



2008 ESRI Developer Summit

March 17–20, 2008 • Palm Springs, CA

Introduction to ArcGIS Server Development

Java Web ADF

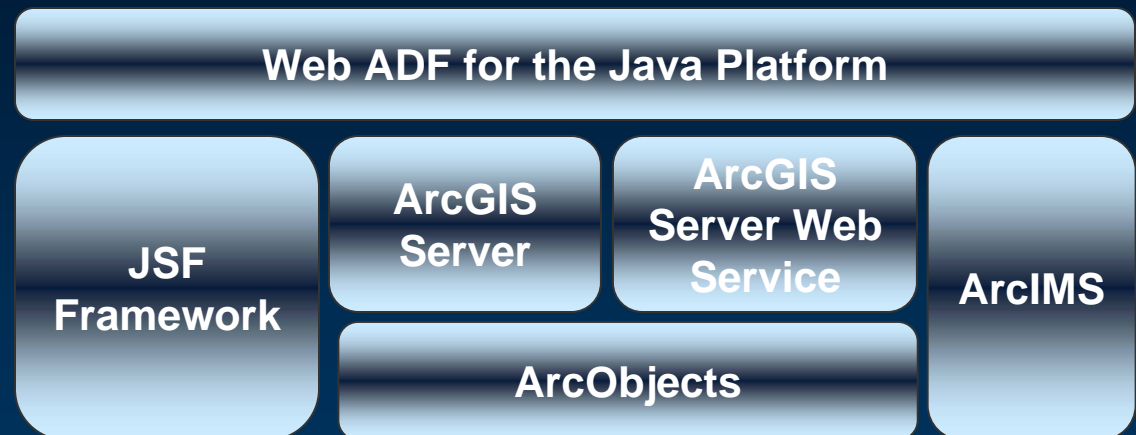
Sathya Prasad

Topics

- **Introduction to the Web ADF**
- **Web ADF controls**
- **Toolbars, commands and tools**
- **Tasks**
- **Introduction to developer APIs**
 - **Common API**
 - **Resource Specific APIs**
 - **Library Specific APIs**
- **Graphics**

Introducing the Web Application Developer Framework (ADF)

- Set of controls, tasks, and Web applications to enable creation of GIS applications
- Developer libraries
- Templates
- Documentation
- Samples
- Geospatial Enterprise Java Beans (EJB)
 - Enterprise ADF



Features of the Web ADF

- **ArcGIS Server Manager**
 - Build, host, and manage your Web applications
- **Custom JSF components**
 - Object-oriented, AJAX-enabled JavaScript library
- **Multisource web controls**
 - ArcGIS Server, ArcIMS, ArcWeb, WMS, and Custom
- **Task framework**
- **Seamless map navigation**
- **Client side Javascript libraries**
- **Integration with Eclipse and Creator IDEs**

Creating Web ADF applications

1. Use ArcGIS Server Manager

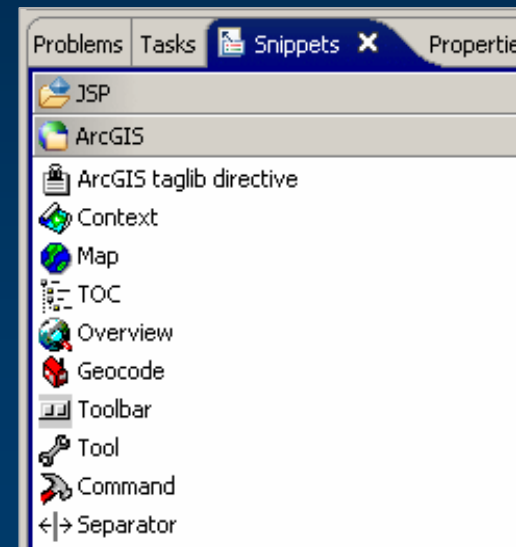
- Wizard to build and deploy
- No programming necessary
- Configure tasks
- Export to WAR file and modify



2. IDE integrated template

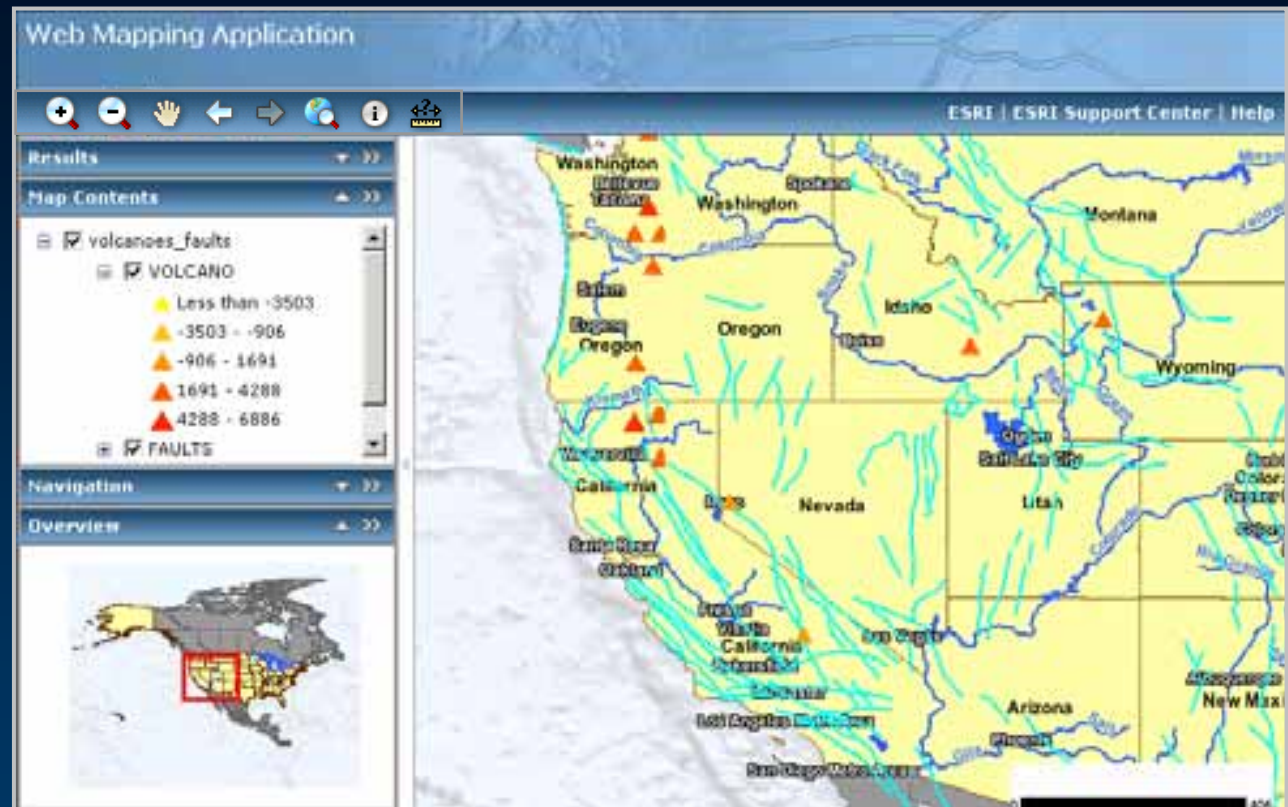
- Same template used by ArcGIS Server Manager

3. Create applications directly using Web controls



Web Mapping Application template

- Web pages
- Controls
- Images/icons
- Themes
- JavaScript libraries
- Custom tools
- Help system



Advantages of using the template

- Start with a complete GIS Web application
- The template provides:
 - Layout
 - Commonly used MapTools
 - zoom, pan, measure, identify, full extent
 - Themes
 - Customizable Help system
 - Code for docking items, closing application



Demo

- **Export web application from Manager**
- **Import the web application into Eclipse**
- **Run the web application**
- **Make changes to the web application**
- **Run the web application from Eclipse**

Topics

- Introduction to the Web ADF
- **Web ADF controls**
- Toolbars, commands and tools
- Tasks
- Introduction to developer APIs
 - Common API
 - Resource Specific APIs
 - Library Specific APIs
- **Graphics**

Web ADF controls

- **Set of custom JSF Web controls**
- **Expose server object functionality**

Web ADF controls

- Set of custom JSF Web controls
- Expose server object functionality
- **Types of controls**

Map/Overview

- Displays data



Web ADF controls

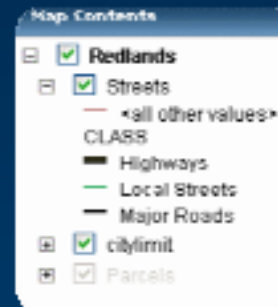
- Set of custom JSF Web controls
- Expose server object functionality
- **Types of controls**

Map/Overview

- Displays data

Table of contents (TOC)

- Map contents
- Serves as a legend



Web ADF controls

- Set of custom JSF Web controls
- Expose server object functionality
- **Types of controls**

Map/Overview

- Displays data

Table of contents (TOC)

- Map contents
- Serves as a legend

Context

- Non-visual control
- Manages all other controls

Web ADF controls

- Set of custom JSF Web controls
- Expose server object functionality
- **Types of controls**

Map/Overview

- Displays data

Context

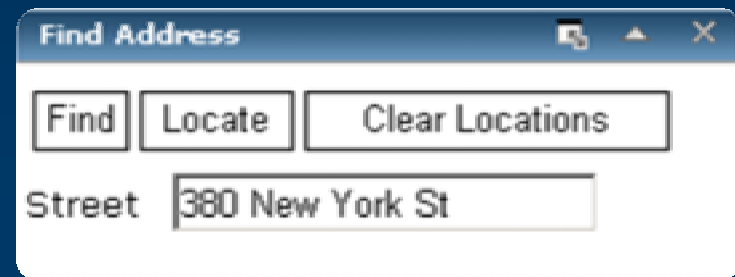
- Non-visual control
- Manages all other controls

Table of contents (TOC)

- Map contents
- Serves as a legend

Task

- Container for functionality



Web ADF controls

- Set of custom JSF Web controls
- Expose server object functionality
- **Types of controls**

Map/Overview

- Displays data

Table of contents (TOC)

- Map contents
- Serves as a legend

Context

- Non-visual control
- Manages all other controls

Task

- Container for functionality

Toolbar

- Container for tools and commands



What do the Web ADF controls do?

- **Provide a gateway to resource specific APIs**
- **Provide an extensible framework**
 - Through event model and action listeners
 - Ability to associate multiple renderers (e.g., HTML, WML, DHTML)
- **Manage state between HTTP requests**
- **Manage relationships with other JSF components**

Demo

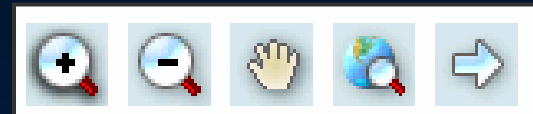
- **Create a new JSP page in Eclipse**
- **Explore the web control tags**
 - Intellisense, Required attributes
- **Add WebControls**
 - Context
 - Map
 - Overview
 - TOC

Topics

- Introduction to the Web ADF
- Web ADF controls
- **Toolbars, commands and tools**
- **Tasks**
- Introduction to developer APIs
 - Common API
 - Resource Specific APIs
 - Library Specific APIs
- **Graphics**

Toolbar control

- Displays a collection of tools and commands
- Associated to a Map control
- Contains a collection of toolbar items
 - Tool
 - Command
 - Drop-down list
 - Separators
- Required attributes
 - mapId
 - id



```
<a:toolbar mapId="map1" id="toolbar1"  
    activeTool="toolZoomIn" >
```

Commands

- **Server side action triggered by events**
 - Example: User clicks on *Full Extent* button
- **No interaction with Map required**

- **Out of the box commands**
 - `DirectionalPanListener`
 - `ZoomFullExtentListener`

- **Creating commands**
 - Bind a Command control to a server-side action listener
 - When a command button is clicked, listener class processes business logic

Tools

- **Consist of client-side and server-side actions**
- **Client-side action**
 - Processing carried out on the browser client
 - Controlled by JavaScript functions
- **Server-side action**
 - Business logic performed on the server
 - `com.esri.adf.web.faces.event`
 - Zoom In, Zoom Out, Pan etc..
 - Custom method

Tool control

- Executes a client action followed by a server action
- Client post back
- Attributes
 - clientAction
 - JavaScript function
 - Performed on the client
 - serverAction
 - Processes the request
 - serverMethod
 - Points to custom method

- ◆ EsriMapRectangle
- ◆ EsriMapPan
- ◆ EsriMapContinuousPan
- ◆ EsriMapPoint
- ◆ EsriMapLine
- ◆ EsriMapPoly
- ◆ EsriMapOval

- ◆ ZoomInToolAction
- ◆ ZoomOutToolAction
- ◆ PanToolAction

```
myMethod(MapEvent e){ }
```

Topics

- Introduction to the Web ADF
- Web ADF controls
- Toolbars, commands and tools
- **Tasks**
- Introduction to developer APIs
 - Common API
 - Resource Specific APIs
 - Library Specific APIs
- **Graphics**

Web ADF tasks

- **Objects that encapsulate business logic**
- **Out-of-the-box tasks configurable using Manager or IDE**

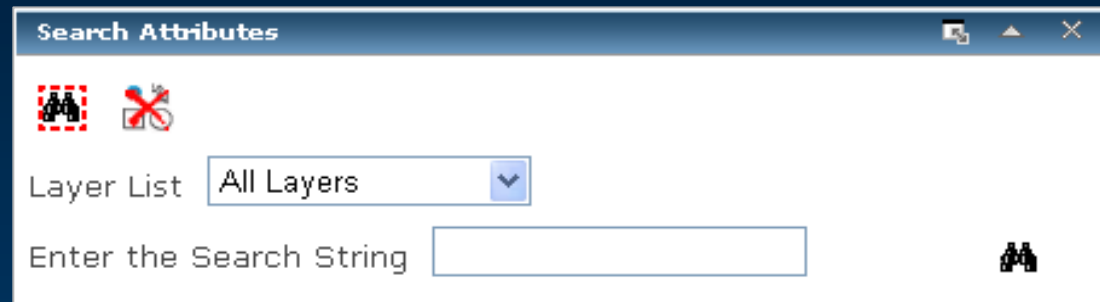
Web ADF tasks

- Objects that encapsulate business logic
- Out-of-the-box tasks configurable using Manager or IDE
- Available tasks
 - Map Tools



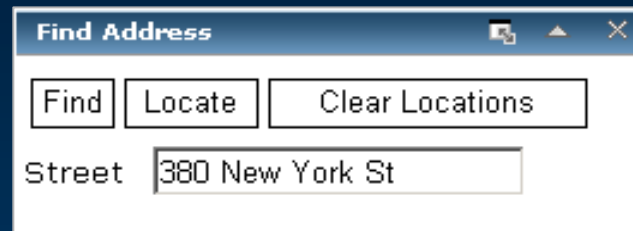
Web ADF tasks

- Objects that encapsulate business logic
- Out-of-the-box tasks configurable using Manager or IDE
- Available tasks
 - Map Tools
 - Search Attributes



Web ADF tasks

- Objects that encapsulate business logic
- Out-of-the-box tasks configurable using Manager or IDE
- Available tasks
 - Map Tools
 - Search Attributes
 - Geocoding



Web ADF tasks

- Objects that encapsulate business logic
- Out-of-the-box tasks configurable using Manager or IDE
- Available tasks
 - Map Tools
 - Search Attributes
 - Geocoding
 - Editing
 - Geoprocessing

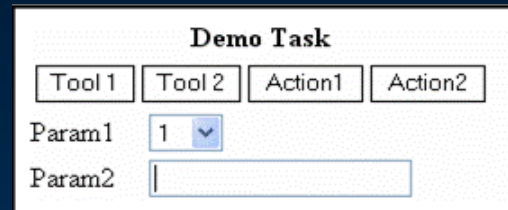
The screenshot shows a web application window titled "Clip Parcels with Buffer". The window has a standard browser-style title bar with a maximize button, a double-left arrow button, and a close button. Below the title bar, the main content area is titled "Input Features". It contains a large empty rectangular box for feature selection. To the left of this box are two vertical buttons with upward and downward arrows. To the right are three buttons: a checkmark, an 'X', and a grid icon. Below the box, the text "0 feature(s) entered." is displayed. At the bottom of the window, there is a "Linear unit" section with a text input field containing "50" and a dropdown menu set to "Meters". There are also "Help" and "Submit" buttons at the bottom.

Task framework

- Exposes a way to add custom functionality

Task framework

- Exposes a way to add custom functionality
- Tasks can contain:
 - Parameters
 - Commands/actions
 - Tools



The screenshot shows a window titled "Demo Task". At the top, there are four buttons labeled "Tool 1", "Tool 2", "Action1", and "Action2". Below these buttons, there are two parameter fields. "Param1" is a dropdown menu currently showing the value "1". "Param2" is an empty text input field.

Task framework

- Exposes a way to add custom functionality
- Tasks can contain:
 - Parameters
 - Commands/actions
 - Tools
- **Why use the Task framework?**
 - Tight integration with ADF
 - **User interface (UI) built for you**
 - **Event handling for commands and tools**

Task framework

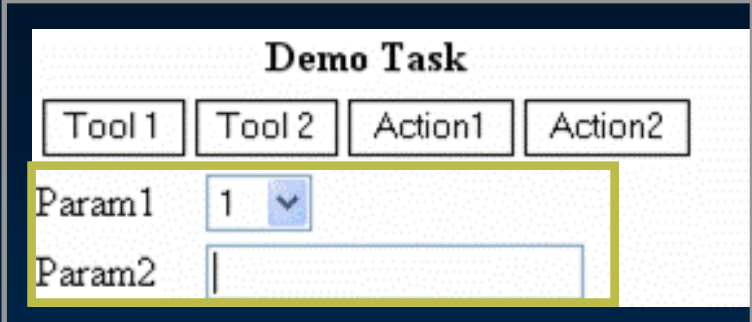
- Exposes a way to add custom functionality
- Tasks can contain:
 - Parameters
 - Commands/actions
 - Tools
- **Why use the Task framework?**
 - Tight integration with ADF
 - User interface (UI) built for you
 - Event handling for commands and tools
 - Organizes related functionality

Task framework

- Exposes a way to add custom functionality
- Tasks can contain:
 - Parameters
 - Commands/actions
 - Tools
- Why use the Task framework?
 - Tight integration with ADF
 - User interface (UI) built for you
 - Event handling for commands and tools
 - Organizes related functionality
 - Development similar to implementing standard JavaBean

Anatomy of a task

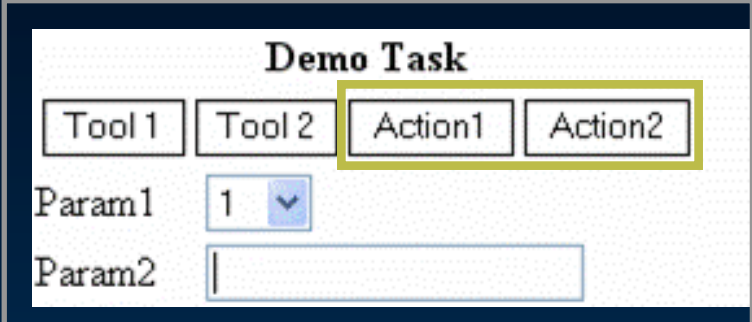
- **Parameter**
 - Provides inputs to a task
 - Examples: Layer name, zoom factor



The screenshot shows a user interface for a 'Demo Task'. At the top, the title 'Demo Task' is centered. Below the title, there are four buttons labeled 'Tool 1', 'Tool 2', 'Action 1', and 'Action 2'. Underneath these buttons, there are two parameter input fields. 'Param1' is a dropdown menu currently showing the value '1'. 'Param2' is a text input field. A yellow rectangular box highlights the 'Param1' dropdown and the 'Param2' text input field.

Anatomy of a task

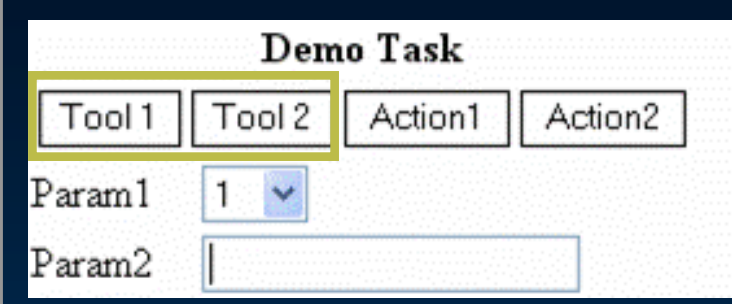
- **Parameter**
 - Provides inputs to a task
 - Examples: Layer name, zoom factor
- **Command (action)**
 - Executes business logic without user interaction with the map
 - Example: Zoom to full extent



The screenshot shows a user interface titled "Demo Task". At the top, there are four buttons: "Tool 1", "Tool 2", "Action1", and "Action2". The "Action1" and "Action2" buttons are highlighted with a yellow border. Below the buttons, there are two parameter fields. "Param1" is a dropdown menu currently showing the value "1". "Param2" is an empty text input field.

Anatomy of a task

- **Parameter**
 - Provides inputs to a task
 - Examples: Layer name, zoom factor
- **Command (action)**
 - Executes business logic without user interaction with the map
 - Example: Zoom to full extent
- **Tool**
 - Requires user interaction with the map
 - Requires client-side action
 - Examples: Identify, dynamic navigation (e.g., Zoom In/Out)



The screenshot shows a user interface titled "Demo Task". At the top, there are four buttons: "Tool 1", "Tool 2", "Action1", and "Action2". The "Tool 1" and "Tool 2" buttons are highlighted with a yellow border. Below the buttons, there are two parameter fields. "Param1" is a dropdown menu currently showing the value "1". "Param2" is an empty text input field.

Steps to implement a custom task

1. Create a standard Java Class

- Signature of method dictates a command or tool

2. Register the Java class as a managed bean in the faces-config

- Makes it available in the ADF

3. Add a task tag to the jsp, with reference to the managed bean

- Visually displays the UI to the user

Example: Implement a custom task (1)

① Create a standard Java Class

MyTask.java

```
package com.demo;  
public class MyTask {  
  
}
```

② Register Java class as a managed bean

faces-config.xml

```
<managed-bean>  
  <managed-bean-name>myTask</managed-bean-name>  
  <managed-bean-class>com.demo.MyTask</managed-bean-class>  
  <managed-bean-scope>session</managed-bean-scope>  
</managed-bean>
```

Example: Implement a custom task (2)

③ Add the control to the jsp

- Value attribute = value of managed-bean-name in faces-config

map.jsp

```
<a:task value="#{myTask}" mapId="map1" />
```

faces-config.xml

```
<managed-bean>  
  <managed-bean-name>myTask</managed-bean-name>
```


Demo

- **Adding a custom task to the web mapping application**

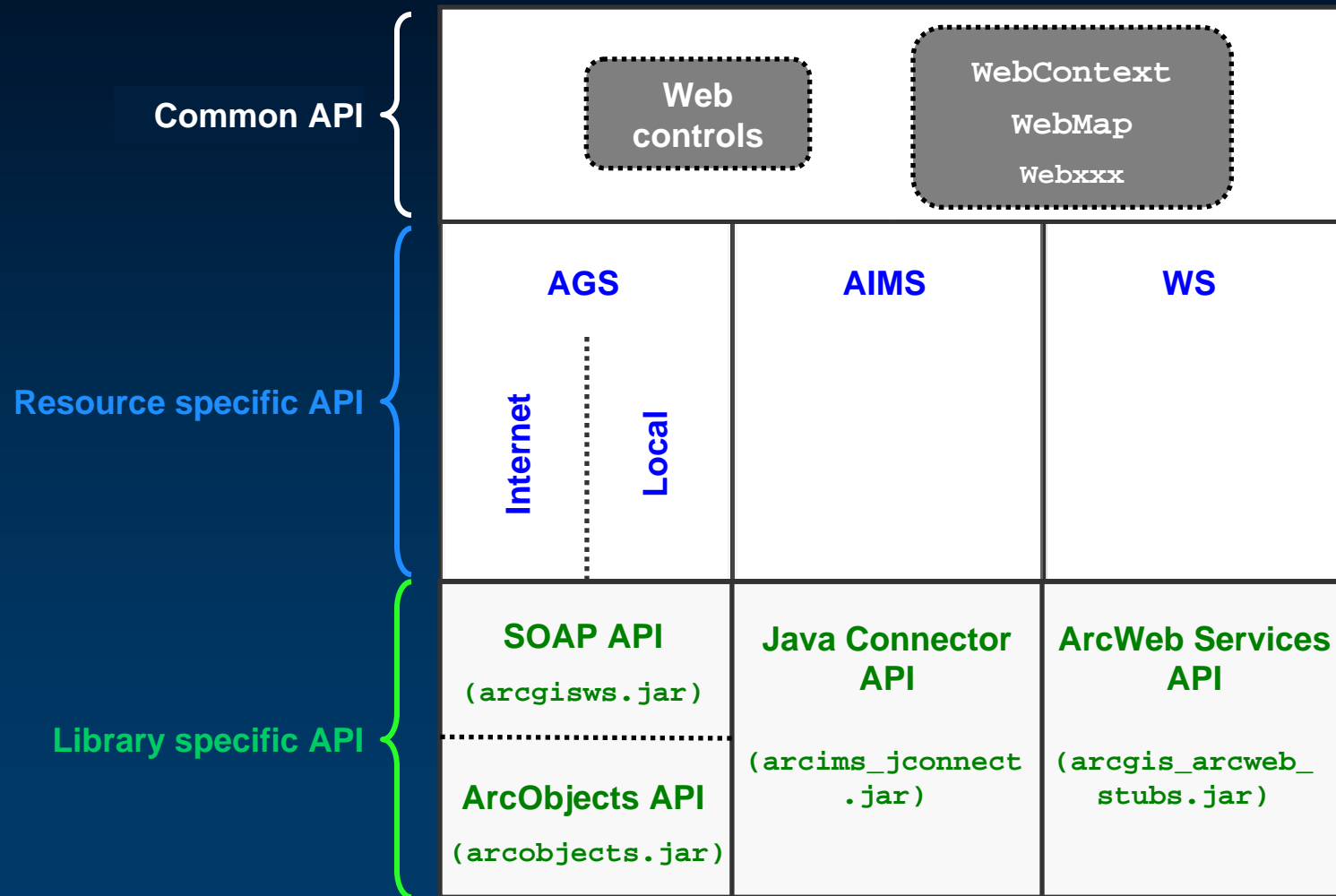
Topics

- Introduction to the Web ADF
- Web ADF controls
- Toolbars, commands and tools
- Tasks
- **Introduction to developer APIs**
 - Common API
 - Resource Specific APIs
 - Library Specific APIs
- Graphics

The Web ADF API

- **Web ADF can be divided into different areas:**
 - **Common API**
 - **WebControls**
 - **Resource specific APIs**
 - **ArcGIS Server, ArcIMS, ArcWeb Services, WMS**
- **Web ADF APIs use:**
 - **Library specific APIs based on type of resource**
 - **ArcObjects, Java Connector etc..**

Web ADF development paths



Organization of the Developer APIs

com.esri.adf.web
com.esri.adf.web.data
com.esri.adf.web.data.geometry
com.esri.adf.web.data.query
com.esri.adf.web.data.renderer
com.esri.adf.web.data.results
com.esri.adf.web.data.symbol
com.esri.adf.web.data.tasks
com.esri.adf.web.faces.*
com.esri.adf.web.tasks
com.esri.adf.web.util

Common API

com.esri.adf.web.wms.*
com.esri.adf.web.ags.*
com.esri.adf.web.aims.*
com.esri.adf.web.aws.*
com.esri.adf.web.ejb.*

Resource Specific API

com.esri.arcgisws.*
com.esri.aims.mtier.*
com.esri.arcgis.*

Library Specific API

Common API

- Provides a generic way for Web controls to access functionality
- Exposes commonly used functionality
 - Query, QueryCriteria, Highlight, Zoom etc..
- Exposes business objects
 - WebMap, WebQuery, WebGraphics, WebResults
- Gives access to other APIs



Advantages of the Common API

- **Removes business and GIS logic from Web controls**
- **Support for multiple data sources using the same API**
- **Easy to program against different data sources**
- **Implement your own custom data sources**
- **Code is portable**
- **Abstracts Implementation**

Resource Specific APIs

- **Each resource exposes a different set of functionality**
 - ArcGIS Server - SOAP, ArcObjects – (HTTP , DCOM)
 - ArcIMS – ArcXML – (TCP, HTTP)
 - ArcWeb Services – SOAP (HTTP)
- **What does this mean?**
 - Many other resource specific classes available
 - More value objects, business/GIS logic
 - Different APIs use different communication protocols
 - Requires different programming patterns for each data source
 - Available functionality depends on:
 - ① Underlying capabilities of the server object
 - ② Capabilities exposed through the ADF API

Library Specific APIs

- **Has more functionality available than common or Resource specific APIs**
- **Use it only when the higher level APIs do not provide the required functionality (Example: Buffer)**
 - Custom tasks to execute business logic
- **Example: ArcObjects API**
 - Gives you access to the server objects
 - MapServer, geocodeServer, etc.
 - Gives you access to fine grained ArcObjects
 - IMapServer, IMapServerInfo, IMapServerLayout etc

Topics

- Introduction to the Web ADF
- Web ADF controls
- Toolbars, commands and tools
- Tasks
- Introduction to developer APIs
 - Common API
 - Resource Specific APIs
 - Library Specific APIs
- **Graphics**

Graphics layer

- Drawn on top of layers in the map
- Used by other tasks
 - Highlighting features
 - Labeling text
 - Displaying buffers
 - Editing
 - Custom tasks
- Rendering occurs independently from the map
 - Does not require a map redraw



Using the graphics layer

- **WebQuery**
 - Uses a default symbol to display graphics
 - Can apply different point, line, or polygon symbols
 - Provides convenient access to the graphics layer
 - Ideal for displaying a few features in one color
 - Buffer, selected features, etc.

Using the graphics layer

- **WebQuery**

- Uses a default symbol to display graphics
- Can apply different point, line, or polygon symbols
- Provides convenient access to the graphics layer
- Ideal for displaying a few features in one color
 - Buffer, selected features, etc.

- **WebGraphics**

- Uses a renderer to display graphics
 - More control over display
- Can render based on attribute values
- Ideal for displaying large number of features

Summary

- **Introduction to the Web ADF**
- **Web ADF controls**
- **Toolbars, commands and tools**
- **Tasks**
- **Introduction to developer APIs**
 - **Common API**
 - **Resource Specific APIs**
 - **Library Specific APIs**
- **Graphics**

In Conclusion...

- All sessions are recorded and will be available on EDN
 - Slides and code will also be available
- Please fill out session surveys!
- Still have questions?
 1. Tech talk, Demo Theatres, Meet the Team
 2. “Ask a Developer” link on web page
 - www.esri.com/devsummit/techquestions