



Creating Geoprocessing Services

Jason Pardy, Arnaud Ferrand

**Esri Developer
Summit Europe**

11-13 November 2013
Park Plaza Riverbank London



Questions for you

- **Have you created Geoprocessing tools (models, scripts, COM)?**
- **Have you created Geoprocessing Services?**
 - **What version of ArcGIS?**

What are Geoprocessing services?

- A geoprocessing task that takes data captured in a web application, processes it, and returns meaningful and useful output in the form of features, maps, reports, and files
- Services can be used by many client applications:
 - ArcGIS for Desktop
 - ArcGIS for Engine
 - ArcGIS for Explorer
 - REST
 - JavaScript
 - FLEX
 - Silverlight

To create good Geoprocessing Services:

1. Knowledge about using geoprocessing tools and environments is important for knowing how to create a good geoprocessing service
2. Knowing and understanding the input data required for the service is important

Geoprocessing Service Behavior

- Before authoring or publishing, identify what you want your service to do and how you want it to behave with clients:
 - Does input data come from the client or select it from the server?
 - Draw the results with map server or download and draw data on the client?

How to create a service

- Changed at 10.1 – easier
- All services start from a successful **result**
 - The result acts as a template to build the service
- Quick tour of Publishing:
<http://esriurl.com/gpSrvQuick>

Workflow to create a Geoprocessing Service

- Create tool
- Document tool
- Run/Test tool
- From Results Window, share as service
- Set service name, parameters, etc in the Service Editor
- Analyze
- Publish
- Consume

Document your tool

- **Tools must be documented**
 - Parameters, description, tags, etc.
- **Best practice to fill out the item description**
- **You can update metadata specific to the task inside the Service Editor**

Parameter transformation

- Parameter types are converted to supported types when publishing
- You can update the Input Mode depending on the parameter type
 - User defined value
 - Choice list
 - Constant value

Data Store

- **New concept @ 10.1**
 - Without a data store entry, all required data is copied to the server
 - Either an enterprise database or folder
 - All data can live here where all servers have access to this data (can be local or unc)

- **Data Store: <http://esriurl.com/datastore>**

Data Store

- **Must specify data store folder**
 - LAS Dataset
- **If these exist in the data store, analyzer will generate and error. You must convert to file geodatabase:**
 - Access (.mdb)
 - Coverage feature classes
 - INFO table
 - Excel table

Execution Mode

- Defines how the client interacts with the service while it executes
 - **Synchronously:** client waits for the server to finish executing and then gets the result
 - **Asynchronously:** client must ask the server if its finished, then get the result. The client is free to do other work during this time.
 - ** Can only use Result Map Server with Asynchronous
 - ** Synchronous services are typically fast services
 - ** If the result is a nontransportable dataset such as a TIN or CAD drawing, the only way to send geography to the client is with a result map service

Demo

Creating Watershed Service

Synchronous execution
No data store

Result Map Service (RMS)

- Provides an additional way to get results from a 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
 - Impractical to render a large dataset in a client
- Execution must be Asynchronous when using RMS

Demo

Creating LandUse Service

Asynchronous execution

Data store

GP service settings: Messages

- **Message levels:**
 - None
 - Informative
 - Warnings
 - Errors
- **CAUTION:** All messages, regardless of level, may contain dataset paths and names, and this may pose a security risk. The Info level is verbose and typically contains more references to dataset paths and names. In general, you will want to return messages during development but turn them off in production.

Output & Intermediate data paths

- **Model Tools:**

- Intermediate data is deleted automatically after the service executes
- %scratchGDB%\output
- %scratchFolder%\out.shp
- in_memory\output

Script Tools:

- Paths and data handled same as models
- `os.path.join(arcpy.env.scratchGDB, "output")`
- `os.path.join(arcpy.env.scratchFolder, "output.shp")`
- in_memory\output

Publishing – Best practices

- Number of instances per machine = number of cores (**never more than number of cores)
- Have at least 2GB memory per instance, more is better
- If execution time > 10 minutes, increase the max execution time
- If uploading large files or raster (>45 GB), increase heap size
- Publish multiple results at one time (multiple tasks per GP Service). Consumes less resources and memory.

Publishing – Best practices

- Register large dataset location with the server data store to prevent copying large datasets during publishing
- Two ways to create data store:
 - Share the data location between desktop and server and register the shared location (OR)
 - Copy data manually before publishing to server and register the locations with data store

Creating Tools – Best practices

- **Pre-compute input data**
 - Add attribute indexes, add spatial indexes (i.e. shapefiles)
- **Use a map layer as project data**
- **Use in_memory workspace**
- **Limit processing extent**
 - Set extent environments
- **Avoid project on the fly**

Tools that use the Topology Engine

- For server services that do large data processing, one process per node (different machines)
 - Reason: Topo engine checks for available memory for the system and takes 60% or more. Since all cores share the same RAM, a large percentage of available RAM is consumed
- Tools include:
 - Overlay tools (Intersect, Union, etc.)
 - Dissolve
 - Feature to Line, Feature to Polygon, etc.

Function Tools & 3rd Party Python packages

- http://resources.arcgis.com/en/help/main/10.1/#/Deploying_custom_NET_and_C_tools/00570000008p000000/
- http://resources.arcgis.com/en/help/main/10.1/#/Deploying_custom_Java_function_tools/00570000008n000000/
- http://resources.arcgis.com/en/help/main/10.1/#/Deploying_custom_Python_packages_for_ArcGIS_Server/00570000008v000000/

Useful help topics

- Quick tour of authoring and sharing geoprocessing services – <http://esriurl.com/gpSrvQuick>
- Javascript Help –
- <http://developers.arcgis.com/en/javascript/>