



# Building Web Services with ArcGIS Server and Java

*Eric Miller*

*Antony Jayaprakash*

*Jay Theodore*

*Steven Citron-Pousty*

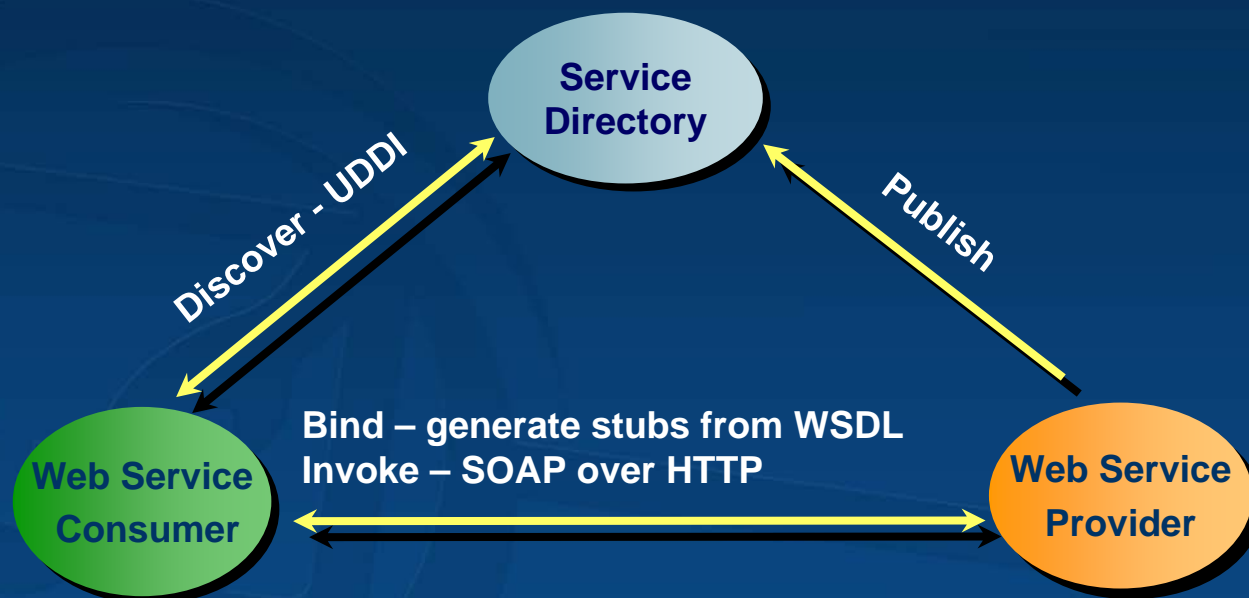
# Overview



- Web Services Overview
- ArcGIS Server Web Services
- Creating and Consuming ArcGIS Server Web Services
- Application Web Services
- Creating and Consuming Application Web Services
- New at 9.2
- API references and Documentation

# Web Services Overview

**Web Service:** a software component, described via WSDL, that is accessible via SOAP over HTTP.



- **ArcGIS Server Clients**
  - ArcMap
  - ArcCatalog
- **Custom Clients**
  - Web Applications
  - Desktop Applications

- **ArcGIS Server Web Services**
  - Catalog
  - Map Server
  - Geocode Server
- **Application Web Services**
  - Custom service



# AGS Web Services @ 9.1

- Exposes Server Objects as web services
  - MapServer
  - GeocodeServer
- Provides a pseudo discovery service
  - Web Service Catalog
    - Also a web service
    - Hosts server object web services
  - Published using web service catalog template.
- Interoperability
  - Consumed by both Java and .NET
- Standard web application security
  - Basic, Digest, Forms based authentication.
  - HTTPS channel encryption.

# AGS Web Services @ 9.1(cont)



- Designed for stateless consumption by ArcGIS clients; SOAP responses can be used by the AGSClient ArcObjects library.
  - ArcMap
  - ArcCatalog
  - Custom clients
- Implemented using JSP pages.
  - SOAP request is forwarded to server  
*HttpSoapHandler.sendSoapResponse(null,  
String soapRequest, IRequestHandler requestHandler, String out);*
  - Each server object type implements IRequestHandler which handles SOAP input by parsing it into native ArcObjects requests.

# ArcGIS Server (AGS) Web Services

## Discovery and Request



### Client Applications

- ArcMap
- ArcCatalog



1. SOAP request/response for URLs of hosted web services



2. SOAP request/response to/from MapServer Web Service

3. DCOM Call with embedded SOAP request/response

### ArcGIS Server



### Catalog Web Service

USA\_MapServer Web Service

Conservation\_MapServer Web Service

Germany\_MapServer Web Service

Portland\_GeocodeServer Web Service

### 4. Server Object Implements IRequestHandler

# Creating AGS Web Services



```
C:\WINDOWS\system32\cmd.exe

C:\Program Files\ArcGIS\DeveloperKit\Templates\Java>ls
BufferSelection  Search          build           error.jsp
Geocoding        Thematic       build.xml       images
MapView         WEB-INF        common.properties  js
PageLayoutViewer WebServiceCatalog  css            timeout.htm

C:\Program Files\ArcGIS\DeveloperKit\Templates\Java>arcgisant build
```

ArcGIS Server Java ADF

Create a web application from a template

1. Application name:

2. Connect to a GIS Server

GIS Server:

Connect to server as:

Domain:

Username:

Password:

3. Choose template and set properties

Name:

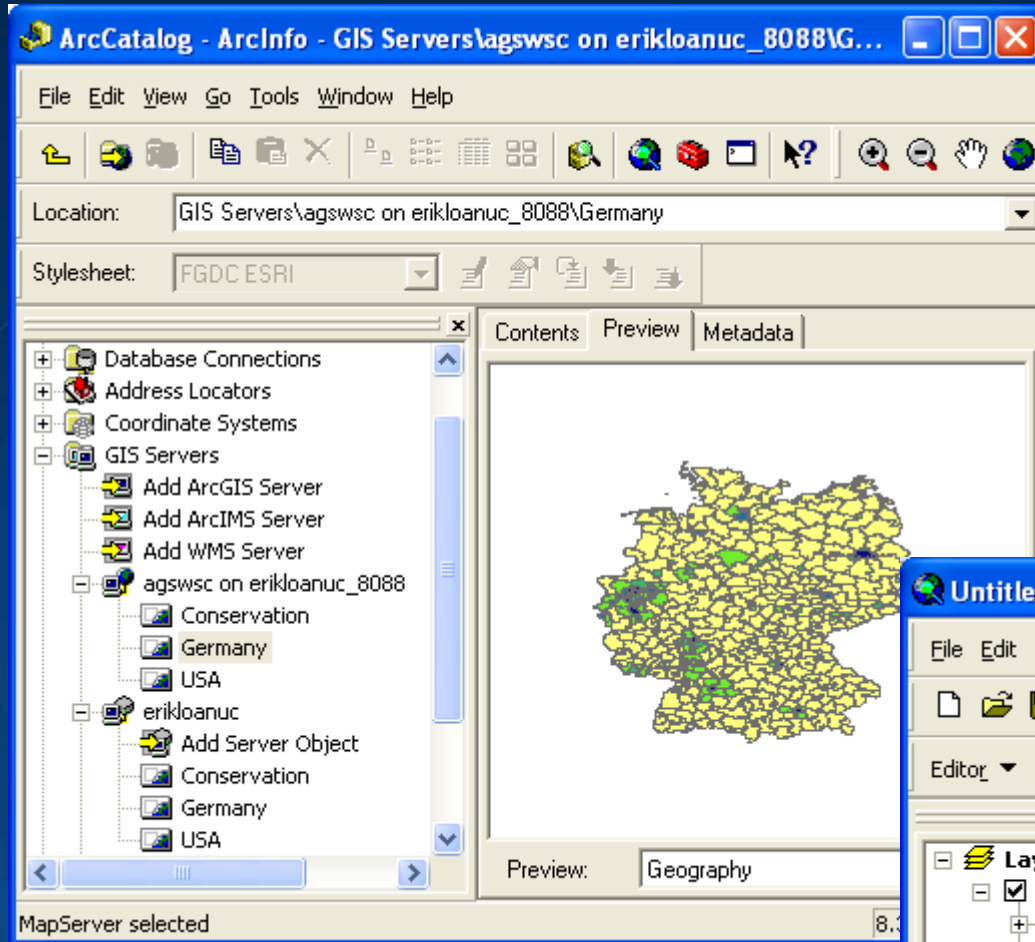
Map Server

- Conservation
- Germany
- USA

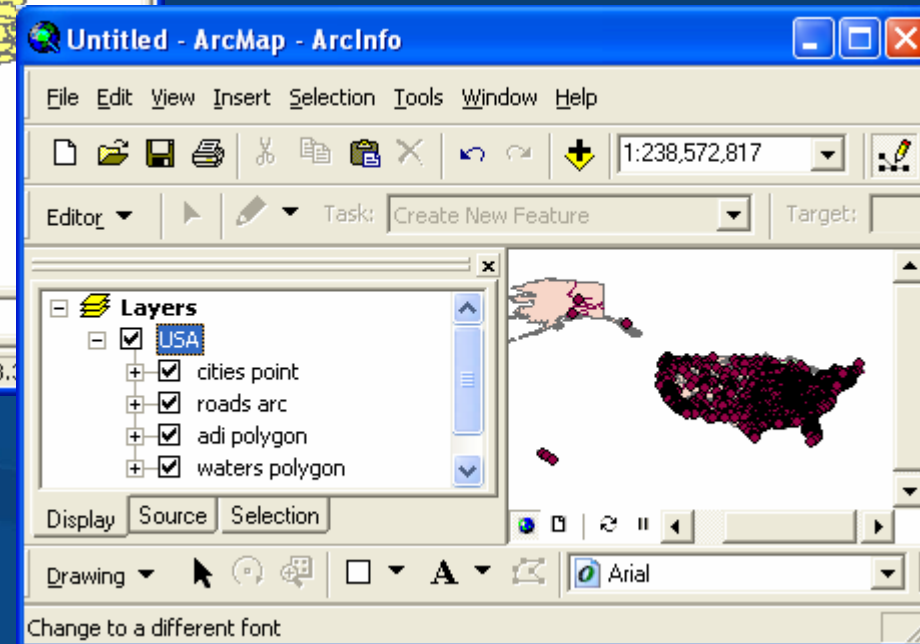
A simple matter of:

- Create a Server Object configuration.
- Start the Server object configuration.
- Create a web service catalog web service from template in Developer Kit.

# Consuming AGS Web Services



ArcGIS Desktop Clients



# Consuming AGS Web Services (cont)



## Custom Clients

The screenshot shows the 'Map Server Browser' application interface. At the top, there are navigation icons (home, back, forward, globe, hand) and a 'Bookmark' dropdown menu set to '<Default Extent>'. Below this, the 'Web Service Catalog URL' is set to 'http://erikloanuc:8088/agswsc/default.jsp' with a 'Get Web Services' button. The 'Map Server Web Service' is set to 'http://erikloanuc:8088/agswsc/Conservation\_MapServer.jsp'. The 'Data Frame' is set to 'Layers'. The main map area displays an aerial view with several overlays: two large blue circular areas labeled 'Pivot', a blue line representing a water body or boundary, and several red areas labeled 'Flood'. A legend in the bottom right corner of the map area identifies the red areas as 'Flood' and provides specific details: 'Pivot: 24.6 Acres (Implemented)' and 'Gated Pipe: 43.0 Acres (Approved)'. The map also shows a grid of green lines and a central road.



# Consuming AGS Web Services (cont)



## Custom Client Implementation:

### Get MapServer Web Service URLs from ServiceCatalog.

```
// get all MapServer web services and add their URLs
// to the MapServer JComboBox
URL url = new URL(urlTextField.getText());
ServiceCatalogBindingStub servicecatalog = (ServiceCatalogBindingStub)
    new _defaultLocator().getServiceCatalogPort(url);
ArrayOfServiceDescription sdArray =
    servicecatalog.getServiceDescriptions();
ServiceDescription sd[] = sdArray.getServiceDescription();

// add MapServers to the JComboBox
for (int i = 0; i < sd.length; i++) {
    if ( sd[i].getType().equals("MapServer") ) {
        mapserverComboBox.addItem(sd[i].getUrl());
    }
}
```

# Consuming AGS Web Services (cont)



## Custom Client Implementation:

### Get data frames from MapServer Web Service.

```
// Connect to MapServer web service
MapServerBindingStub mapserver =
    (MapServerBindingStub) new MapLocator().getMapServerPort(url);
dfComboBox.removeAllItems();
int numMaps = mapserver.getMapCount();
for (int i = 0; i < numMaps; i++) {
    dfComboBox.addItem(mapserver.getMapName(i));
}
```

# Consuming AGS Web Services (cont)



## Custom Client Implementation:

### Get map document bookmarks from MapServer Web Service.

```
MapServerBindingStub mapserver = (MapServerBindingStub)
    new MapLocator().getMapServerPort(url);
String name =
    mapserver.getMapName(dfComboBox.getSelectedIndex());
MapServerInfo mapInfo = mapserver.getServerInfo(name);
ArrayOfMapServerBookmark bookmarkArray =
    mapInfo.getBookmarks();
MapServerBookmark[] bookmarks =
    bookmarkArray.getMapServerBookmark();
if ( bookmarks != null ) {
    for (int i = 0; i < bookmarks.length; i++) {
        bookmarkComboBox.addItem(bookmarks[i].getName());
    }
}
```

# Consuming AGS Web Services (cont)



## Custom Client Implementation:

Draw map at extent of map document bookmark.

```
MapServerInfo mapInfo = mapserver.getServerInfo(name);
MapDescription mapDesc = mapDescription;
ArrayOfMapServerBookmark bookmarkArray = mapInfo.getBookmarks();
MapServerBookmark[] bookmarks =
    bookmarkArray.getMapServerBookmark();
if ( bookmarks != null ) {
    for (int i = 0; i < bookmarks.length; i++) {
        if (bookmarks[i].getName().equals(bookmarkComboBox.
            getSelectedItem().toString())) {
            mapDesc.setMapArea(bookmarks[i]);
        }
    }
}
mapDescription = mapDesc;
drawMap(mapDescription);
```

# Consuming AGS Web Services (cont)



## Custom Client Implementation:

Create an image by calling `ExportMapImage()`.

```
ImageType it = new ImageType();
it.setImageFormat(EsrImageFormat.esrImageJPG);
it.setImageReturnType(EsrImageReturnType.esrImageReturnMimeData);
ImageDisplay idisp = new ImageDisplay();
idisp.setImageHeight(400);
idisp.setImageWidth(552);
idisp.setImageDPI(150);
ImageDescription iDesc = new ImageDescription();
iDesc.setImageDisplay(idisp);
iDesc.setImageType(it);
MapImage mi = mapserver.exportMapImage(mapDesc, iDesc);
byte[] data = mi.getImageData();
mapimage = java.awt.Toolkit.getDefaultToolkit().createImage(data);
```



# Application Web Services

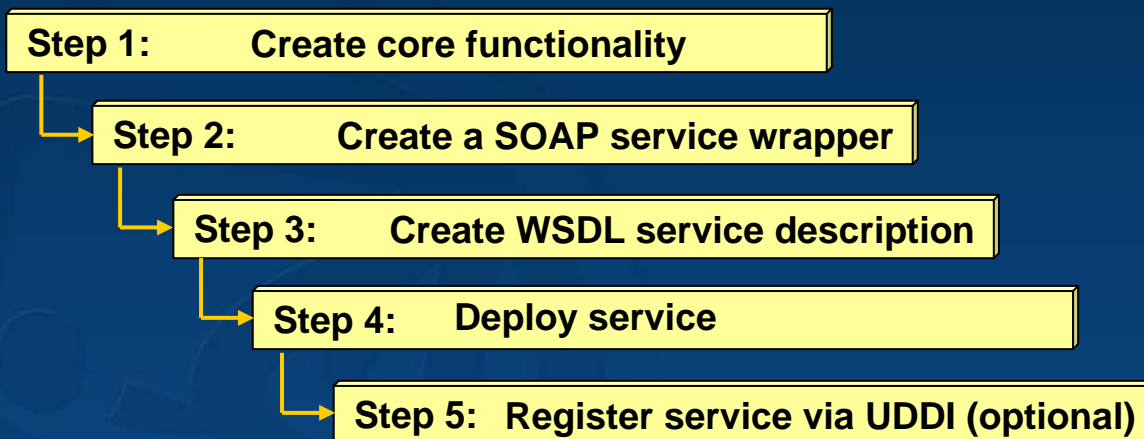


- Software components that accept and return simple non-ArcObjects Java types.
- Use the ArcObjects Java Server API to solve your business logic, but accept and return only simple Java types as input and output.
- Example:
  - Locate ideal locations to build a school by building farthest from toxic waste sites
    - [in] double x, [in] double y
    - [out] double distance

# Creating Application Web Services



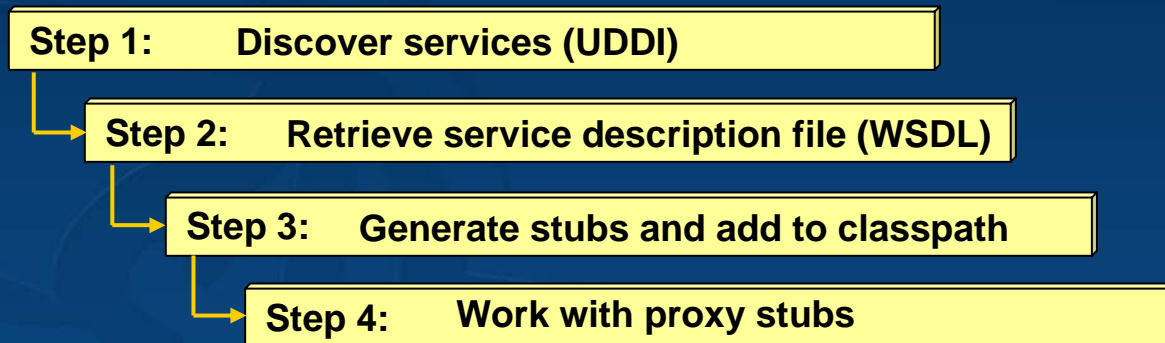
- Development plan



# Consuming Application Web Services



- Development plan



# New at 9.2 - AGS Web Services



- Exposes Server Objects and Extensions as web services
  - MapServer (KML, WMS, NA, Mobile), GlobeServer, GeocodeServer, GeoDataServer
- Capabilities can be restricted or enabled
  - MapServer – map, query, data
  - GeocodeServer – geocode, reversegeocode
  - GeoDataServer – query, extraction, replication
  - GlobeServer – globe, animation, query
- Provides a pseudo discovery service
  - Web Service Catalog
    - Also a web service
    - Hosts server object SOAP web services.
    - Hosts server object binary endpoints.
  - Published using standalone publisher (under development)

# New at 9.2 - AGS Web Services (cont)



- Designed for stateless consumption by ArcGIS clients; SOAP responses can be used by the AGSClient ArcObjects library.
  - ArcMap
  - ArcGlobe
  - ArcExplorer 2
  - ADF Clients
  - ArcCatalog
  - Custom clients
- Implemented using a Java Servlet
  - SOAP request is forwarded to server  
*HttpSoapHandler.sendSoapResponse(null, String soapRequest, IRequestHandler requestHandler, String out);*
  - Each server object type implements IRequestHandler which handles SOAP input (IRequestHandler.HandleStringRequest()) by parsing it into native ArcObjects requests.
  - Binary requests are also forwarded to the server and handled by IRequestHandler.HandleBinaryRequest().

# New at 9.2 - AGS Web Services (cont)



## Publishing ArcGIS Server Web Services

**ArcGIS Server Administration** Sign Out »

Home | Server Objects | **Web Service Catalogs** | Server Properties | Monitor

Updated: Thursday, March 16, 2006 8:59:23 PM

Create Catalog | Cancel

---

**Create Web Service Catalog**

Catalog Name:

Message Format:  SOAP  Binary

Contents:

Name Type/Extension  
Click add to add items to the catalog

**Add items to the catalog** [X]

<input type="checkbox"/>	Name	Type/Extension
<input type="checkbox"/>	Germany	MapServer
<input type="checkbox"/>	Germany	KmlServerExtension
<input type="checkbox"/>	Japan	GlobeServer
<input type="checkbox"/>	Portland	GeocodeServer
<input type="checkbox"/>	SC	MapServer
<input type="checkbox"/>	SC	KmlServerExtension
<input type="checkbox"/>	Win2kTest	GeoDataServer
<input type="checkbox"/>	southern_california	MapServer

Done



# With ArcGIS Server Manager You Can:



- Create and deploy GIS Services
- Create and deploy Web Applications
- Create and deploy Mobile Applications
- Manage and Monitor GIS Services
- Build custom services and applications

# ArcGIS Server Manager: Complete Solution



- Single consolidated view of ArcGIS Server
- Zero programming web-application builder
- Simplified publishing model just one-click away
- Centralized management and monitoring

A screenshot of the ArcGIS Server Manager web interface. The page has a blue header with the title "ArcGIS Server Manager" and the ESRI logo. Below the header, there is a navigation bar with tabs for "Home", "Applications", "Services", and "Manage &amp; Monitor", along with "Preferences" and "Help" buttons. The main content area features a welcome message: "Welcome to ArcGIS Server" and "ArcGIS server offers a complete end-to-end solution for enhanced decision support by offering web-based GIS to integrate spatial capabilities across an entire organization." Below this, there is a section titled "Take a 360 degree tour of ArcGIS Server" with a small image of a map. To the right of the image is a list of four tasks, each with a green checkmark icon: "Create and deploy GIS Services", "Create and deploy web applications", "Manage and Monitor GIS services", and "Build custom services and applications". At the bottom of the main content area, there are two buttons: "Getting Started and Help with ArcGIS Server" and "My Application and Services Shortcuts".

# Simplified Publishing .....Just One-Click Away



ArcGIS Server Manager

ArcGIS Server Manager You are connected as L\_am\_admin

Home Applications **Services** Manage & Monitor Preferences Help

Here is a summary of all the GIS resources that have been exposed as service for consumption by web clients. Use the **Publish GIS Resource** button to publish your GIS resource as a service. Use the **New Service** button to create a GIS Service based on an existing GIS resource

[Settings](#)

**default**

[Add New Folder](#) [Edit Folders](#)

- [default](#)
- [caIGC](#)
- [newzealand](#)
- [redlands](#)

<input type="checkbox"/>	Services	Capabilities	Description	Status	Type
<input type="checkbox"/>	<a href="#">SanBernardino_Trails</a>	capabilities	Trail Maps for Hiking		
<input type="checkbox"/>	<a href="#">IrrigationCanals</a>	capabilities	Canals for sugarcane fields		

# Integrated Server Administration ...Point and Click Ease



ArcGIS Server Manager

ArcGIS Server Manager You are connected as **L\_am\_admin**

Home Applications Services **Manage & Monitor** Preferences Help

Server Configuration Monitor

Listed below are machines that host your services. To add a host, specify the name of the machine that you want to use as a Server Object Container machine. Note: Machine must already have ArcGIS Server software installed.

**Hosts**

[hosts](#)  
[directories](#)

<input type="checkbox"/> <b>Server Object Container</b>	<b>Description</b>
<input type="checkbox"/>	<a href="#">nila</a>
<input type="checkbox"/>	<a href="#">mainhost</a>

# Features



- Easy to use Web Application Builder – zero programming
- Provides extensive support for Mapping, KML, OGC WMS, Network Analysis, Caching...
- Provides flexible deployment to J2EE environments
- Easy to customize applications
- Integrate GIS Services into Web Applications
- Access full power of ArcGIS Server within your applications
- Easy to use out-of-the box applications and tools

# API References and Documentation



- JavaDoc generated from WSDL-generated stub classes.
- Match descriptions to similarly named classes in Server API.
- Developer scenarios
- Samples

# Session Evaluations Reminder



Session Attendees:  
Please turn in your session evaluations.

*... Thank you*