Creating Geoprocessing Services with Python Script Tools

Andrew Ortego

1. Data Preparation
2. Importing Modules
3. Tool Validation
4. Tips and Tricks

Data Preparation (1/2)

- All input-data is consolidated and published to the server while publishing a service
  - After publishing, it is referenced in memory (watch your server heap size)
- Python is data-greedy and will consolidate data even if not used in the script
  - Consolidation copies files and geodataset's (.gdb, raster, and .lyrx or .mapx references)
- This includes all data found in all string's in the Python script
  - Absolute and relative path's
  - Layer's in the Table of Contents/Content pane
- Build path's with os module

os.path.join(directory, file_name)

Data Preparation (2/2)

- Avoid absolute and relative path's whenever possible, and use Parameters instead
  - Familiarize yourself with unsupported parameter-type's while building the script tool
- Use parameters to reduce the number of string's
  - Also forces client to use *existing* data; avoids issues with missing data

arcpy.GetParameterAsText()

1. Data Preparation
2. Importing Modules
3. Tool Validation
4. Tips and Tricks

Importing Modules

- Modules are consolidated, and searched for in the following locations:
  - Script's root folder
  - PATH environment variable
  - PYTHONPATH environment variable
- Use Python to add additional paths if modules are located elsewhere

```
sys.path.append(r"C:/module/directory")
```

- Third-party modules are not consolidated and must be located on the server
- Verify that the Python modules used to create the script can be used on the server
  - 32-bit vs. 64-bit issues could be a potential problem

1. Data Preparation
2. Importing Modules
3. Tool Validation
4. Tips and Tricks

Tool Validation

- Is published along with Web Tool, and executed on SubmitJob
  - Set up as normal in the Script Tool prior to running
- ToolValidator class will run on the server-side
  - Client will see output from ToolValidator class
- Recall the supported input-types for Web Tools:
  - Feature set/layer, record set, table view, file, double, long, string, date, linear units
- Consider using a Geoprocessing Package to maintain ToolValidator logic

param[0].filter.list = [“option1”, “option2”]

1. Data Preparation
2. Importing Modules
3. Tool Validation
4. Tips and Tricks

Tips and Tricks

- Use layers for project data, not paths
  - Get performance boost by opening one dataset instead of multiple datasets/paths
- Use server-side (local) data whenever possible
  - Data already found in the Data Store will not be consolidated when publishing
  - Avoids “lag” issues by reducing use of network-data
- Avoid different coordinate systems
- Write intermediate data to memory

  os.path.join(“in_memory”, “output”)
- Use REST endpoint to automate testing with Python

  Use requests and json module’s
- Use arcpy.ImportToolbox() to access Web Tool with arcpy

Helpful Links

- **“What Is a Web Tool”: all documentation on Web Tool’s**
- **“Authoring geoprocessing tasks with Python scripts”: for building script tool’s**
- **Read about Python’s os module and sys module**
  - **Python Module of the Week**
  - **Official documentation**
  - **“Python for Kids” as an intro to Python**
- **Check with MyEsri to see if you have a Support License**


esri
THE
SCIENCE
OF
WHERE