DEVELOPER SUMMIT

March 10–13

Creating Geoprocessing Services with Python Script Tools

Kevin Hibma

March 12, 2015 - 10:30am

- Help topic http://esriurl.com/gpSrvPy

- Script tools as a GP Service usually:
  - Have some project data (data the server has access to)
  - User supplied inputs (small datasets, strings, longs, etc)
- Because of this the authoring pattern is usually straight forward

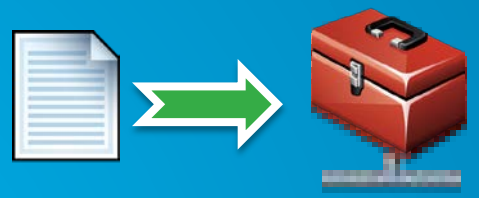
Topics

- Understanding and writing good input and output data paths:
  - + (plus) is for math, not paths
  - *scratchFolder* and *scratchGDB*; variables you've been missing
- #1 Performance tip:
  - Do this (layers), not that (make feature layer)
- Validator code:
  - When, where and huh?
- 3rd party modules and packages
  - Folder variables can be used to do what?
- Messages:
  - AddMessage, AddWarning, AddError: bend them to your will
- The nuts and bolts of it…

What happens during publishing?

- Script is scanned
- Data identified

- If <u>in</u> datastore, update path (if necessary)



- **If <u>not</u> in datastore, consolidate data, update path**

Data Paths (inputs)

- Build up good paths – OS package is your best friend
  - Inputs / project data can be absolute or relative:
  - `pts = os.path.join(sys.path[0], “Tooldata”, “points.shp”)`
  - `pts = “c:\\projects\analysis\tooldata\points.shp”`
- Don’t build paths with “+” operator!
  - `pts = r“c:\projects\analysis” + “\tooldata\points.shp”`
- `“c:\\data”` or `r“c:\data”` or `“c:/data”`

Data Paths (outputs)

- Keep using os.path
  - Write outputs (intermediates and terminal outputs) with a combination of scratch environments and os.path
  -
```
output = os.path.join(arcpy.env.scratchGDB, “outPoints”)
```
  - (outputs to an absolute path will be fixed in the published script to the scratch environment)
  - `arcpy.env.scratchFolder`
  - `arcpy.env.scratchGDB`
  - `arcpy.env.workspace`

Performance!

- User layers!
  - Name match them in your script to the layer in the ToC
- Write output to in_memory
  - os.path.join("in_memory", "output") or "in_memory\\output"
- http://server.arcgis.com/en/server/latest/publish-services/windows/performance-tips-for-geoprocessing-services.htm
  - Local data (not network data)
  - Preprocess data (don't run tools needlessly)
  - Attribute indexes / Spatial indexes
  - Avoid different coordinate systems

Validator Code

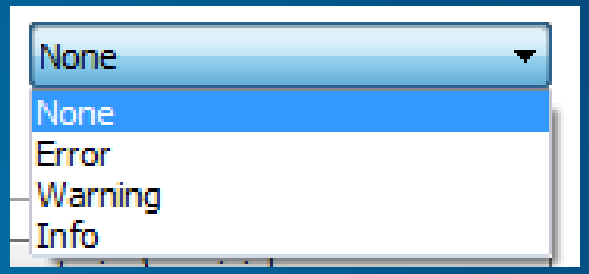
- Validator code is fired at service execution, not before on the client.
  - Where should you “validate”?
- Create a variable pointing to a folder
  - Folder will be copied or referenced depending on data store setup

3rd Party Modules (folder variables)

- 3rd party packages (SciPy) for example are not consolidated/moved to the server. You must ensure they have been installed on the Server

Messages

- arcpy.AddError(“Oh-oh!”)
- arcpy.AddWarning(“Hello?”)
- arcpy.AddMessage(“Hello!”)
- Remember: client needs to get and display these. ArcMap ‘just does it’. WebApps need code.



When we parse the script

- Scripts that have been parsed will not be parsed again
- What does that mean exactly?

Rate This Session

www.esri.com/RateMyDevSummitSession