



Building ArcGIS Engine Applications with Visual-Java Beans

Divesh Goyal

Ranjit Iyer

Introductions ...

- **Who are we?**
 - ArcGIS Engine Java Dev team members.
- **Who are you?**
 - ArcGIS Desktop developers?
 - MapObjects Java developers?
 - Current ArcGIS Engine developers?
 - Target Platforms?
 - Preferred IDE?

Our assumptions

- **You are familiar with Java AWT or Swing**
- **You have some experience with ArcObjects**
- **You are excited to be here!**

Agenda

- **Introduction**
- **Creating an application with Visual JavaBeans**
- **Customizing your application**
- **Avoiding common pitfalls**
- **Optimizing Display**
- **Deploying your application**

Overview of ArcGIS Engine Java

- **Java SDK for developing cross platform desktop GIS Applications.**
- **Collection of object libraries called ‘arcobjects’ for mapping, visualization, data management and GIS analysis.**
- **Create stand alone GIS apps or Embed GIS functions in existing applications**

ArcGIS Engine Capabilities

- **2D & 3D Visualization**
 - with Visual-Java Beans
- **Data creation and analysis**
 - data formats supported by ArcGIS Desktop
- **Geoprocessing**
 - invoke tools and models
- **Map authoring**
 - create and edit .mxd, .lyr files
- **Advanced capabilities with Extensions**

ArcGIS Engine Extensions

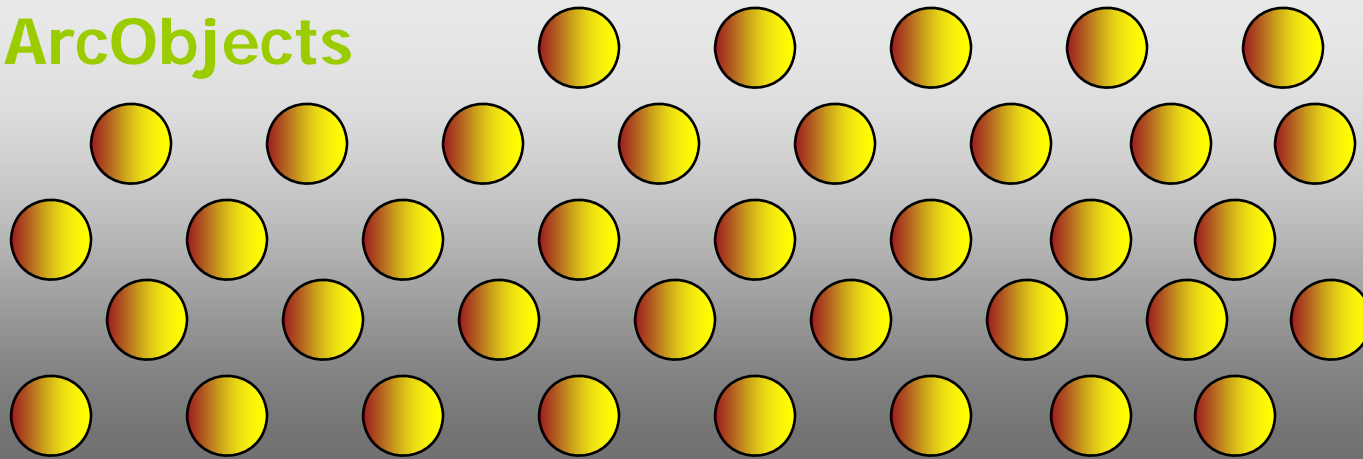
- **Spatial**
- **3D**
- **Geodatabase Update**
- **Network**
- **Data Interoperability**
- **Schematics**
- **Maplex**
- **Tracking**
- **StreetMap USA**

Architecture

Java Proxies & Visual JavaBeans

Interop Bridge

ArcObjects

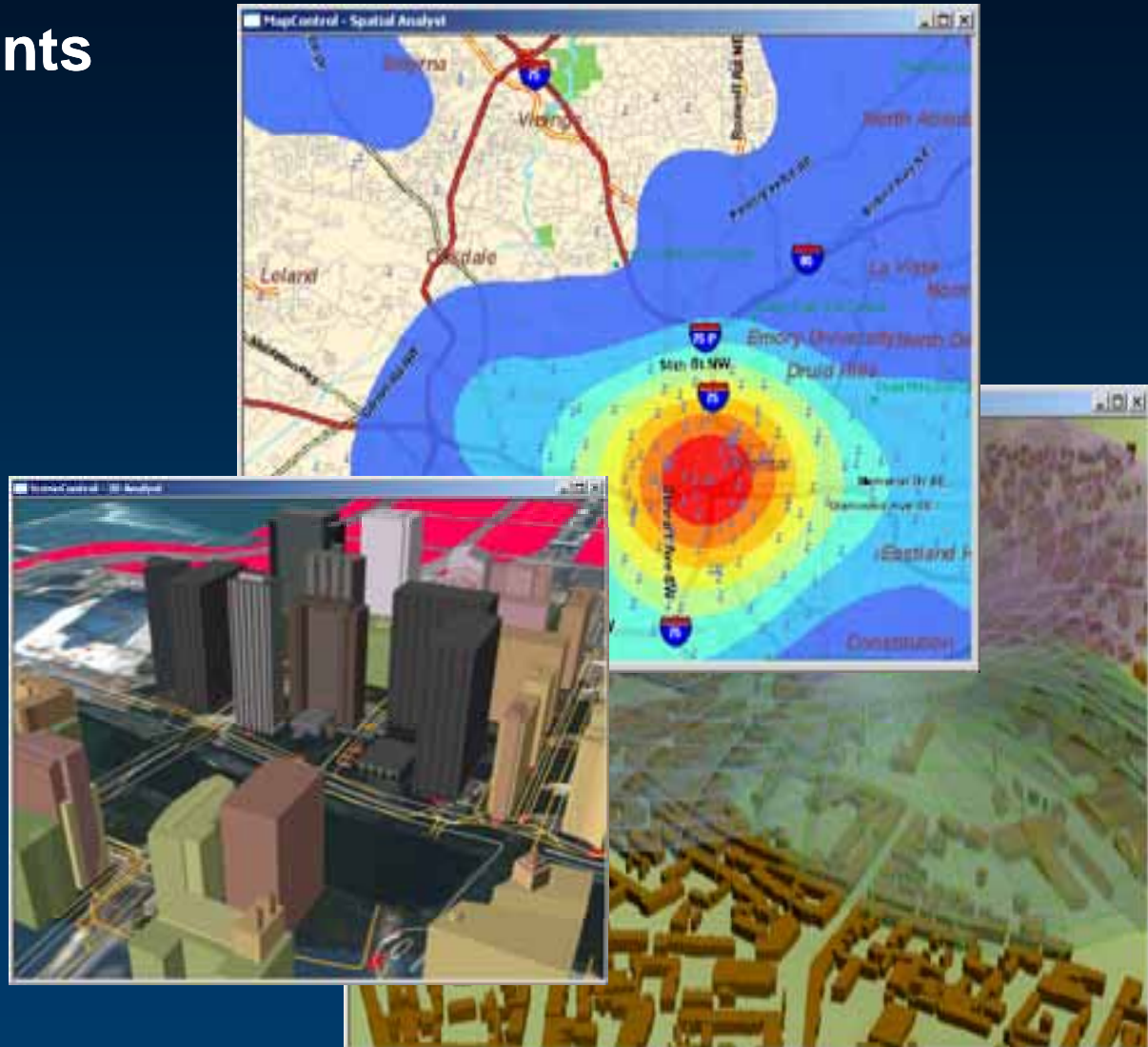


Windows / Solaris / Linux

What are Visual Java Beans?

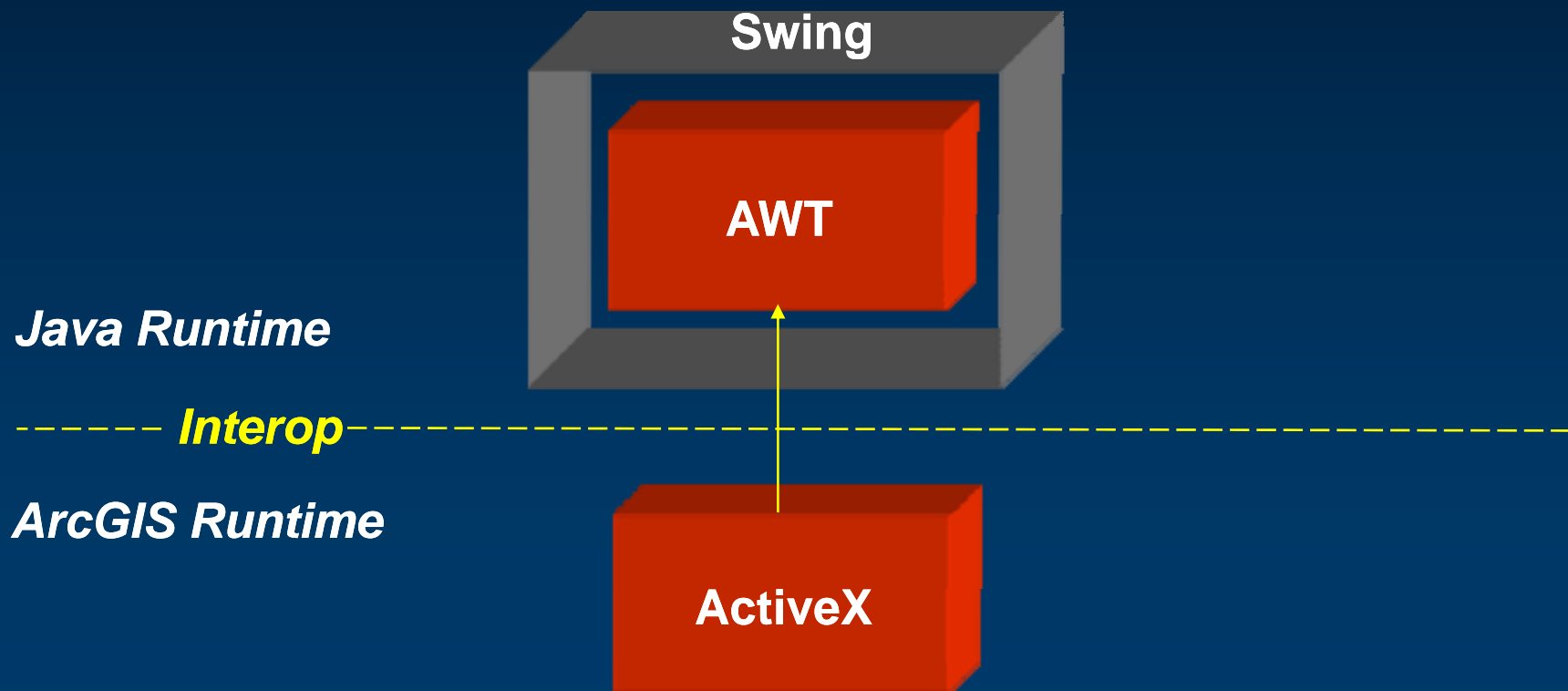
- **Visual components to build GUI**

- Map
- PageLayout
- Globe
- Scene
- Toolbar
- TOC
- Symbology



... but what are they really?

- ArcGIS ActiveX Controls
- Exposed as AWT controls by the Interop
- Wrapped in Swing for IDE design environment



Visual-JavaBean Event Listeners

- **Event Listeners**
 - correspond to Outbound interfaces
- **Respond to events**
 - mouse
 - keyboard
 - ArcGIS framework events
- **Use Adapters for convenience**

Agenda

- Introduction
- **Creating an application with Visual JavaBeans**
- Customizing your application
- Avoiding common pitfalls
- Optimizing Display
- Deploying your application

Application Lifecycle

1. Design UI

2. Initialize

- `EngineInitializer::initializeVisualBeans()`

3. License

- `AoInitialize::initialize()`
- `AoInitialize::checkOutExtension()`

4. Display UI

5. Shutdown

- `AoInitialize::checkInExtension()`
- `AoInitialize::shutDown()`

ArcGIS plugin for Eclipse

- **Design with Visual Java-beans**
 - Drag, Drop and Property Pages
- **Use code snippets for common tasks**
- **Explore sample projects**
- **Browse the integrated help system**

Agenda

- Introduction
- Creating an application with Visual JavaBeans
- **Customizing your application**
- Avoiding common pitfalls
- Optimizing Display
- Deploying your application

Commands

- Execute GIS function at the click of a button
- 150+ standard commands
- Create your own custom commands
 - implement ICommand
 - or, extend BaseCommand



Command : using BaseCommand

```
class MyCommand extends BaseCommand{
    HookHelper helper;
    public void onCreate(Object hook) {
        //initialize HookHelper
        helper = new HookHelper();
        helper.setHookByRef(hook);
        //set properties
        enabled = true;
        caption = "MyCommand";
        tooltip = "My Command";
        bitmapPath = "C:/myicons/image1.bmp";
    }
    public void onClick() {
        //use HookHelper
        IMap map = helper.getFocusMap();
        IActiveView view = helper.getActiveView();
        //do your thing
    }
}
```

IHookHelper

```
public void onCreate(Object hook) {  
    //getting ActiveView without HookHelper  
    if(hook instanceof MapControl){  
        ((MapControl)hook).getActiveView();  
    }else if(hook instanceof PageLayoutControl){
```

```
        public void onCreate(Object hook) {  
            //with HookHelper  
            helper = new HookHelper();  
            helper.setHookByRef(hook);  
            helper.getActiveView();  
            . . .  
        }  
    }  
}
```

```
    }  
    }  
    }  
    . . .  
}
```



cts

Tools

- Execute a GIS function after interacting with a map, globe, scene, or pagelayout.
- 60+ Standard Tools
- Create your own custom tools
 - implement ICommand & ITool
 - or, extend BaseTool

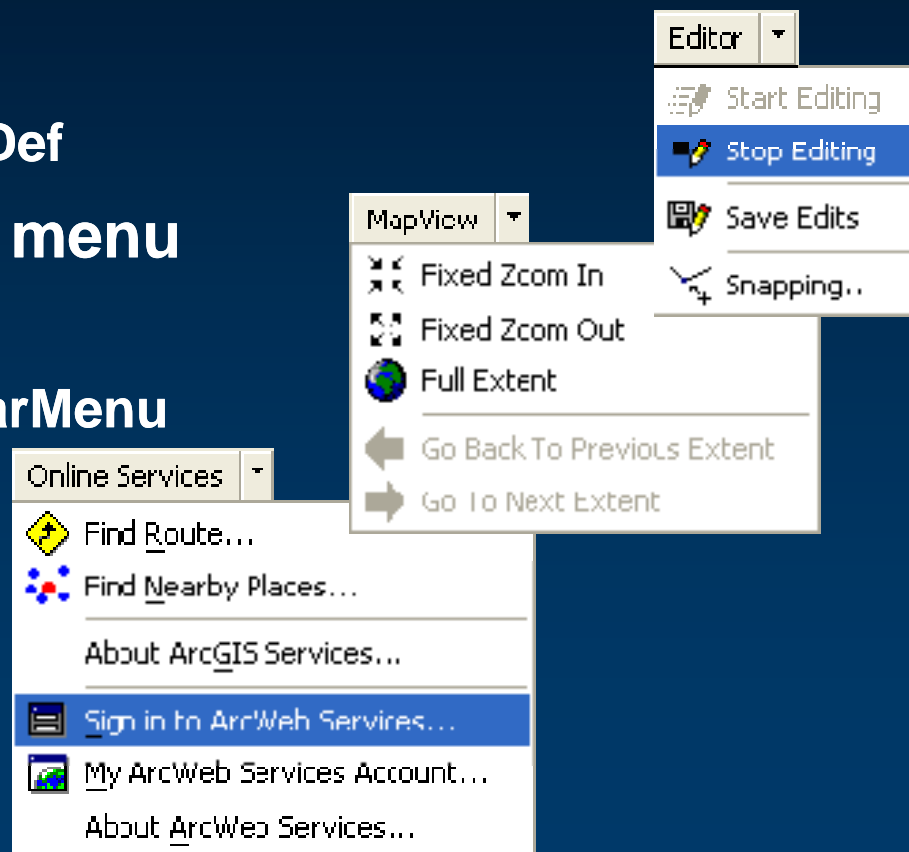


Tool : using BaseTool

```
class MyTool extends BaseTool{
    HookHelper helper;
    public void onCreate(Object hook) {
        //initialize HookHelper
        helper = new HookHelper();
        helper.setHookByRef(hook);
        //set properties
        enabled = true;
        caption = "MyCommand";
        toolTip = "My Command";
        bitmapPath = "C:/myicons/image1.bmp";
        cursorPath = "C:/myicons/foot.cur";
    }
    public void onMouseDown(int button, int shift, int x, int y) {
        //use HookHelper, do your thing
    }
}
```

Menus

- Group similar items together
- Save screen real-estate
- 8 Standard menus
 - all implement IMenuDef
- Create your custom menu
 - implement IMenuDef
 - or, instantiate ToolbarMenu




Custom Menu : using IMenuDef

```
class MyMenu implements IMenuDef{
    ...
    public int getItemCount(){
        return 2; //the number of menu items
    }
    public void getItemInfo(int pos, IItemDef def){
        //Add standard components to the menu
        switch(pos){
            case 0:
                def.setID(ControlsMapRotateTool.getClSID()); break;
            case 1:
                def.setID(ControlsMapBookmarkMenu.getClSID()); break;
        }
    }
    ...
    toolbarBean.addSubMenu(new MyMenu(),...); //Add to ToolbarBean
}
```

Custom Menu ▾

 Rotate Data Frame

Bookmarks ▸

 Create Bookmark...

Manage Bookmarks...

Custom Menu : using ToolbarMenu

```
//Instantiate ToolbarMenu
ToolbarMenu myToolbarMenu = new ToolbarMenu();
myToolbarMenu.setCaption("Custom Menu");

//Add standard components
myToolbarMenu.addItem(ControlsMapMeasureTool.getClsid(),...);

//Add custom components
ICommand myCommand = new MyCommand();
myToolbarMenu.addItem(myCommand,...);
IMenuDef mySubMenu = new MySubMenu();
myToolbarMenu.addSubMenu(mySubMenu,...);

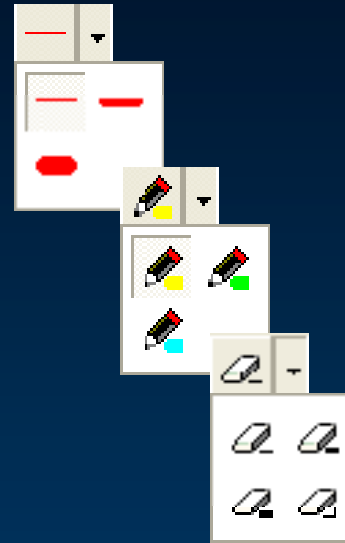
//Add to the toolbar
toolbarBean.addSubMenu(myToolbarMenu,...);
//or pop-up
myToolbarMenu.popUp(x,y,...);
```

Custom Menus

IMenuDef	ToolBarMenu
<ul style="list-style-type: none">• can contain standard components (CLSID based)	<ul style="list-style-type: none">• can contain standard components (CLSID based)• can contain custom components (Object Instance based)<ul style="list-style-type: none">– ICommand– IMultitem– IMenuDef– IPaletteDef– ToolBarMenu– ToolBarPalette
<ul style="list-style-type: none">• can appear on ToolBarControl	<ul style="list-style-type: none">• can appear on ToolBarControl• can appear as pop-up

Palettes

- **Similar to Menus**
- **3 standard Palettes**
 - all implement `IPaletteDef`
- **Create your custom palette**
 - implement `IPaletteDef`
 - or, instantiate `ToolbarPalette`



Custom Palette : using IPaletteDef

```
class MyPalette implements IPaletteDef{
    ...
    public int getItemCount(){
        return 3; //the number of menu items
    }
    public void getItemInfo(int pos, IItemDef def){
        //Add standard components to the menu
        switch(pos){
            case 0:
                def.setID(ControlsNewMarkerTool.getClsid()); break;
            case 1:
                def.setID(ControlsNewPolygonTool.getClsid()); break;
            case 2:
                def.setID(ControlsNewCircleTool.getClsid()); break;
        }
    }
    ...
    toolbarBean.addItem(new MyPalette(),...); //Add to ToolbarBean
}
```



Custom Palette : using ToolbarPalette

```
//Instantiate ToolbarPalette
ToolbarPalette myToolbarPalette = new ToolbarPalette ();
myToolbarPalette.setCaption("Custom Palette");

//Add standard components
myToolbarPalette.addItem(ControlsMapMeasureTool.getClsid(),...);

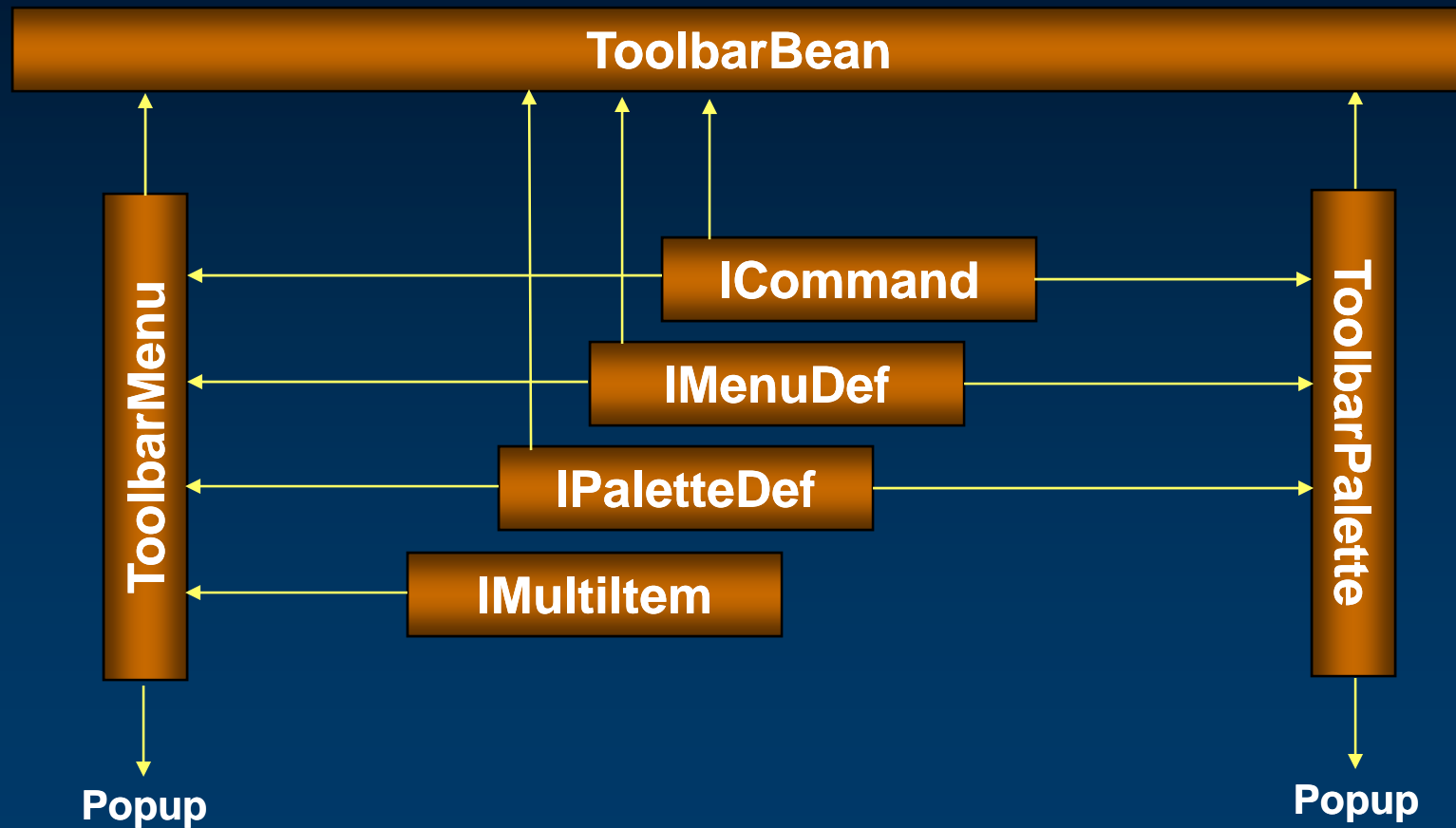
//Add custom components
ICommand myCommand = new MyCommand();
myToolbarPalette.addItem(myCommand,...);
IMenuDef mySubMenu = new MySubMenu();
myToolbarPalette.addSubMenu(mySubMenu,...);

//Add to the toolbar
toolbarBean.addItem(myToolbarPalette,...);
//or pop-up
myToolbarPalette.popUp(x,y,...);
```

Custom Palettes

IPaletteDef	ToolbarPalette
<ul style="list-style-type: none">• can contain standard components (CLSID based)	<ul style="list-style-type: none">• can contain standard components (CLSID based)• can contain custom components (Object Instance based)<ul style="list-style-type: none">–ICommand–IMenuDef–IPaletteDef–ToolbarPalette
<ul style="list-style-type: none">• can appear on ToolbarControl	<ul style="list-style-type: none">• can appear on ToolbarControl• can appear as pop-up

Putting it all together

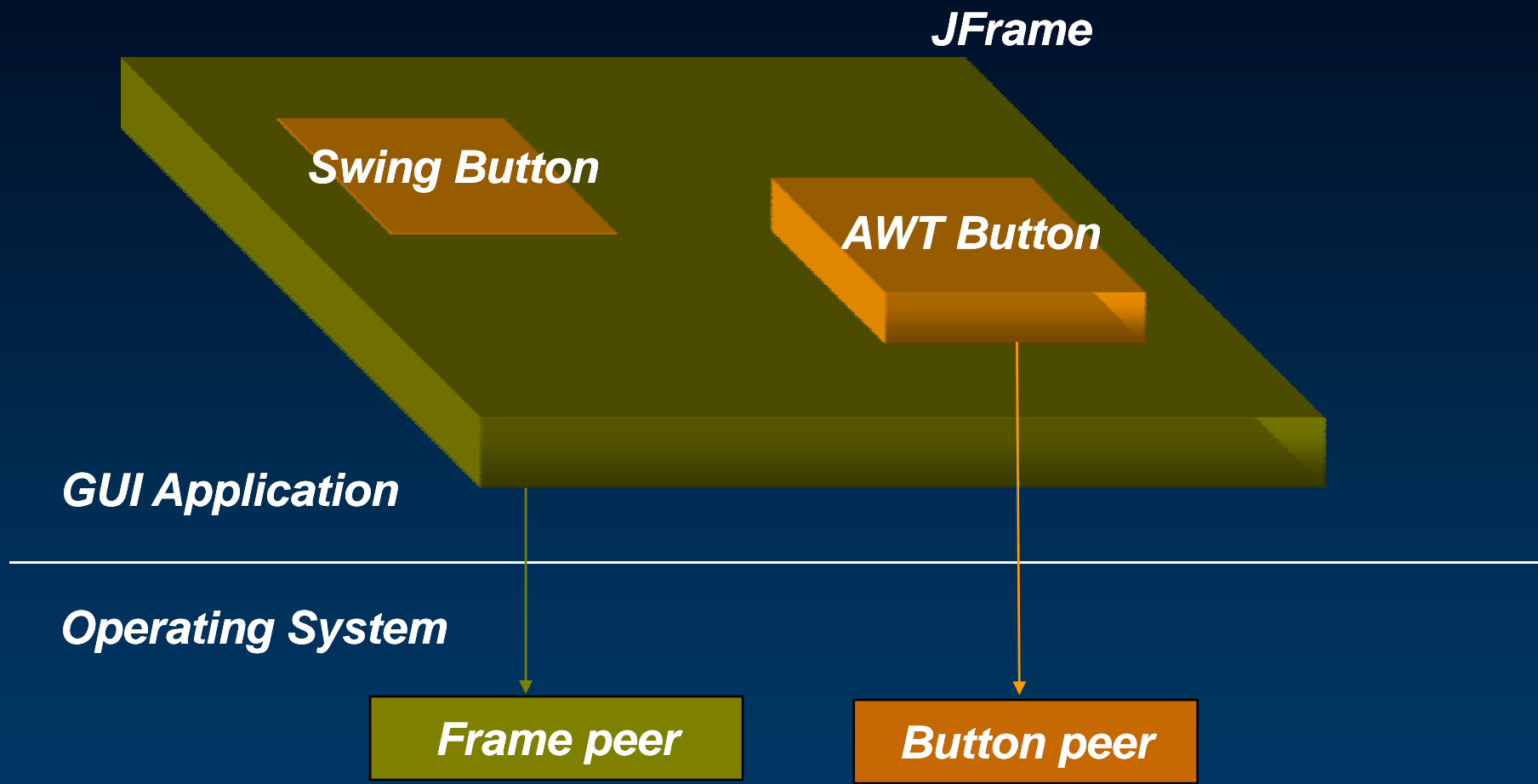


Agenda

- Introduction
- Creating an application with Visual JavaBeans
- Customizing your application
- **Avoiding common pitfalls**
- Optimizing Display
- Deploying your application

AWT and Swing under the hood

- **AWT components are *heavyweight***
 - They each get their own native screen resources (native peer)
 - A “Frame” in AWT is really a “Window” created using the Win32 API
- **Swing components are *lightweight* (except for the top level windows : JFrame, JDialog, JWindow, JApplet)**
 - They do not get their own native screen resources
 - They are simply drawn on their Parent’s canvas

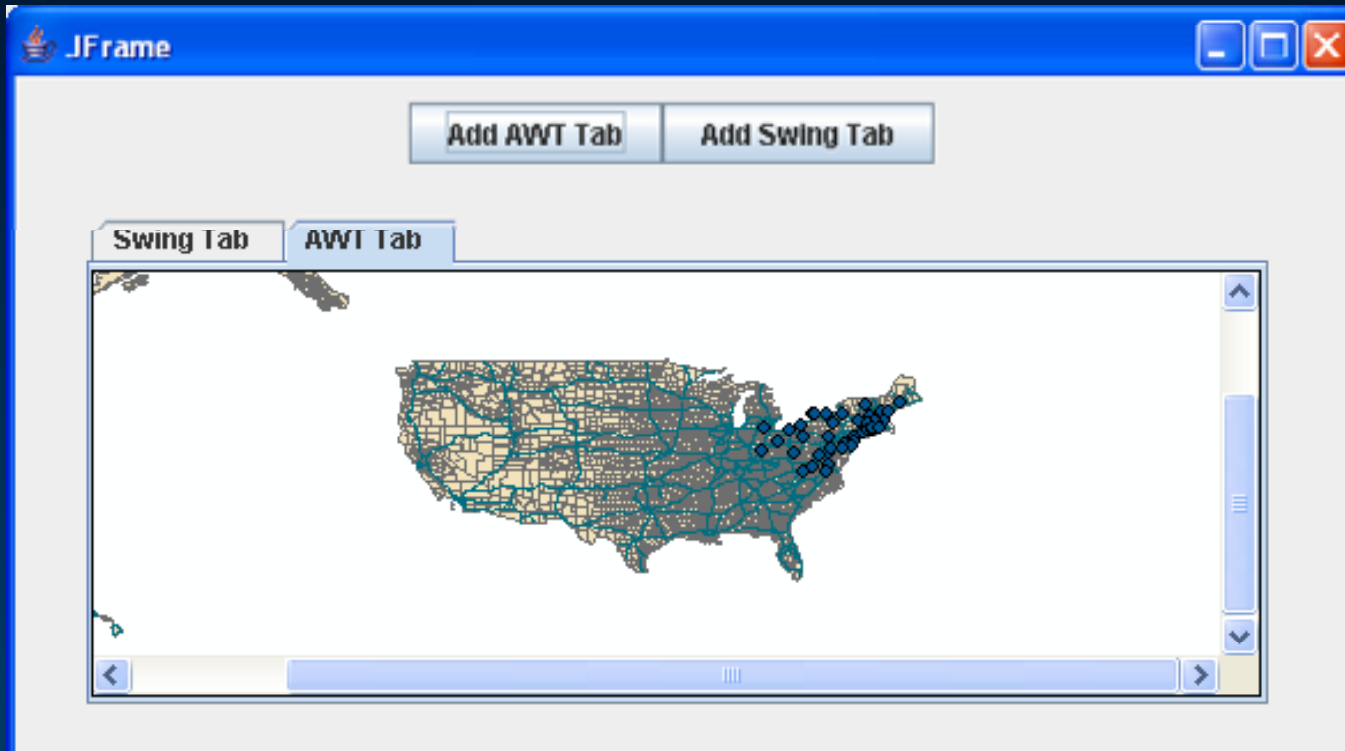


ESRI Engine Controls are AWT components

Mixing AWT and Swing

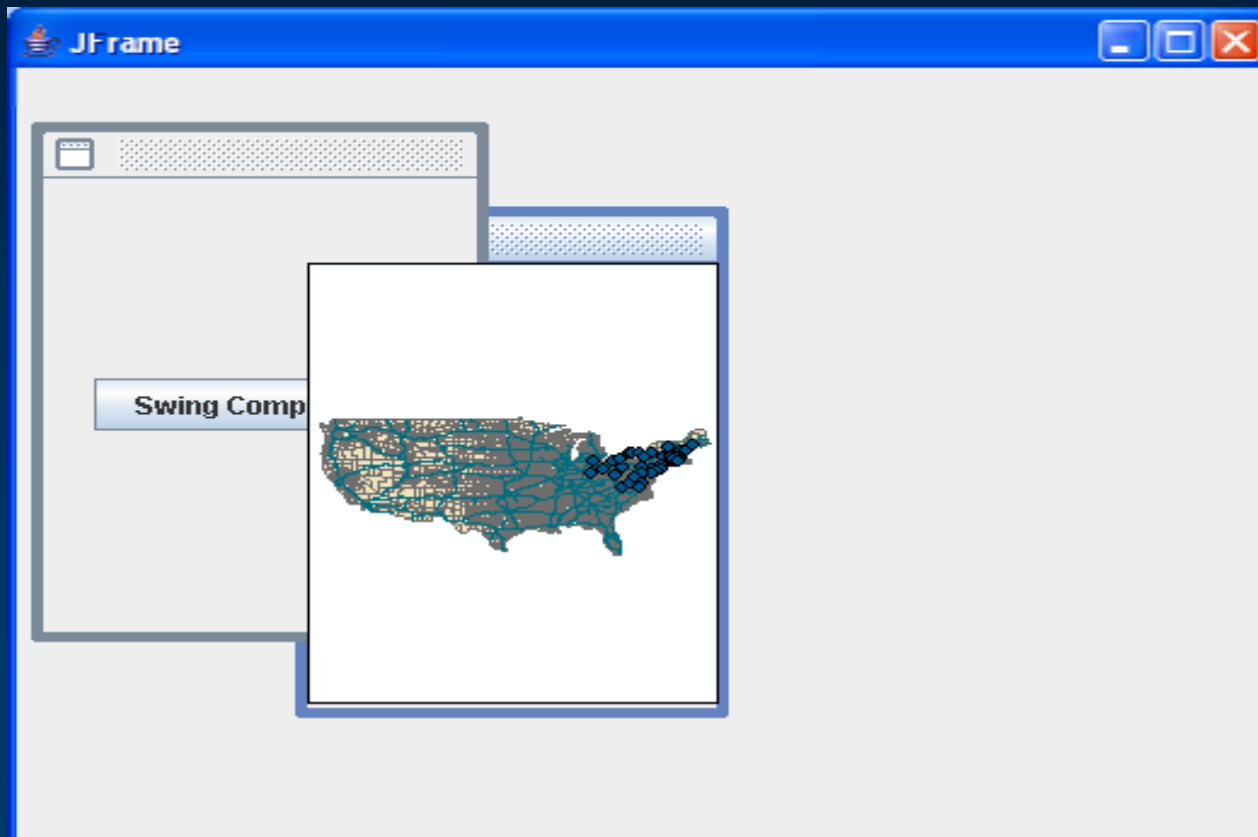
- **When would you mix AWT and Swing components**
 - You are in the process of migrating from AWT to Swing
 - You want to integrate ESRI Engine Controls in your applications
 - You want a native look and feel

A sample Swing/AWT application



Be careful though!

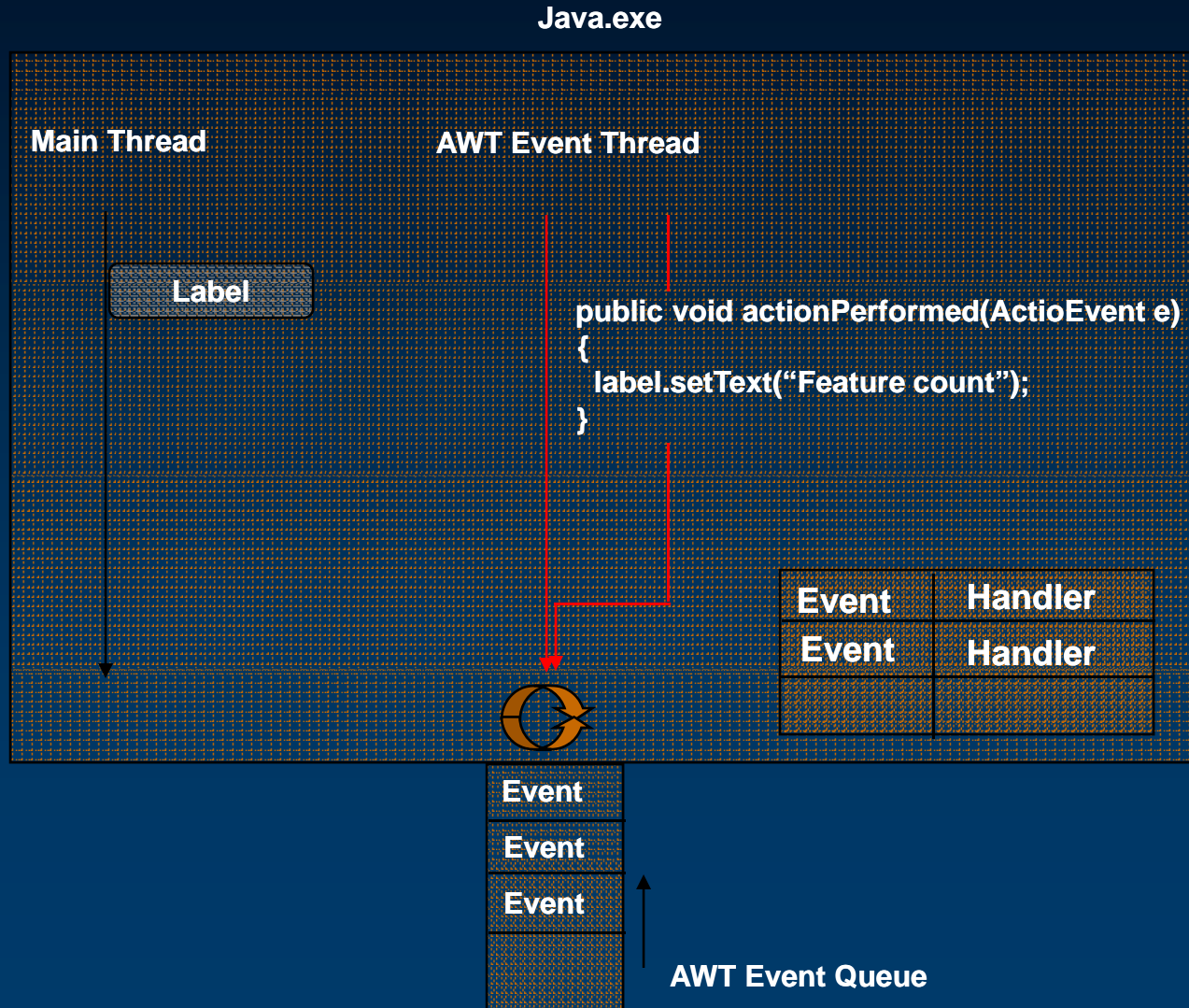
- Z-order issues cause the most grief
 - AWT controls can hide Swing components



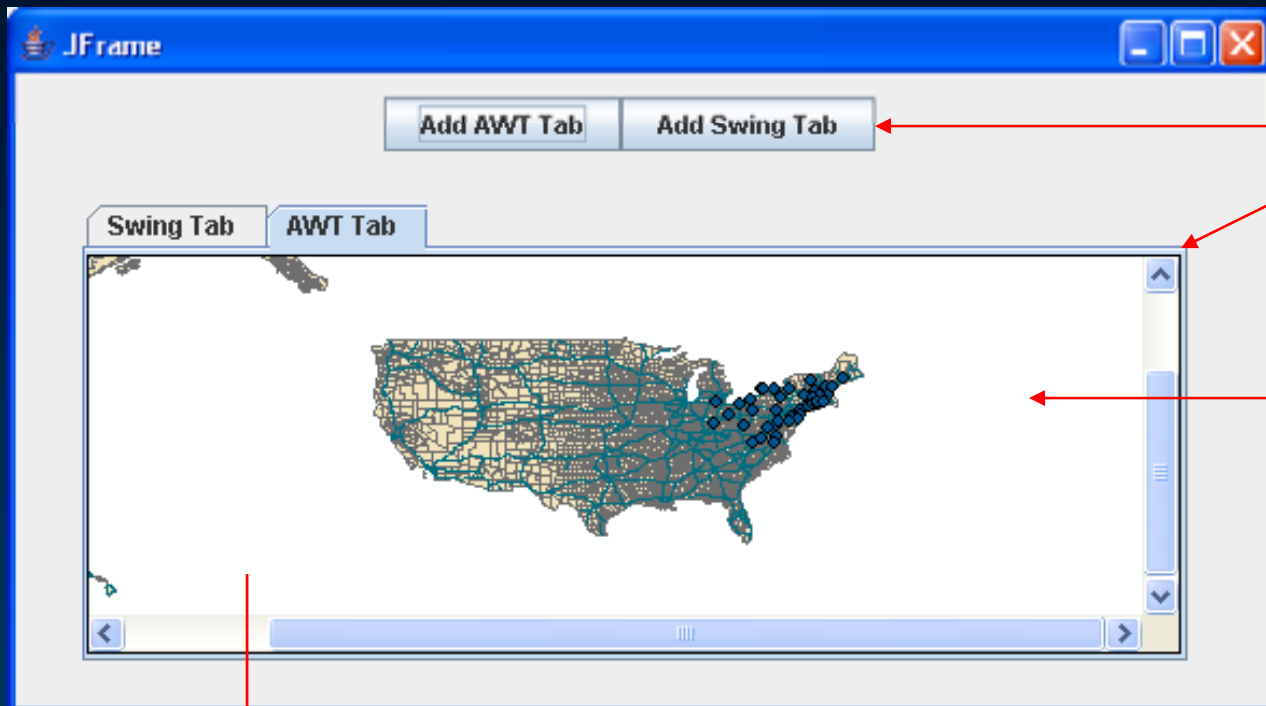
Potential Problems

- **What can be done?**
 - **Be careful while overlapping Swing components and Engine Controls**
 - **Exercise caution when embedding Engine Controls inside a JFrame**
 - **Avoid using Engine Controls inside of a JScrollPane**
 - **Use Heavy Weight Popups**
 - `JPopupMenu.setDefaultLightWeightPopupEnabled(false);`

Anatomy of a Java GUI Application



ArcObjects Event Listeners



JVM "owns" these components

ArcObjects "owns" MapControl

MapControl Events

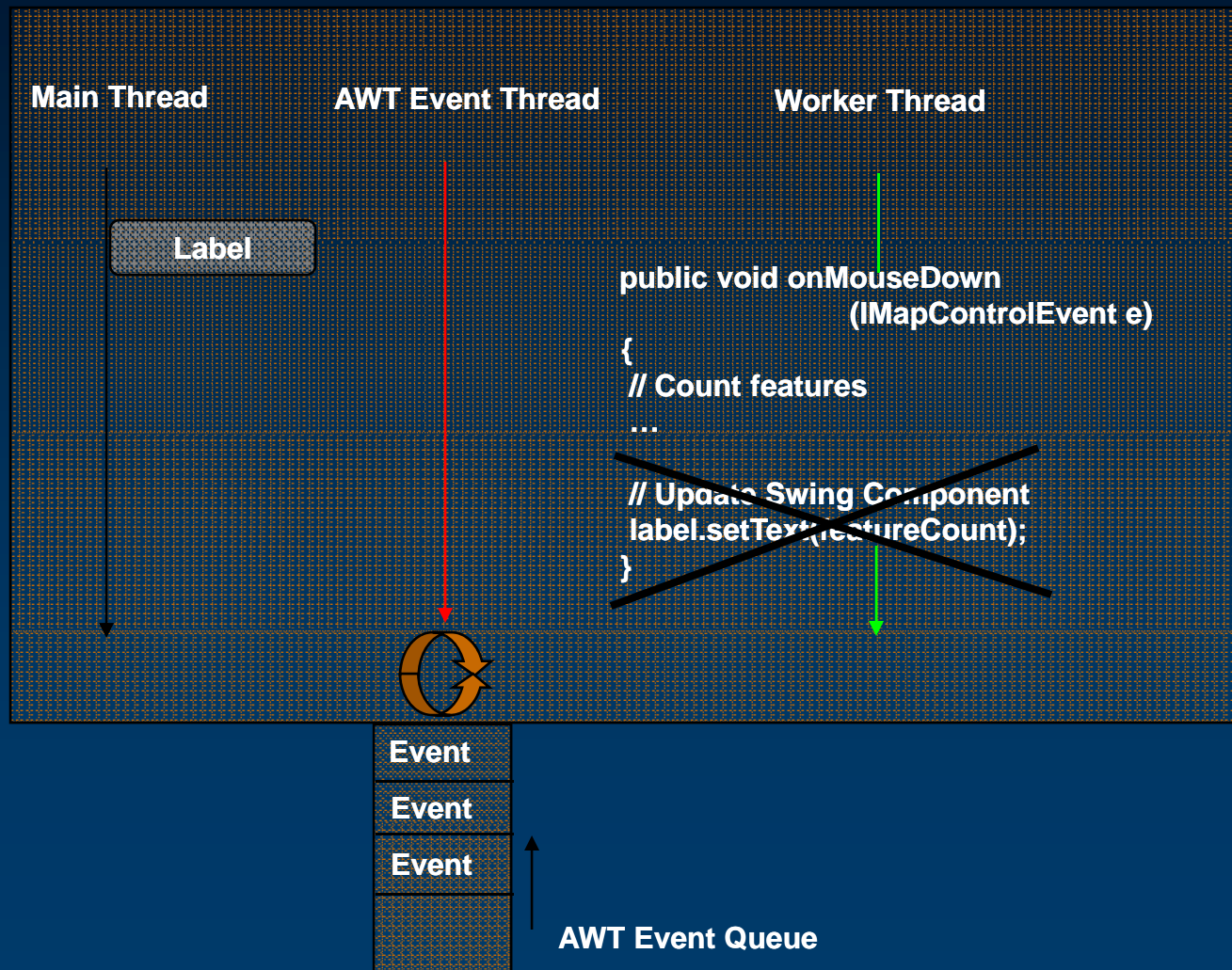
Call Event Handler

Java-COM Interop Bridge

ArcObjects

ArcObjects Event Listeners

Java.exe



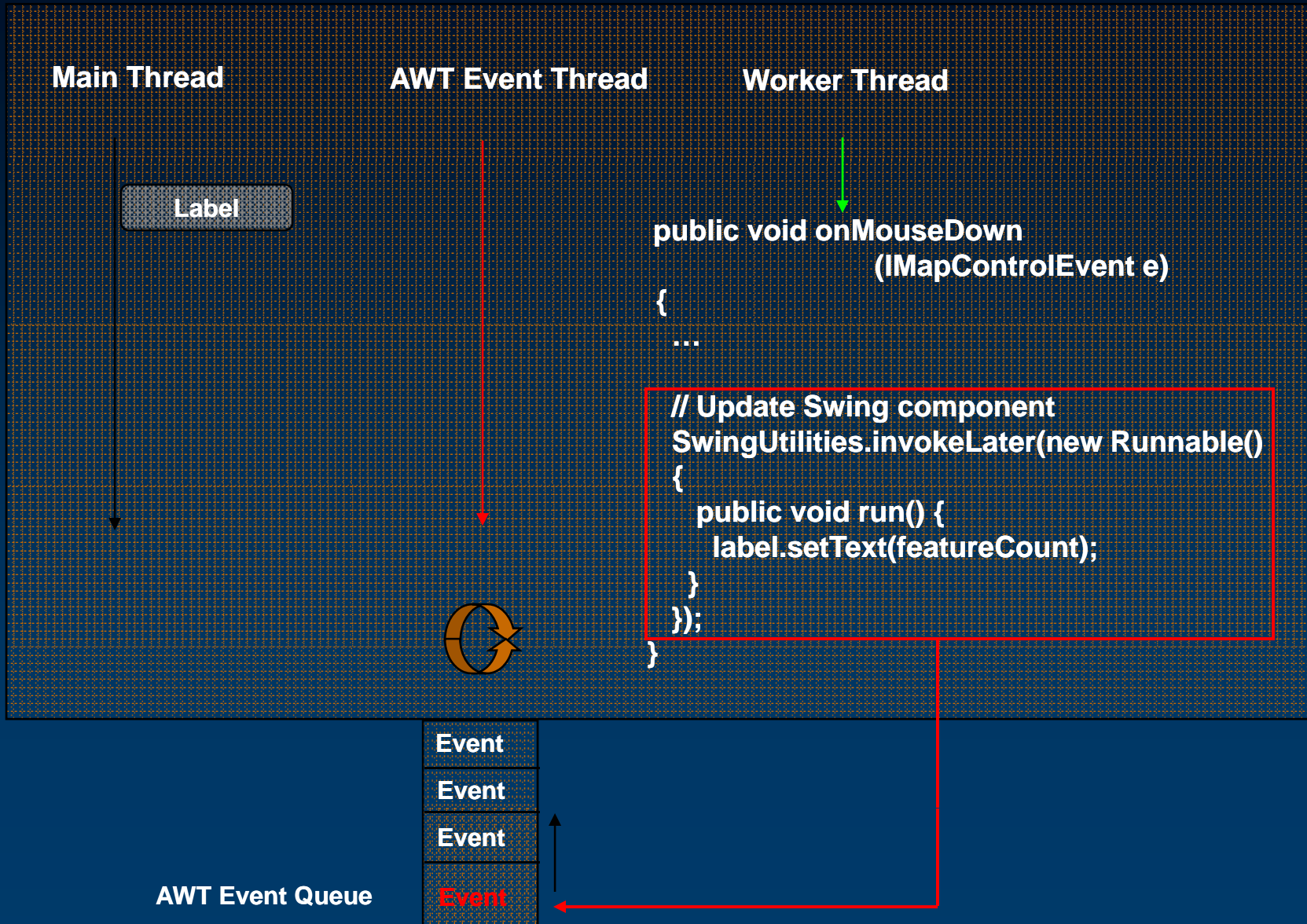
SwingUtilities to the rescue

- The `SwingUtilities` class provides two static methods
 - `invokeLater(Runnable task)`
 - `invokeAndWait(Runnable task)`

```
// MapControl event listener
public void onMouseDown(IMapControlEvents2OnMouseDownEvent event)
{
    // Perform some ArcObjects work
    ...

    // Safely update Swing/AWT components
    SwingUtilities.invokeLater(new Runnable()
    {
        public void run()
        {
            label.setText(featureCount);
        }
    });
}
```

Java.exe

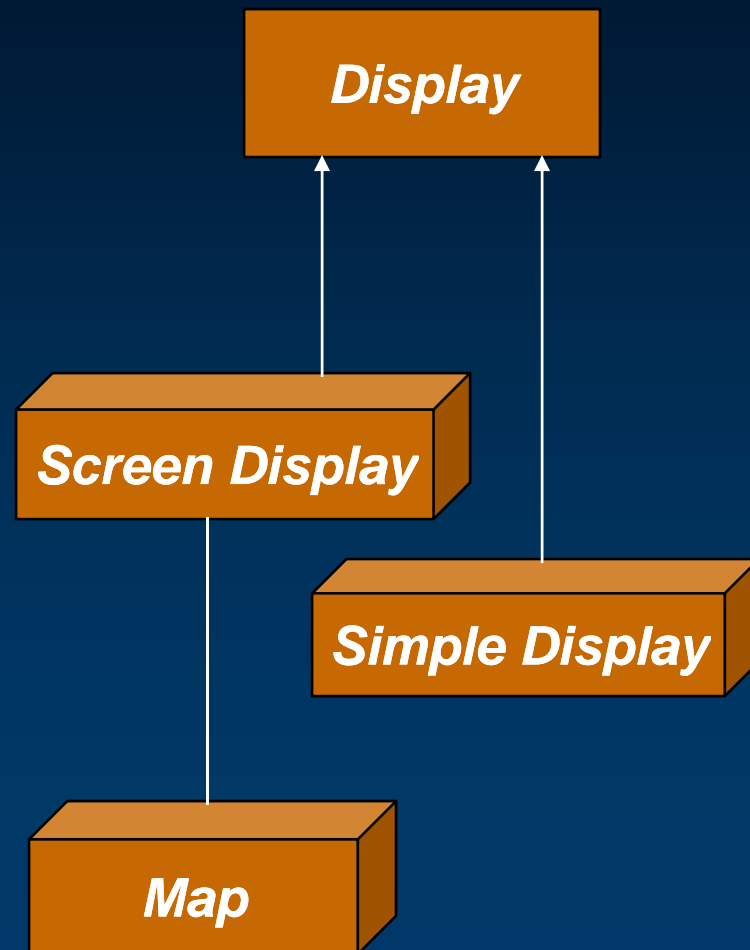


Agenda

- Introduction
- Creating an application with Visual JavaBeans
- Customizing your application
- Avoiding common pitfalls
- **Optimizing Display**
- Deploying your application

Display

- **Screen Display** abstracts an application window
- **Simple Display** abstracts other devices (printers, bitmaps, etc)
- The **ArcObject Map** uses the **Display** objects to render a map and manage display caches



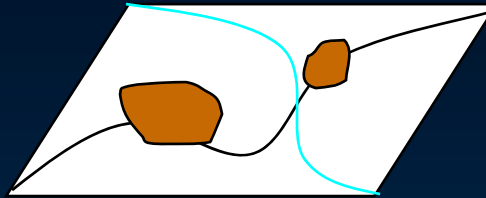
Display Caches

- **Map uses caching techniques to avoid drawing from disk**
- **Creates and maintains default caches**
 - Layers cache (features)
 - Graphics cache (labels, annotations, graphic elements)
 - Feature Selection cache
- **Allows creating private caches**

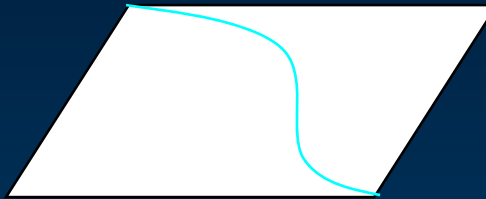
Display caches...

To Screen

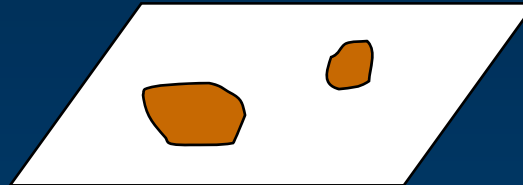
Recording Cache



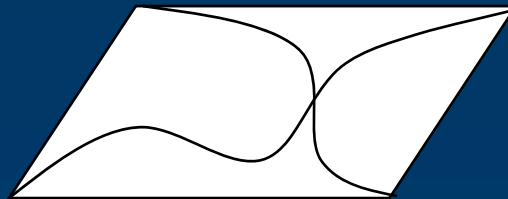
Feature Selection Cache



Graphics Cache



Layers Cache



Drawing Order

<i>Object</i>	<i>Phase</i>	<i>Cache</i>
<i>Graphic Selection</i>	<i>esriViewForeground</i>	<i>none</i>
<i>Feature Selection</i>	<i>esriViewGeoSelection</i>	<i>selection</i>
<i>Labels</i>	<i>esriViewGraphics</i>	<i>annotation</i>
<i>Graphics</i>	<i>esriViewGraphics</i>	<i>annotation</i>
<i>Layer Annotations</i>	<i>esriViewGraphics</i>	<i>annotation</i>
<i>Layers</i>	<i>esriViewGeography</i>	<i>layer</i>

Refreshing the Map

- **IActiveView::Refresh** is expensive
- **Use IActiveView::PartialRefresh** when possible
 - Lets you specify what part of the display to redraw

```
// Prototype of the partialRefresh method
```

```
IActiveView::partialRefresh(int drawPhase,  
    Object modifiedObject, IEnvelope refreshRegion);
```

```
pMap.partialRefresh(esriViewGraphics, 0, 0);
```

```
// Refresh a single Element
```

```
pMap.partialRefresh(esriViewGraphics, pElement, 0);
```

Adding a Graphic element

```
// Creating a graphic element
```

```
CircularElement element = new CircularElement();
```

```
SimpleFillSymbol symbol = new SimpleFillSymbol();
```

```
...
```

```
// Adding it to the graphics container
```

```
graphicsContainer.addElement(element, 0);
```

```
// Expensive
```

```
activeView.refresh();
```

```
// Recommended way to refresh.
```

```
activeView.partialRefresh(esriViewDrawPhase.esriViewG  
raphics, null, null);
```

Private cache

- **When do you need a private cache?**
 - Need to redraw features on a layer at regular intervals
 - Other layers in the Map don't change but are expensive to redraw
- **How do you create it?**
 - Set the cached property on the layer
 - Refresh the Map's caches

Creating and using a private cache

```
// Set the cache property on the layer
ILayer cachedLayer = ...;
cachedLayer.setCached(true);

// Refresh the Map's caches
Map map = (Map)mapControl.getMap();
map.refreshCaches();

...

// Refresh the layer. Other layers draw from cache
IActiveView activeView = mapControl.getActiveView();
activeView.partialRefresh(esriViewDrawPhase.esriViewGeography
    , cachedLayer, null);
```

Map Events

- The Map generates events when its state changes
- Clients implement *IActiveViewEvents* to register a listener and receive these events
- Two kinds of events
 - Reports a new state
 - `itemAdded()`, `contentsChanged()`
 - Allows participation during the Map's state change
 - `afterDraw()`, `afterItemDraw()`

Demo : Refresh Explorer

Map Events...

- **The `afterDraw()` method is called at the end of each draw phase**
- **Useful for drawing custom graphics while the Map refreshes**
- **Pick the draw phase and perform the drawing**
- **The drawing directly goes either to a cache or to display**

```
// Wiring up an event listener
map.addIActiveViewEventsListener(new IActiveViewEventsAdapter()
{
    public void afterDraw(IActiveViewEventsAfterDrawEvent evnt)
    {
        // The phase after which you will draw
        int drawPhase = evnt.getPhase();

        // The Screen Display
        IDisplay display = evnt.getDisplay();

        if(drawPhase == esriViewDrawPhase.esriViewGraphics)
        {
            // All drawings performed here go into the graphics
            // cache
        }
    }
});
```

Drawing to the screen

- Give instantaneous feedback without refreshing the Map
- No dependency on refresh cycles

```
// Start the drawing session. Draw to the screen directly
screenDisplay.startDrawing(0, esriNoScreenCache);

// Set the symbol for the geometry
screenDisplay.setSymbol(markerSymbol);

// Draw the geometry
screenDisplay.drawPoint(point);

// End the drawing session
screenDisplay.finishDrawing();
```

Agenda

- Introduction
- Creating an application with Visual JavaBeans
- Customizing your application
- Avoiding common pitfalls
- Optimizing Display
- **Deploying your application**

Java WebStart

- **One-Click deployment solution for Java Applications**
- **Webstart allows java applications to be deployed over a network**
- **Distributed like Applets, Perform like Applications**
- **Deployment Descriptor : JNLP – Java Network Launch Protocol**

Deploying your application using WebStart

- **Code your application**
- **Create and Sign your application's jar**
- **Create the .jnlp descriptor**
 - declare the main class
 - grant full permissions
- **Place application and descriptor on webserver**

Example .jnlp file

```
<jnlp spec="1.0+"  
  codebase="http://mytomcat:8080/devsummit/">  
  ...  
<security>  
  <all-permissions/>  
</security>  
<resources>  
  ...  
  <jar href="application.jar" main="true"/>  
  <jar href="utilities.jar"/>  
</resources>  
<application-desc main-class="com.myorg.Main" />  
</jnlp>
```

Presentation materials

- PowerPoint presentation and code are posted on the conference web site
 - <http://www.esri.com/events/devsummit/index.html>
- EDN – downloads and videos

Demo : Launching application using WebStart

Demo : Launching application using WebStart

Further questions?

- **TECH-TALK AREA 6**
 - in the Community Center (Oasis 3)
 - during the next 30 minutes
- **Meet the Java team**
 - Tuesday 4 PM, March 20 in Oasis 4
- **Java SIG**
 - Wednesday 12 PM, March 21 in Primrose C/D
- **ESRI Developers Network (EDN) website**
 - <http://edn.esri.com>

Thank you for coming!