



**Esri International User Conference | San Diego, CA**  
**Technical Workshops | July 13, 2011**

# **Creating Geoprocessing Services**

Kevin Hibma, Shing Lin

# Geoprocessing Services

- The geoprocessing service allows you to publish custom tools to be used via ArcGIS Server
- Geoprocessing services can be used by many different client applications
  - ArcGIS Desktop
  - ArcGIS Engine
  - ArcGIS Explorer
  - WSDL
  - REST
    - JavaScript
    - FLEX
    - Silverlight

# Geoprocessing Services

- The service is composed of both the tools and the data needed by the tools
- Endless array of tasks can be created
  - Spatial analysis (vector, raster, network...)
  - Data Management (geodatabase, file based data)
  - Conversion (ETL and data loading )
- You need to be knowledgeable about using geoprocessing tools to create a good geoprocessing service

# Geoprocessing Service Behavior

- **Geoprocessing Services are very flexible and allow many different behaviors**
- **Before Authoring and Publishing, identify what you want your service to do and how you want it to behave with clients.**
  - **Input data from the client or select data on the server?**
  - **Draw results with map server or download and draw data on the client?**
  - **Save data on the server?**

# Authoring a Tool Suitable for Publishing

- **Models and scripts that are run by services need to have certain characteristics to run correctly**
- **Changes will likely be required to allow existing models and scripts to be published**
- **Things to consider when creating a model for use as a Geoprocessing Service**
  - **Data Type of Parameters**
  - **Data Management (Source, Intermediate, Output)**
  - **Symbology**
  - **Optimization**

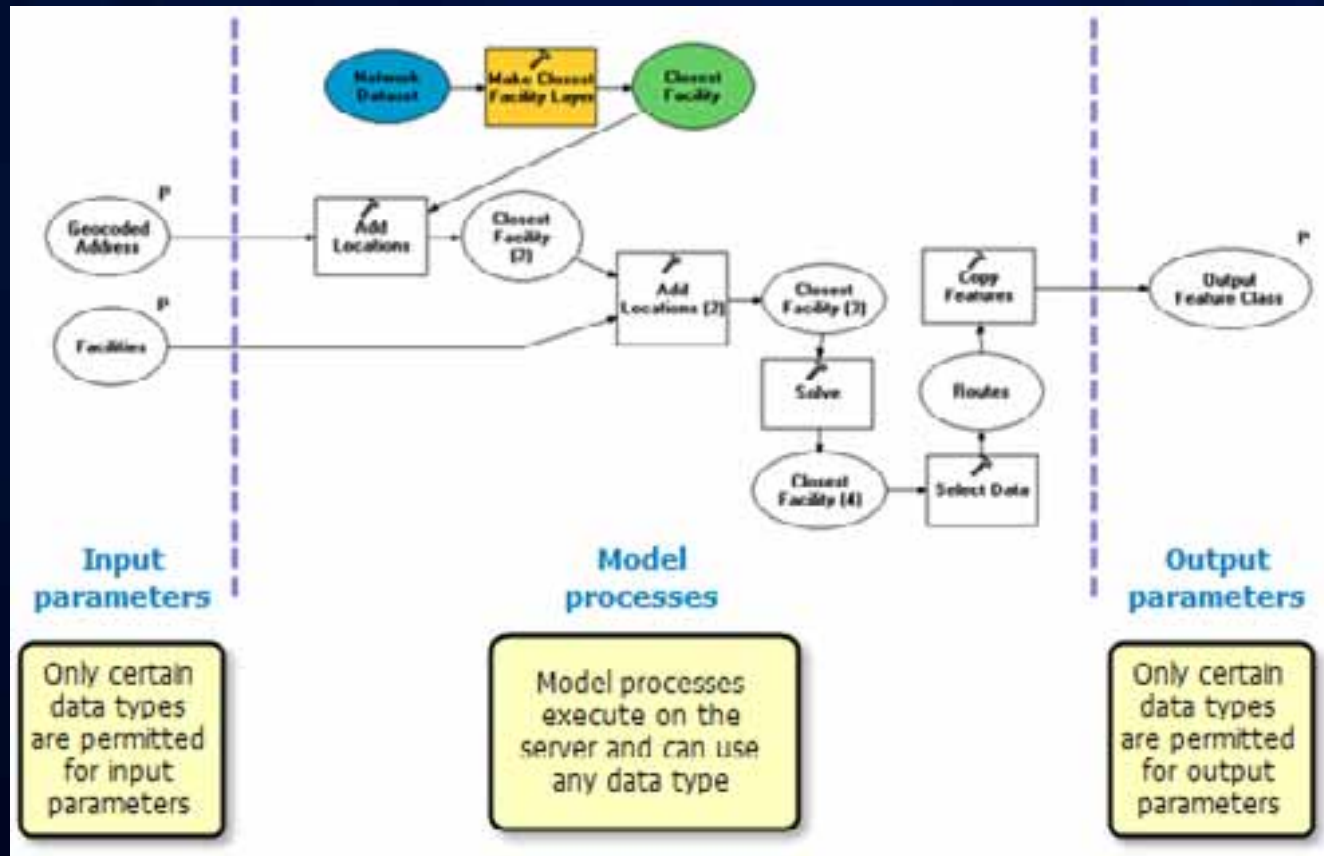
# Geoprocessing Service Parameter Types

- A subset of desktop data types are supported as valid parameters for a geoprocessing service
- Subset determined by light weight clients
  - ArcGIS Explorer
  - Web Mapping Applications
- Publishable tools need to be built using these parameter types

# Parameter Types

Parameter
Feature Set
Record Set
Raster
Boolean, Double, Long, Date, String
File
Linear Unit
Layer (Feature layer, Raster layer, etc)

# Geoprocessing Service Parameter Types





# Unsupported Input Types

- The data type will be converted to the String representation (which is supported). For example, “Areal Unit” or “Field”

Or

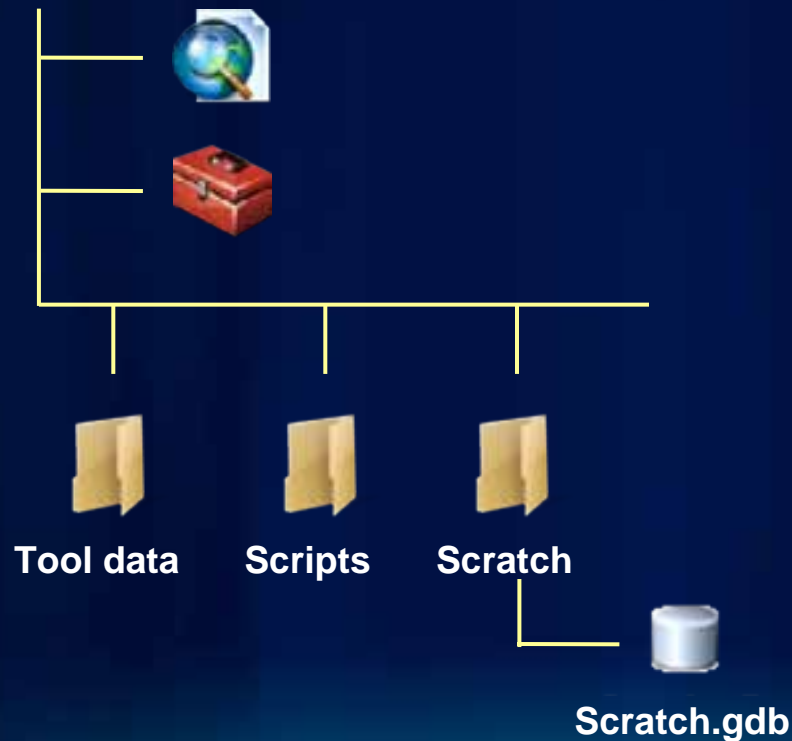
- Will receive an error when publishing that the parameter is invalid

# Unsupported Output Types

- Feature Class or Feature Layer converted to Feature Set
- Table or Table View converted to Record Set
- The data type will be converted to the String representation  
or
- Will receive an error when publishing that the parameter is invalid

# Good Patterns to Service Setup

- **Setting up your directory makes your tool portable and decreases the risk of breaking data paths**



- **The tool sharing directory contains all the contents of the geoprocessing service**
  - **Toolbox**
  - **Map**
  - **Scripts**
  - **Scratch workspace to test the service**
  - **Map to test the service**
  - **Tool data (optional)**
  - **SDE connection files**
  - **Other toolboxes used by the tools being published**
- **Set relative paths on your models, scripts, and map documents**

## Intermediate and Output

- When the server runs a model, the output and intermediate data should be written to the job directory
- Use the %scratchworkspace% inline variable in your output paths
  - %scratchworkspace%\outputfc.shp
- A file geodatabase named “scratch.gdb” is guaranteed to be in the job directory created for each task submission
  - %scratchworkspace%\scratch.gdb\outputfc

# Authoring a Tool Suitable for Publishing

- Check list for authoring and publishing a geoprocessing service
- <http://esriurl.com/gpCheckList>

# Demo

## Calculate Population

- Calculate total population based on input feature class
- Author a desktop tool so it can be published as a server tool
  - Modify Data Types:
    - Configure input parameter from **feature class** to **feature set**
  - Change output and intermediate paths to **%scratchworkspace%**
  - Symbology

# Geoprocessing Services Configurations

- **Geoprocessing Service from a toolbox**
  - Each tool becomes a task
  - Client draws the results



- **Geoprocessing Service from a map**
  - Each tool layer becomes a task
  - Use layers as inputs to the tool
    - Improve performance by pre-loading layers (Network Analyst Layer)
    - Choice list of layers on the server
  - Results can be viewed using a Result Map Service





# Tool Layer

- **What is a tool layer?**
  - A special group layer containing outputs of a tool
  - Defines parameter symbology
- **How to create?**
  1. Drag and drop a tool into a map
  2. Open tool dialog and run
    - Tool outputs are added as sub-layers

## **Use a Map Document if:**

- **Using a layer improves the performance of the tool**
  - **Network Analyst Layer**
  - **Select by Location or Select by Attribute**
- **Want a choice list of layers in the map document**
  - **Clip and ship with the ability to select the layer to clip**
- **Want to use a map service to draw the results**
  - **Cont...**

# Result Map Service

- A result map service (RMS) provides an alternative way to get results from the Geoprocessing Service.
- An image is returned to the client.
  - The data can still be downloaded.
- Use a RMS when:
  - Want better cartography than the client can support
  - It is impractical to render a large dataset in a client.
- Execution must be Asynchronous when using a RMS

# Asynchronous vs. Synchronous

- **Execution mode defines how the client interacts with service while it executes**
  - **Asynchronously:** client must ask the server if its finished then get the result. The client is free to do other work during this time.
  - **Synchronously:** the client waits for the server to finish executing and then gets the result.
  - **Can only use a Result Map Service with Async.**
  - **Synchronous services are typically fast services**

# Asynchronous

- **Asynchronous (Submit Job)**
  - Results are saved on the server
  - Results can be drawn on the server
  - Results can also be downloaded if desired
  - Clients free to do other tasks
    - you can pan/zoom, run other tools while the job is running
  - Appropriate for longer processing jobs.

# Synchronous

- **Synchronous (Execute)**
  - Client always receives and draws data.
  - Desktop Client waits until job is completed and results are returned
  - Appropriate for faster processing jobs. (<10 seconds)

# Demo

## Publishing and Consuming Services from ArcMap

- **Crime Hot Spot Analysis**
  - Data Type: Input is **feature set** and output is **raster** saved on the **server**
  - Execution Type: An **asynchronous** service with a result map service



## **Data Management – Source Data**

- **If the data is not large or centralized, it can be useful to package everything in a folder and use relative paths**
- **If the data is distributed around the network, use UNC paths when building the model or script tool**
- **Copy SDE Connection files relative to the toolbox**
  - **Will not find “database connections” path**



# Publishing Considerations

- **Maximum Number of Records**
  - This property limits the number of features returned from the server
    - The default is 1000
  - Prevents large amounts of data from being transported across the internet
  - Can be a gotcha. If your results don't display because of this reason, there should be a message in the tool messages
- **Show Messages**
  - Usually only used for debugging initial development

# Publishing Considerations

- **How long to keep results**
  - **Server administrator sets jobs directory clean up schedule**
  - **10 minutes, 1 hour, 1 day, unlimited? You decide**
- **Number of Instances**
  - **How many concurrent requests can run**
- **Timeout**
  - **How long before the service automatically kills itself**

# Tuning: UNC Paths

- Reading and writing data to UNC paths is slower
- If using one server machine avoid UNC where possible
  - Use local path for jobs directory
  - Use local path to source data if possible
- If using a distributed server (many machines)
  - Jobs directory must be a UNC path
  - Can use the **in\_memory** workspace for feature classes and tables
  - **Use LocalJobsDirectory setting!**

# Local Jobs Directory

- **Local Jobs Directory reduces the use of UNC paths.**
- **When specified all intermediate and output data are written to a local job directory**
- **If the service is Asynchronous the local job directory is copied to the main server jobs directory**
- **Only relevant if server is distributed or jobs directory is a UNC location.**

# Setting the Local Jobs Directory


ArcGIS Server - Geoprocessing Service Properties

General | Parameters | Capabilities | Pooling | Processes


Execution Type: ☐ Synchronous ☒ Asynchronous

The tools exposed by the Geoprocessing Service are stored in:

☒ A toolbox

Toolbox:  

☐ A map

Map Document:  

Data Frame:

Result Map Service:

Jobs Directory:

Virtual Jobs Directory:

Output Directory:

Virtual Output Directory:

Maximum Number of Records Returned by Server:

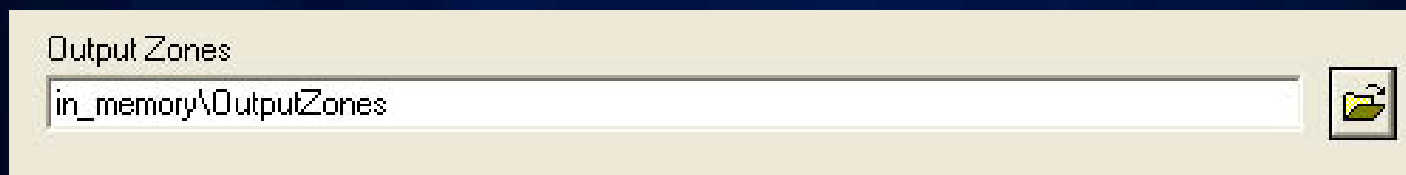
☐ Show Messages

☒ Local Jobs Directory

OK Cancel Apply

## Tuning: In Memory Workspace

- Data can be written to the “**in\_memory**” workspace.
  - Only for feature class and table
  - Only appropriate when overhead of writing to disk is significant portion of the total time it takes to run the model
  - If output is “in\_memory” the client must draw the result
  - Do not use for output that is drawn by the result map server



# Tuning: Pre-Processing

- **Pre-process any geoprocessing operations that you can and remove them from your model**
  - **Eg: A suitability model may use slope and aspect as criteria. It is not necessary to run slope every time the model is executed. Pre-process slope and aspect.**

# Tuning: Large Input Datasets

- Large input record set or raster
  - Increase the web server maximum size setting
    - C:\inetpub\wwwroot\ArcGIS\Services\web.config file

```
<system.web>
```

```
  <httpRuntime maxRequestLength="20000"></httpRuntime>
```

```
</system.web>
```



# Tuning: Timeouts

- Client wait timeout

C:\inetpub\wwwroot\ArcGIS

```
<system.web>
```

```
  <httpRuntime executionTimeout="600" />
```

```
</system.web>
```

- ArcGIS Server wait time out
  - If you are using synchronous execution with lots of users, you may need to increase the wait timeouts
- ArcGIS Server execution timeout
  - If the execution time is greater than 10 minutes, make sure to increase the execution timeout

# Troubleshooting

- **Publishing fails if these errors are detected**
  - Check path to the toolbox or map document
  - Check parameter data types
    - Need to rework the tool to limit the parameter types
  - Check of invalid sub-layers in the tool layer
    - Usually means the output data used to create the tool layer has been deleted or not visible to the map server
- **Most common problems**
  - Tool data paths not to the scratch workspace
  - Tool layer and sub-layer (parameter name) are the same

# Troubleshooting

- The tool will be run by the ArcGISSOC account which may be different than the login you used to author the tool
- Make sure the ArcGISSOC account has access to all the input data
- Make sure the ArcGISSOC account has access to other software used by the service
  - e.g. When using a custom .Net tool make sure the dll is usable by the ArcGISSOC account
- Debug by logging in as ArcGISSOC
  - Run in ArcMap if ArcGIS Desktop is installed on the server machine
  - Run with a python script if ArcGIS Desktop is not installed

# Troubleshooting

- Check the log files if you are getting errors
  - Via manager or on disk (ArcGIS\server\user\log)
- Debugging
  - Change to Asynchronous
  - Turn Messages On
  - Run and check Job Directory
    - See “Messages.xml”
    - See intermediate and output data

# Demo

## Create a PDF file from a PDX file

- A script tool to convert a PDX file to feature class, create map layers and save as a PDF file output
- Data type: both input and output are file
- Configuration type: publish from the tool
- Execution type: asynchronous

# Geoprocessing Service Documentation

- **Help location**
  - **Geoprocessing/Geoprocessing with ArcGIS Server**
- **Key topics**
  - **Check list for authoring and publishing a geoprocessing service**
    - **Guidelines to avoid most common problems**
  - **Key concepts for geoprocessing services**
  - **Input and Output data types**

# Sessions of Interest

- **Wednesday**

- **Geoprocessing Services in JavaScript** (12-12:30 Spatial Analysis Demo Theater in the Showcase)
- **Geoprocessing Services in Silverlight** (12:30-1 Spatial Analysis Demo Theater in the Showcase)
- **Debugging a Geoprocessing Service** (1:00-2:00 Spatial Analysis Demo Theater in the Showcase)
- **Performance tips for a Geoprocessing Service** (2-2:30 Spatial Analysis Demo Theater in the Showcase)

- **Thursday**

- **Creating Geoprocessing Services** (1:30-2:45 14B, repeat)
- ArcGIS Server: Road Ahead



esri