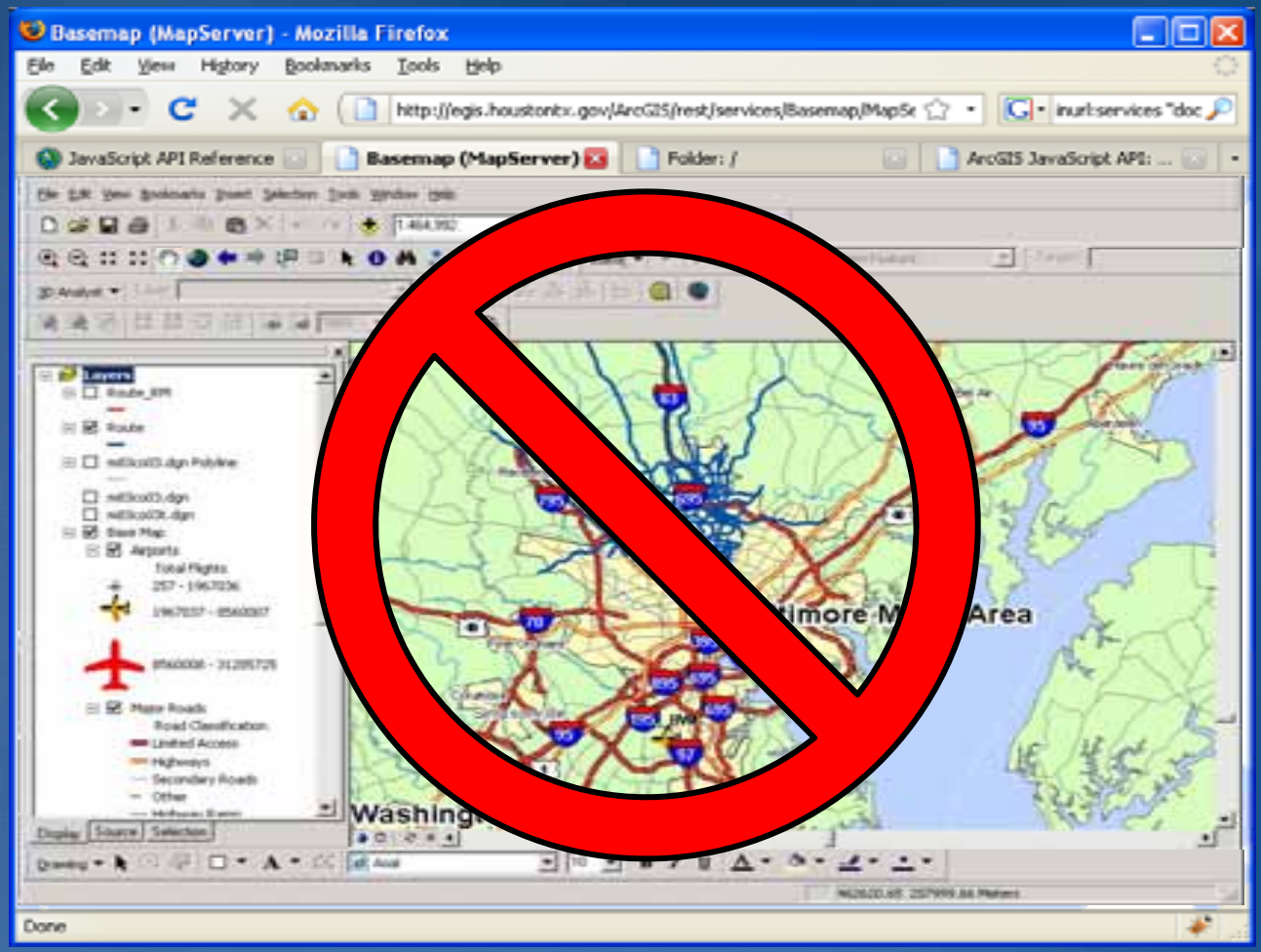
ArcMap - A great desktop application



We Applications

User Experience

ArcMap user experience in a web browser

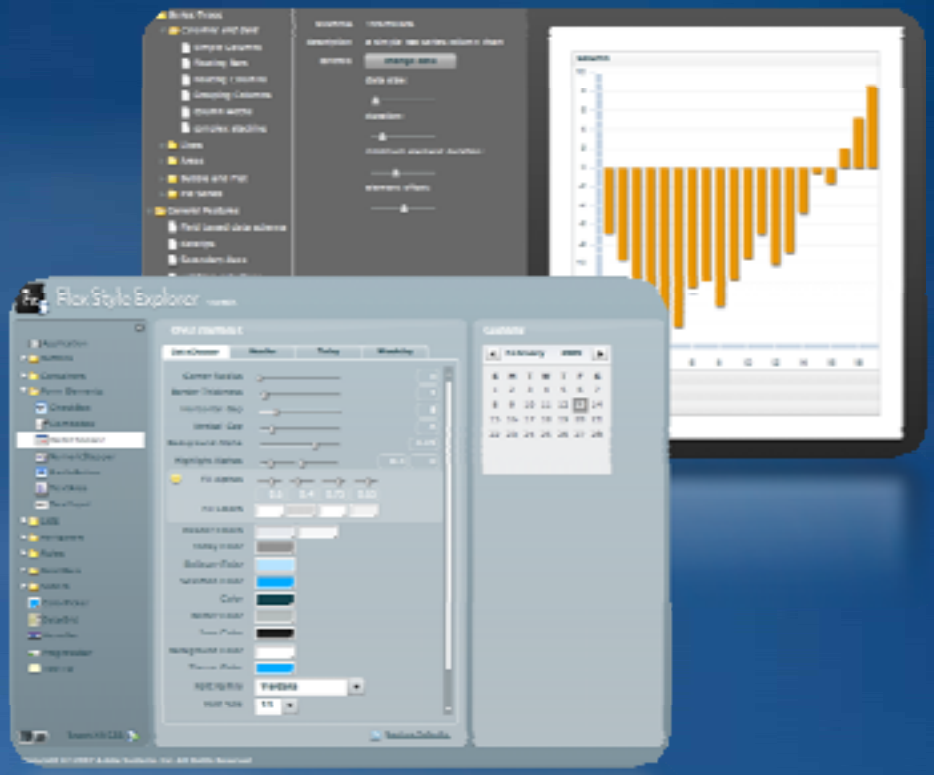


Not a good web experience …

Designing User Experience

Tools

- **Wire framing**
  - Balsamiq, iPlotz
- **Flex explorers**
  - Component explorer
  - Style explorer
  - Charts explorer
  - Effects explorer
  - Filter explorer
  - Button skin explorer
- **Flex interface guide**
  - http://www.adobe.com/devnet/flex/?navID=fig
- **Adobe experience design**
  - http://xd.adobe.com/



FLEX PRACTICES

FLEX PRACTICES

Trace

```
private function doSomething() : void
{
        trace( “Entering doSometing”);
        var foobar : String = someOtherFunction();
        trace( “foobar = “, foobar );
        trace( “Leaving doSomething”);
}
```

Trace

- Ye good olde fashion debugging
- System.out.println(“I am here !”);
- Output is in FlexBuilder Console
- Stripped out on release build

BUILD RELEASE IN ACTION

Logging

- In Flex Framework
- Logger
  - ILogger
  - Log
- LogTarget
  - TraceTarget
  - LocalConnectionTarget

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="horizontal">
3       <mx:Script>
4           <![CDATA[
5               import mx.logging.Log;
6               import mx.logging.ILogger;

7               private var m_logger:ILogger = Log.getLogger("MyLogApp");

8               private function logInfo_clickHandler():void
9               {
10                  if (Log.isInfo())
11                  {
12                      m_logger.info("This is an info message");
13                  }
14              }
15          ]]>
16      </mx:Script>
17      <mx:TraceTarget/>
18      <mx:Button label="Log Info" click="logInfo_clickHandler()"/>
19  </mx:Application>
```

LOGGING IN ACTION

Logging Targets

- LogBook
  - http://code.google.com/p/cimlogbook/
- DeMonsterDebugger
  - http://www.demonsterdebugger.com/

LogBook

LocalConnection

```
1   private var m_logger : ILogger = Log.getLogger( "LogApp" );
2   private var m_localConnectionTarget : LocalConnectionTarget;
3
4   private function toggleLogging() : void
5   {
6       if( m_localConnectionTarget )
7       {
8           Log.removeTarget( m_localConnectionTarget );
9           m_localConnectionTarget = null;
10      }
11      else
12      {
13          m_localConnectionTarget = new LocalConnectionTarget();
14          Log.addTarget( m_localConnectionTarget );
15      }
16  }
```

LOGBOOK IN ACTION

DeMonsterDebugger

LocalConnection

DEMONSTER IN ACTION

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
3       layout="vertical"
4       creationComplete="creationCompleteHandler()">
5       <mx:Script>
6           <![CDATA[
7               import nl.demonsters.debugger.MonsterDebugger;
8               public var hello : Object = {hello:"world"};

9               private var debugger : MonsterDebugger;
10
11              private function creationCompleteHandler() : void
12              {
13                  debugger = new MonsterDebugger( this );
14              }
15          ]]>
16      </mx:Script>
17      <mx:Label id="myLabel" text="Hello, Monster !"/>
18      <mx:Button id="myButton" label="Debug">
19          <mx:click>
20              <![CDATA[
21                  MonsterDebugger.trace( this, hello);
22              ]]>
23          </mx:click>
24      </mx:Button>
25  </mx:Application>
```

Unit Testing

- Test Driven Development (TDD)
- Flex Unit
  - http://code.google.com/p/as3flexunitlib/
- Fluint
  - http://code.google.com/p/fluint/
  - Same as flex unit but enables UI testing

Projection Interface

```
1  public interface IProjection
2  {

3     function fromGeographic( mapPoint : MapPoint) : MapPoint;

4     function toGeographic( mapPoint : MapPoint) : MapPoint;

5  }
```

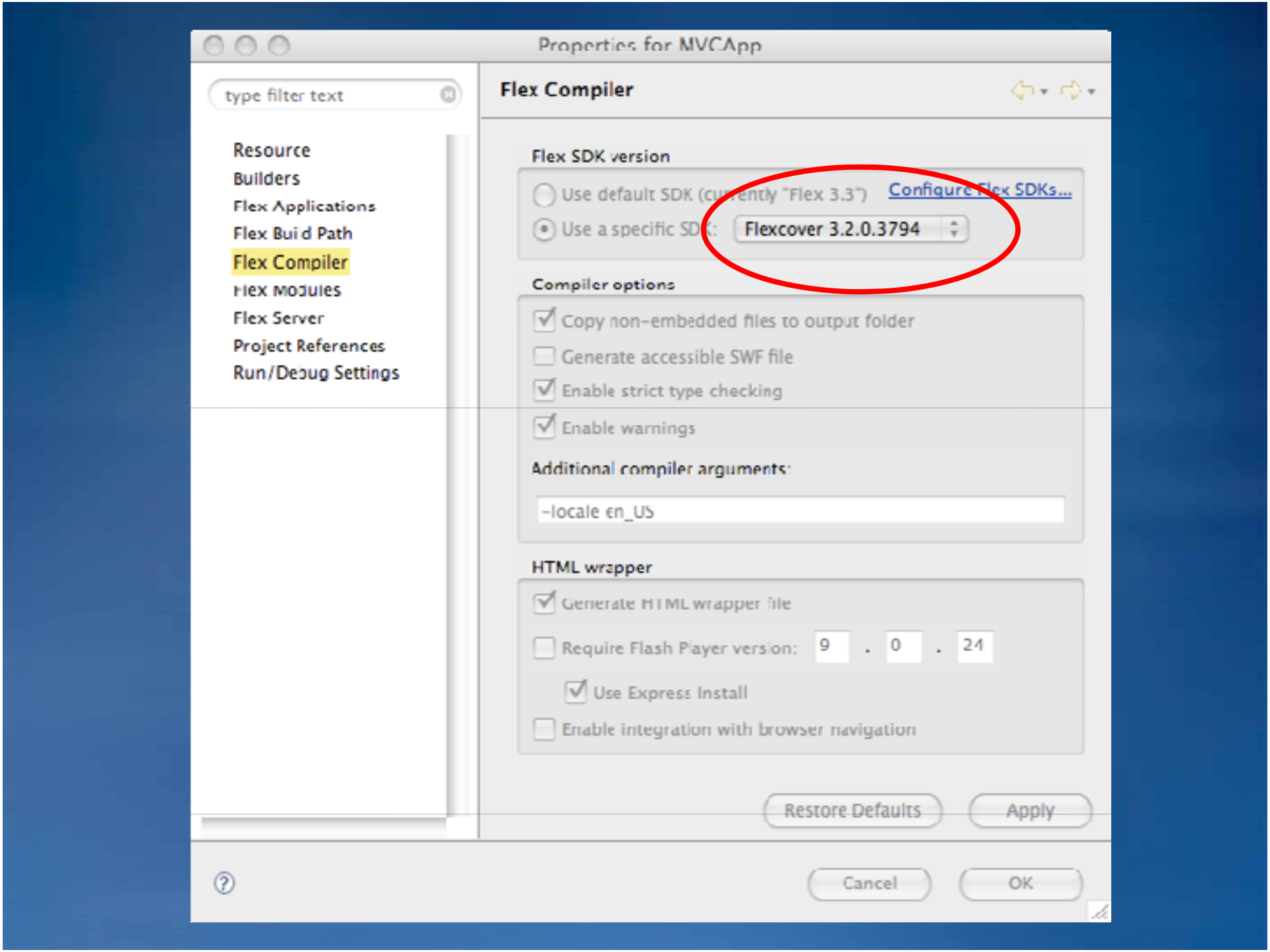
```
1   package com.esri.test
2   {
3       import com.esri.ags.geometry.MapPoint;
4       import com.esri.ags.projection.IProjection;
5       import com.esri.ags.projection.Mercator;
6
7       import net.digitalprimates.fluint.tests.TestCase;

8       public class MercatorTest extends TestCase {
9
10          public function testMercator() : void
11          {
12              const projection : IProjection = new Mercator();
13              const lat : Number = 37.5;
14              const lon : Number = -122.3;
15              const latlon1 : MapPoint = new MapPoint(lon,lat);
16              const meters : MapPoint = projection.fromGeographic( latlon1 );
17              const latlon2 : MapPoint = projection.toGeographic( meters );
18              assertTrue( Math.abs( lon – latlon2.x) < 0.000001);
19              assertTrue( Math.abs( lat – latlon2.y) < 0.000001);
20          }
21      }
22  }
```

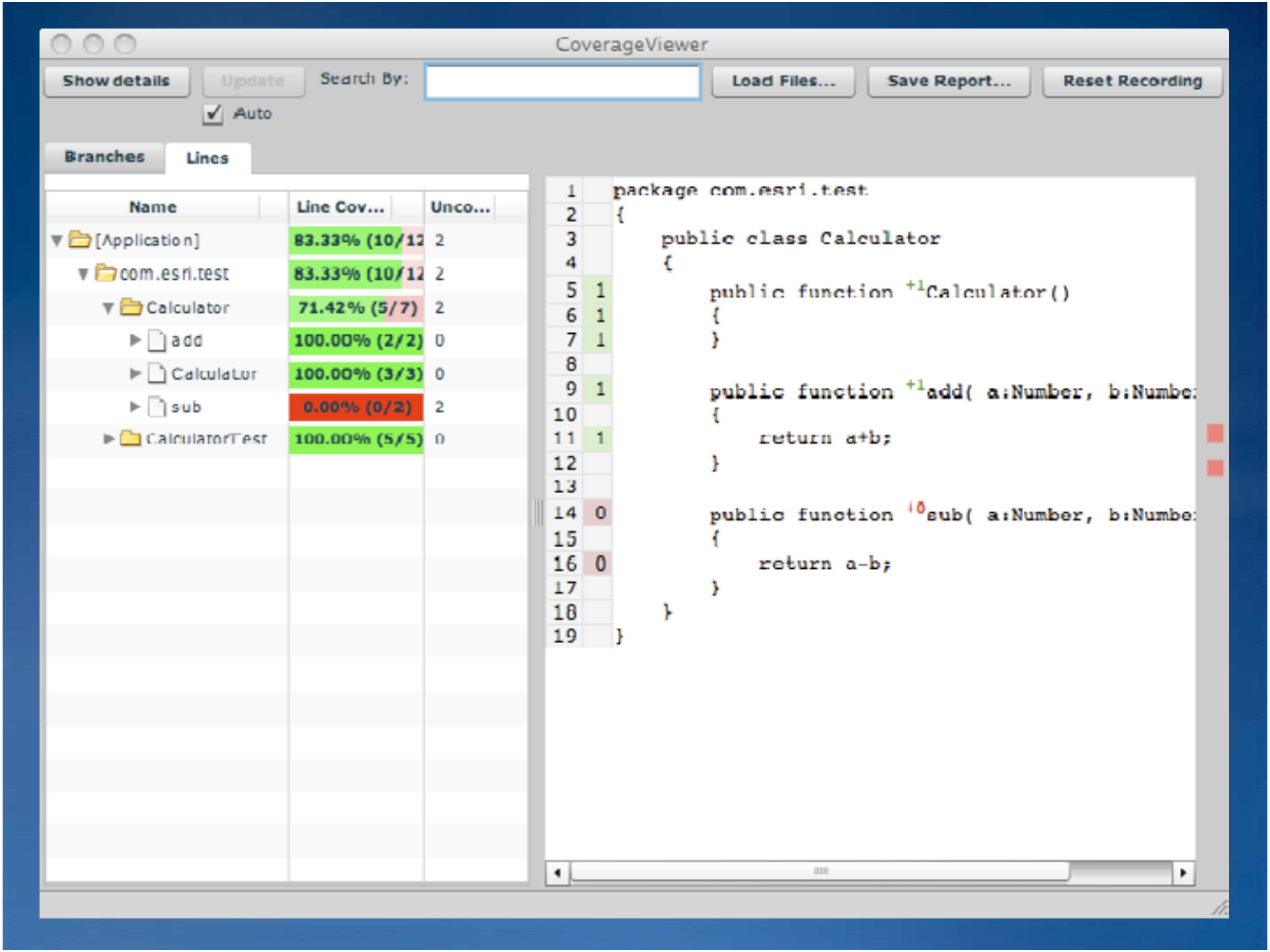
```
1   <?xml version="1.0" encoding="utf-8"?>
2   <mx:Application
3       xmlns:mx="http://www.adobe.com/2006/mxml"
4       xmlns:fluint="http://www.digitalprimates.net/2008/fluint"
5       layout="absolute"
6       creationComplete="creationCompleteHandler(event)"
7       >
8       <mx:Script>
9           <![CDATA[
10              import com.esri.test.MercatorTest;
11              import net.digitalprimates.fluint.tests.TestSuite;
12              protected function creationCompleteHandler( event:Event ) : void
13              {
14                  const testSuite : TestSuite = new TestSuite();
15                  testSuite.addTestCase( new MercatorTest());
16                  testRunner.startTests([testSuite]);
17              }
18
19          ]]>
20      </mx:Script>
21      <fluint:TestResultDisplay width="100%" height="100%" />
22      <fluint:TestRunner id="testRunner"/>
23  </mx:Application>
```

UNIT TEST IN ACTION

Code Coverage

- http://code.google.com/p/flexcover/
- Compile code with modified SDK
- Instruments code
- Results are view in AIR application
- Excellent companion to Unit Testing


Properties for MVCApp
type filter text
Resource
Builders
Flex Applications
Flex Build Path
Flex Compiler
Flex Modules
Flex Server
Project References
Run/Debug Settings
Flex Compiler
Flex SDK version
Use default SDK (currently "Flex 3.3")
Configure Flex SDKs...
Use a specific SDK:
Flexcover 3.2.0.3794
Compiler options
Copy non-embedded files to output folder
Generate accessible SWF file
Enable strict type checking
Enable warnings
Additional compiler arguments:
-locale en_US
HTML wrapper
Generate HTML wrapper file
Require Flash Player version:
9
0
24
Use Express Install
Enable integration with browser navigation
Restore Defaults
Apply
Cancel
OK


CoverageViewer
Show details
Update
Search By:
Load Files...
Save Report...
Reset Recording
Auto
Branches
Lines
Name
Line Cov...
Unco...
[Application] 83.33% (10/12 2
com.esri.test 83.33% (10/12 2
Calculator 71.42% (5/7) 2
add 100.00% (2/2) 0
CalculaLor 100.00% (3/3) 0
sub 0.00% (0/2) 2
CalculatorTest 100.00% (5/5) 0
1 package com.esri.test
2 {
3 public class Calculator
4 {
5 1 public function +1Calculator()
6 1 {
7 1 }
8
9 1 public function +1add(a:Number, b:Numbe
10 {
11 1 return a+b;
12 }
13
14 0 public function +0sub(a:Number, b:Numbe
15 {
16 0 return a-b;
17 }
18 }
19 }

COVERAGE IN ACTION

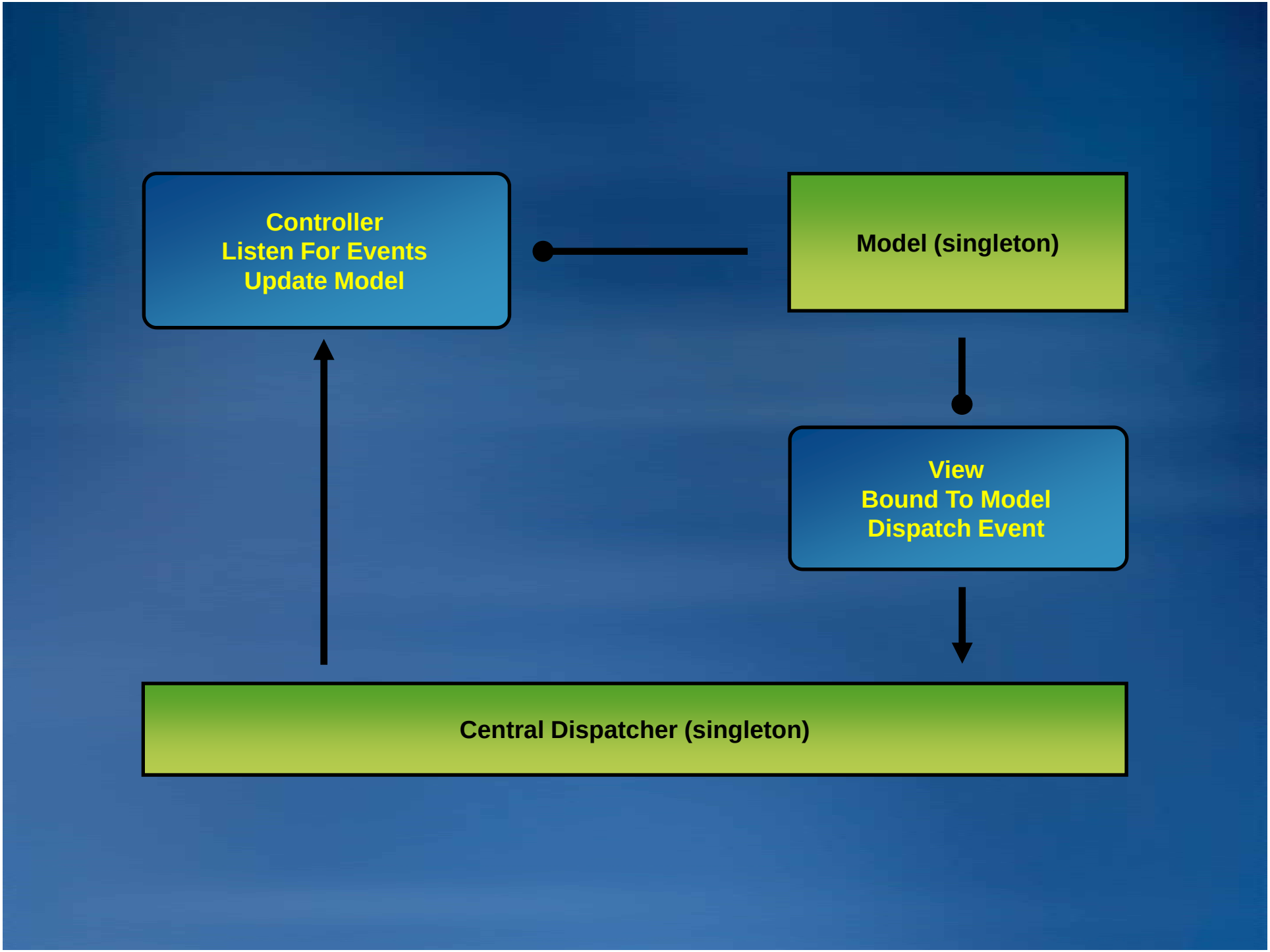
BUILDING RIA’S

Model View Controller (MVC) Frameworks

- Cairngorm
- PureMVC
- Mate
- Swiz
- Spring
- Own ?

Design Pattern Usage in MVC

- Singleton Model
- Central Dispatcher - Mediator
- Dynamic Event
- Controller as MXML
- Delegate vs Inheritance
- Unit testing


Controller
Listen For Events
Update Model
Model (singleton)
View
Bound To Model
Dispatch Event
Central Dispatcher (singleton)

Central Dispatcher

- Singleton
- Mediates events between _all_ compoments
- Based on EventDispatcher

```
1   package com.esri.events {
2       import flash.events.EventDispatcher;

3       public class CentralDispatcher extends EventDispatcher {
4           public function CentralDispatcher(singletonEnforcer : SingletonEnforcer) {
5               super(null);
6           }
7
8           private static var m_instance : CentralDispatcher;
9
10          public static function get instance() : CentralDispatcher {
11              if(m_instance == null ) {
12                  m_instance = new CentralDispatcher(new SingletonEnforcer());
13              }
14              return m_instance;
15          }
16
17      }
18  }

19  class SingletonEnforcer {
20  }
```

Model

- Singleton
- Holds the state of the application
- Bindable properties

```
1   package com.esri.model {
2       import mx.collections.ArrayCollection;
3
4       public class Model
5       {
6           [Bindable]
7           public var arrcol : ArrayCollection = new ArrayCollection();
8
9           public function Model( singletonEnforcer : SingletonEnforcer){
10          }
11
12          private static var m_instance : Model;
13
14          public static function get instance() : Model {
15              if( m_instance == null ) {
16                  m_instance = new Model( new SingletonEnforcer());
17              }
18              return m_instance;
19          }

20      }
21  }

22  class SingletonEnforcer {
23  }
```

View - MyDataGrid

- Subclass Flex DataGrid
- Visualize model changes

```
1   package com.esri.views
2   {
3       import com.esri.model.Model;
4
5       import mx.controls.DataGrid;
```

```
6       public class MyDataGrid extends DataGrid
7       {
8           public function MyDataGrid()
9           {
10              this.dataProvider = Model.instance.arrcol;
11          }
12
13      }
14  }
```

View - MyControlBar

- Subclass Flex ControlBar
- Two text input fields
- One button with ‘click’ handler
- Dispatches an event
- Event encapsulates all info

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <mx:ControlBar xmlns:mx="http://www.adobe.com/2006/mxml">
3      <mx:Script>
4          <![CDATA[
5              import com.esri.events.FooEvent;
6          ]]>
7      </mx:Script>
8      <mx:TextInput id="col1"/>
9      <mx:TextInput id="col2"/>
10     <mx:Button id="addBtn" label="Add">
11         <mx:click>
12             <![CDATA[
13                 const fooEvent : FooEvent = new FooEvent();
14                 fooEvent.col1 = col1.text;
15                 fooEvent.col2 = col2.text;
16                 fooEvent.dispatch();
17             ]]>
18         </mx:click>
19     </mx:Button>
20 </mx:ControlBar>
```

FooEvent

- Subclass Flash Event
- Type of event is “foo”
- Overrides clone function
- Has convenience dispatch function

```
1   package com.esri.events {
2       import flash.events.Event;

3       public class FooEvent extends Event {
4           public static const TYPE : String = "foo";
5
6           public var col1 : String;
7           public var col2 : String;
8
9           public function FooEvent() {
10              super(TYPE);
11          }
12
13          override public function clone() : Event {
14              const fooEvent : FooEvent = new FooEvent();
15              fooEvent.col1 = col1;
16              fooEvent.col2 = col2;
17              return fooEvent;
18          }
19
20          public function dispatch() : void {
21              CentralDispatcher.instance.dispatchEvent(this);
22          }
23      }
24  }
```

Controller - FooCommand

- Named “Foo” by convention not configuration
- Listens for events of type “foo”
- Does some logic
  - Connect to remote services
  - Invokes delegates
    - Better to delegate than to inherit
- Modifies the model

```
1   package com.esri.command
2   {
3       import com.esri.delegate.IItemFactory;
4       import com.esri.delegate.ItemFactory;
5       import com.esri.events.CentralDispatcher;
6       import com.esri.events.FooEvent;
7       import com.esri.model.Model;
8
9       public class FooCommand {
10          [Bindable]
11          public var itemFactory : IItemFactory;
12
13          public function FooCommand() {
14              CentralDispatcher.instance.addEventListener( FooEvent.TYPE, fooHandler
    );
15          }
16
17          private function fooHandler( event : FooEvent ) : void {
18              doFoo(event.col1, event.col2);
19          }
20
21          private function doFoo( val1 : String, val2 : String ) : void {
22              Model.instance.arrcol.addItem( itemFactory.createItem(val1, val2));
23          }
24      }
25  }
```

PUTTING IT ALL TOGETHER

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <mx:Application
3      xmlns:mx="http://www.adobe.com/2006/mxml"
4      xmlns:view="com.esri.views.*"
5      xmlns:command="com.esri.command.*"
6      xmlns:delegate="com.esri.delegate.*"
7      layout="vertical"
8      >
9
10     <delegate:ItemFactory id="itemFactory"/>
11     <command:FooCommand itemFactory="{itemFactory}"/>
12
13     <view:MyDataGrid/>
14     <view:MyControlBar/>
15
16 </mx:Application>
```

UNIT TESTING UI

Simulate User Input

```
const seqRun : SequenceRunner = new SequenceRunner( this );
seqRun.addStep( new SequenceSetter( m_myControlBar.col1, {text:"hello"}));
seqRun.addStep( new SequenceSetter( m_myControlBar.col2, {text:"world"}));
seqRun.addStep( new SequenceEventDispatcher( m_myControlBar.addBtn,
new MouseEvent( MouseEvent.CLICK)));
seqRun.run();
```

Assert User Input Value In Model

```
1    private function collectionChangeHandler(
2       event:CollectionEvent, data:Object):void
3    {
4        if (event.kind == CollectionEventKind.ADD)
5        {
6            assertEquals(1, Model.instance.arrcol.length);
7            const obj:Object = Model.instance.arrcol.getItemAt(0);
8            assertEquals("hello", obj.col1);
9            assertEquals("world", obj.col2);
10       }
11   }
```

MVC IN ACTION

Modular Programming

- Reduce application footprint
- Faster initial download
- Functions used occasionally
- Download on demand
- Built in module support

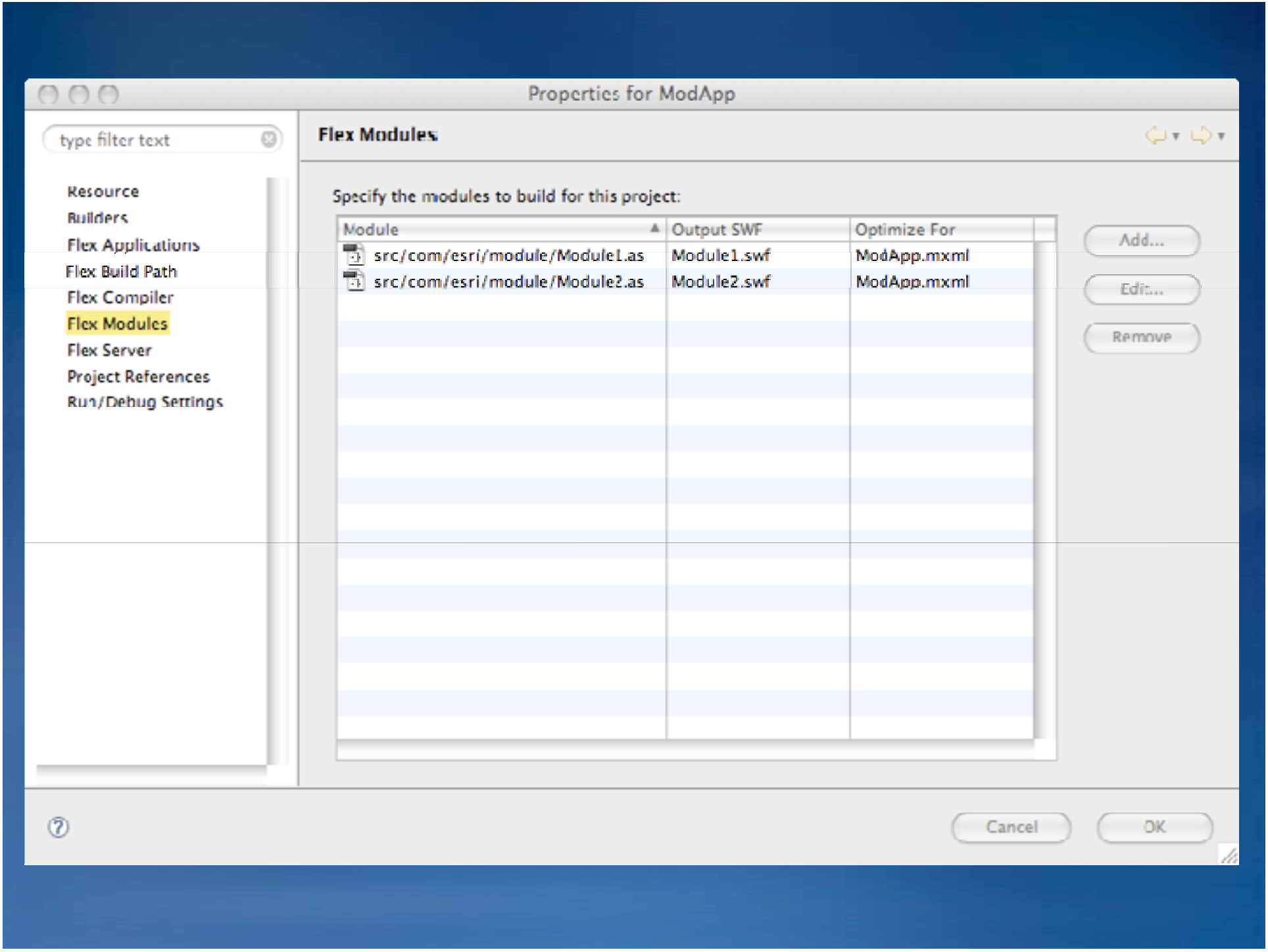
Modular Programming

- Define interface
- Extend ModuleBase
- Implement interface
- Use FlexBuilder to compile as module
- Load at runtime

Define Interface

```
1  package com.esri.bundle
2  {
3      import com.esri.ags.layers.GraphicsLayer;
4
5      public interface IGraphicProvider
6      {
7          function get graphicProvider() : Array;
8      }
9  }
```

```
1   package com.esri.module
2   {
3       import com.esri.ags.Graphic;
4       import com.esri.ags.geometry.MapPoint;
5       import com.esri.bundle.IGraphicProvider;
6
7       import mx.modules.ModuleBase;

8       public class Module1 extends ModuleBase implements IGraphicProvider
9       {
10          public function get graphicProvider() : Array
11          {
12              const arr : Array = [];
13              for( var i:int = 0; i < 100; i++ )
14              {
15                  var x : Number = -180.0 + 360.0 * Math.random();
16                  var y : Number = -90 + 180.0 * Math.random();
17                  arr.push( new Graphic( new MapPoint( x, y )));
18              }
19              return arr;
20          }
21      }
22  }
```


Properties for ModApp
type filter text
Resource
Builders
Flex Applications
Flex Build Path
Flex Compiler
Flex Modules
Flex Server
Project References
Run/Debug Settings
Flex Modules
Specify the modules to build for this project:
Module
Output SWF
Optimize For
src/com/esri/module/Module1.as
Module1.swf
ModApp.mxml
src/com/esri/module/Module2.as
Module2.swf
ModApp.mxml
Add...
Edit...
Remove
Cancel
OK

```
private var m_moduleInfo : IModuleInfo;

private function loadHandler( id : String) : void
{
    if( m_moduleInfo )
    {
        m_moduleInfo.unload();
    }
    m_moduleInfo = ModuleManager.getModule( "com/esri/module/Module"+id+".swf" );
    m_moduleInfo.addEventListener( ModuleEvent.READY, readyHandler );
    m_moduleInfo.load();
}

private function readyHandler( event : ModuleEvent ) : void
{
    const moduleLayer : IGraphicProvider =
      m_moduleInfo.factory.create() as IGraphicProvider;
    if( moduleLayer )
    {
        gl.graphicProvider = moduleLayer.graphicProvider;
    }
}
```

MODULES IN ACTION

BACK TO FLEX API

Best Practices Flex 4 AGS Usage

- Symbol Subclass
  - ArrowLineSymbol
- Geometry Subclass
  - Circle x,y,r
- Layer Subclass
  - HeatMap
- Raw Flash graphics

Create your own symbol

- Symbol draws a graphic
- Override clear function
- Override draw function
  - Call Flash draw primitive
  - Reference to attributes
  - Reference to map

```
1   package com.esri.sample {
2       import com.esri.ags.Map;
3       import com.esri.ags.geometry.Geometry;
4       import com.esri.ags.symbol.Symbol;

5       import flash.display.Sprite;

6       public class SimpleSymbol extends Symbol {

7           override public function clear(sprite:Sprite):void {
8               sprite.graphics.clear();
9           }

10          override public function draw(
11              sprite : Sprite,
12              geometry : Geometry,
13              attributes:  Object,
14              map : Map
15          ):void {
16              with(sprite.graphics) {
17                  // moveTp
18                  // lineTo
19              }
20          }
21      }
22  }
```

Modify Symbol Properties

- Initialize internal var
- Define get function
- Define set function
  - Check if different value
  - Assign new value
  - Dispatch change event

```
1       private var m_arrowWidth:Number = 5;

2       public function get arrowWidth():Number { return m_arrowWidth; }

3       public function set arrowWidth(value:Number):void {
4           if (value !== m_arrowWidth)
5           {
6               m_arrowWidth = value;
7               dispatchEventChange();
8           }
9       }

10      private var m_arrowLength:Number = 15;

11      public function get arrowLength():Number { return m_arrowLength;
    }

12      public function set arrowLength(value:Number):void {
13          if (value !== m_arrowLength)
14          {
15              m_arrowLength = value;
16              dispatchEventChange();
17          }
18      }
```

```
1    <esri:Map openHandCursorVisible="false">
2         <esri:GraphicsLayer>
3            <esri:Graphic>
4               <esri:symbol>
5                  <s:ArrowLineStyle color="0xFF0000" width="3”
6                     arrowLength="{sliderLength.value}”
7                     arrowWidth="{sliderWidth.value}"/>
8               </esri:symbol>
9               <esri:geometry>
10                ….
11              </esri:geometry>
12           </esri:Graphic>
13        </esri:GraphicsLayer>
14     </esri:Map>

15  <mx:HSlider id="sliderLength" minimum="5" maximum="30" liveDragging="true”/>
16  <mx:HSlider id="sliderWidth" minimum="5" maximum="30" liveDragging="true”/>
```

CUSTOM SYMBOL IN ACTION

Custom Geometry

- Subclass Geometry
- Define your own properties
  - Circle (centerLat, centerLon, radius)
- Define your own symbol
  - CircleSymbol

```
1   package com.esri.draw
2   {
3       import com.esri.ags.geometry.Geometry;
4       import com.esri.ags.geometry.MapPoint;

5       public class CircleGeometry extends Geometry
6       {

7           public var center : MapPoint;
8           public var radius : Number;
9
10      }
11  }
```

SYMBOL DRAWS GEOMETRY

```
1   package com.esri.draw {
2       import com.esri.ags.Map;
3       import com.esri.ags.geometry.Geometry;
4       import com.esri.ags.symbol.Symbol;
5
6       import flash.display.Sprite;

7       public class CircleSymbol extends Symbol {
8           override public function clear(sprite:Sprite):void {
9               sprite.graphics.clear();
10          }

11          override public function draw(
12            sprite:Sprite, geometry:Geometry, attributes:Object, map:Map):void
13          {
14              const circleGeometry : CircleGeometry = geometry as CircleGeometry;
15              sprite.graphics.beginFill( 0xFF0000, 0.5 );
16              const cx : Number = toScreenX( map, circleGeometry.center.x );
17              const cy : Number = toScreenY( map, circleGeometry.center.y );
18              sprite.graphics.drawCircle( cx, cy, circleGeometry.radius );
19              sprite.graphics.endFill();
20          }
21
22      }
23  }
```

CUSTOM GEOMETRY IN ACTION

Bindable GP Result

```
1  <esri:Query id=“query” returnGeometry="true”/>
2  <esri:QueryTask id=“queryTask” url=“url-of-gp-task”/>
3  <esri:Map>
4    <esri:GraphicsLayer graphicProvider=“{queryTask.lastResult.features}”/>
5  </esri:Map>
6  <mx:DataGrid dataProvider=“{queryTask.lastResult.attributes}”/>
7  <mx:Button label=“Execute” click=“queryTask.execute(query)”/>
```

“LOTS” OF GRAPHICS

INTERACTIVE GRAPHICS

BITBLT

Faster GP Result Rendering

- Custom Layer
  - Override function updateLayer
- Intercept GP last result features
- Flash bitmap rendering
- Ramp colors cells

```
1   m_bitmapData = new BitmapData(map.width, map.height, true, 0x000000);
2   m_bitmapData.lock();
3   try
4   {
5       for each(var graphic:Graphic in Model.instance.features)
6       {
7           var gridValue:Number = Number(graphic2.attributes.GRID_VALUE);
8           var v:Number = 100.0 * (gridCode - gridMin) / gridDel;
9
10          var mapPoint:MapPoint = MapPoint(graphic2.geometry);
11          if (!map.extent.contains(mapPoint))
12              continue;

13          var ratio:int = Math.round(255.0 * v / 100.0);
14          var rampColor:Number = toRampColor(colorArray, ratio);

15          m_rect.x = toScreenX(mapPoint.x) - size2;
16          m_rect.y = toScreenY(mapPoint.y) - size2;
17          m_bitmapData.fillRect(m_rect, m_alpha | rampColor);
18      }
19  }
20  finally
21  {
22      m_bitmapData.unlock();
23  }
```

```
1   m_matrix.tx = toScreenX(map.extent.xmin);
2   m_matrix.ty = toScreenY(map.extent.ymax);

3   graphics.clear();
4   graphics.beginBitmapFill(m_bitmapData, m_matrix, false, true);
5   graphics.drawRect(m_matrix.tx, m_matrix.ty, map.width, map.height);
6   graphics.endFill();
```

DENSITY MAP IN ACTION

Summary

- Today we covered
  - Tracing
  - Logging
  - Unit testing
  - Code coverage
  - MVC pattern
  - MVC implementation
  - Modules
  - Subclass Symbol
  - Subclass Geometry
  - Subclass Layer

Still have questions?

Summary

- Today we covered
  - User experience considerations
  - Implementation details

Want to Learn More?

ESRI Product and Adobe Resources

- **ArcGIS Resource Center**
  - **Samples, API Reference and Concepts**
    http://resources.esri.com/arcgisserver/apis/flex/
- **Adobe**
  - **Learn-Flex-in-a-week (videos)**
    http://www.adobe.com/devnet/flex/videotraining/
  - **Getting started with Flex 3 (pdf)**
    http://www.adobe.com/devnet/flex/pdfs/getting_started_with_Flex3.pdf
  - **Flex documentation**
    http://livedocs.adobe.com/flex/3

Want to Learn More?

ESRI Training and Education Resources

- **Instructor-Led Training**
  - **Introduction to ArcGIS Server**
    http://training.esri.com/gateway/index.cfm?fa=catalog.courseDetail&CourseID=50106292_9.X
- **Free Web Training Seminar**
  - **Building Rich Internet Applications with ArcGIS API for Flex**
    http://training.esri.com/acb2000/showdetl.cfm?DID=6&Product_ID=940

http://www.esri.com/training

Additional Resources

Questions, answers and information…

- ***Meet the Team***
  - ***Wednesday, 6:00 – 7:00 p.m., Oasis 2***
- ***Other sessions***
  - **Best Practices for Designing Effective Map Services**
    Tuesday, March 24, 2009, 4:30pm-5:45pm
    Wednesday, March 25, 2009, 4:30pm-5:45pm
  - **Advanced Map Caching Topics**
    Wednesday, March 25, 2009, 2:45pm-4:00pm
    Thursday, March 26, 2009, 1:30pm-2:45pm
  - **Building great web maps using ArcGIS**
    Thursday, March 26, 2009, 8:30am-9:45am

Please complete the session survey!