AUTOMATING GIS WITH PYTHON
Spring NEARC

Kristina Grace
GIS Analyst
May 22, 2012
Topics

• Broadband Availability and MassBroadband 123 maps
• Create and document tools made from Python scripts
• Script examples
  – error handling
  – tool parameters
  – check if path exists
  – where clause
  – calendar object
  – ArcToolbox: Conversion and Data Management
  – turn fields on and off in tables views
• Online build phase map
Broadband Availability Map

http://mapping.massbroadband.org

Also incorporated at www.broadbandmap.gov
Create And Document Tools

1. Create script
   • Python IDLE
   • use TEMPLATE.py

2. Add scripts to ArcToolboxes
   • add to tool or toolbox

3. Add description to tools
   • edit metadata
     – keywords
     – summary
     – parameter description
Error Handling

- processes inside try
- error message formatting inside except

```python
msgs = 'GP ERRORS:\n' + arcpy.GetMessages(2)
```
Delete Files And Feature Classes

```python
arcpy.Delete_management(data_element)
arcpy.Delete_management(CWR3LogTbl)
```

```python
arcpy.Delete_management(CWR3LogTblView)
arcpy.Delete_management(workGDBParam + '\\' + CWR3FiberTbl)
arcpy.Delete_management(workGDBParam + '\\' + CWR3CAITbI)
arcpy.Delete_management(workGDBParam + '\\' + CWR3LogTbl)
print 'intermediate tables deleted'
```
Check If Path Exists

- Path supported by ArcGIS:
  
  ```python
  arcpy.Exists("\"")
  ```

- Path not supported by ArcGIS:
  
  ```python
  os.path.exists("\"")
  ```

### Example Code

```python
if arcpy.Exists(dataset):
    # do something

if arcpy.Exists(FC):
    arcpy.Delete_management(FC)
```

---

```
#Export CAI to CAI_today's date table in
if arcpy.Exists(workGDBParam + "\"" + CWR):  
    arcpy.Delete_management(workGDBParam + 
    arcpy.TableToTable_conversion(MB123Networ)  
print 'imported the fiber and CAI tables'
```
Tool Parameters

Param = arcpy.GetParameterAsText(0)

dateChangeParam = arcpy.GetParameterAsText(3)
### Tool Parameters

**Script Fiber and CAI to DBF v1.1 Properties**

#### Parameters Tab

<table>
<thead>
<tr>
<th>Display Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>master file geodata...</td>
<td>Workspace or Feature Dataset</td>
</tr>
<tr>
<td>work GDB</td>
<td>Workspace</td>
</tr>
<tr>
<td>original change log t...</td>
<td>Table</td>
</tr>
<tr>
<td>get logs starting with...</td>
<td>Date</td>
</tr>
<tr>
<td>Output folder for D...</td>
<td>Folder</td>
</tr>
</tbody>
</table>

---

[Image of the tool parameters window]
Extract Data For Cost Analysis

**Script Fiber and CAI to DBF v1.1**

- **Master file geodatabase**
  - N:/Data/MassBroadband123.gdb/Net

- **Work GDB**
  - N:/Data/Cornell/CWR3Work.gdb

- **Original change log table**
  - N:/Data/MassBroadband123.gdb/GIS

- **Get logs starting with the following date**
  - 5/22/2012

- **Output folder for DBF files**
  - N:/Data/Cornell/MB123_Fiber_CAI_DL

---

This script joins GIS log stas by Cornell Robinson III.

To filter the log table so only '5/15/2012', '05/15/2012', 'M

The tool creates two new DB files.

The following fields from the ChangeDate.
Export Feature Classes

Extract Data For Cost Analysis Script

```python
arcpy.FeatureClassToFeatureClass_conversion(FC, out_path, out_name)

arcpy.FeatureClassToFeatureClass_conversion(MB123NetworkDSPParam + '\\CAI', workGDBParam, CWR3CAIFC)
```
Where Clause

- String variable
- Field name needs to be surrounded by quotes
- Used in
  - arcpy.SelectLayerByAttribute_management()
  - arcpy.MakeTableView_management()

```python
whereClause = ' "fieldName" >= ' + variable

whereClause = ' "DateChange" >= ' + dateChange
```

```python
if int(dateList[1]) < 10:

else:
    day = str(dateList[1])
    dateChange = 'date "' + year + '"' + month + '"' + day + '"
    arcpy.AddMessage('dateChange: ' + dateChange)
    whereClause = ' "DateChange" >= ' + dateChange # query that wil
Extract Data For Cost Analysis Script

where_clause = "InClInBuild <> 'N' and InClInBuild <> 'W"

dateChange = 'date \" + year + '-'+ month + '-'+ day + "\"
whereClause = ' "DateChange" >= ' + dateChange
Turn Fields On And Off

Extract Data For Cost Analysis Script

```python
fieldinfo.addField (field_name, new_field_name, visible, split_rule)
```

```python
fieldinfo.addField(field.name, field.name, 'HIDDEN', '')
```

```python
if field.name == 'DateReques':
    fieldinfo.addField(field.name, field.name, 'HIDDEN', '')
elif field.name == 'DateChange':
    fieldinfo.addField(field.name, field.name, 'VISIBLE', '')
```
Join Tables And Feature Classes

Extract Data For Cost Analysis Script

```
import arcpy

# Add an join to the log table with a Derived ID field
arcpy.AddJoin_management(in_layer_or_view, in_field, join_table, join_field, {join_type})

arcpy.AddJoin_management(CWR3LogTblView, "ID", workGDBParam + "\" + CWR3CAITbl, "CAI_ID", 'KEEP_COMMON')
```

```python
arcpy.RemoveJoin_management(CWR3LogTblView, CWR3CAITbl)
pprint 'join removed'

# Inner-join the CAI table to the log table
arcpy.AddJoin_management(CWR3LogTblView, "ID", workGDBParam

print 'CAI join done'
```
Add Fields To Feature Classes

Extract Data For Cost Analysis Script

```python
arcpy.AddField_management(workGDBParam + r'\' + CWR3LogTblJoinFiber, "Cost", "DOUBLE")
```

```
print 'fields added to the log and fiber table'
```
Export file gdb tables to DBF

Extract Data For Cost Analysis Script

```python
arcpy.CopyRows_management(in_rows, out_table)
```

```python
arcpy.CopyRows_management(table, file.dbf)
```
Data Archiving Tool

- Set up archive file gdb ahead of time
- Set today’s date
- Add today’s date to the output feature class when using Feature Class to Feature Class conversion tool

```python
now = datetime.datetime.now()  # Set today's date

'CAI_' + now.strftime('%Y%m%d'))
```
Update Maps Tool

archive GDB
N:\Data\MassBroadband123\SAVED_

input map table
N:\Tools\MBI_MapExport_Settings.gdb

map selection query
"UPDATE_" = 'Y'
Replace Data Source

Update Maps Script

 layer_name.replaceDataSource(gdb, "FILEGDB_WORKSPACE", new source feature class)

CAILayer.replaceDataSource(archiveGDBParam, "FILEGDB_WORKSPACE", 'CAI_' + now.strftime('%Y%m%d'))
Update Map Elements

Element Name

MapDate

ARRA Funded Project

Map Date: 26 April 2012
mapDate = arcpy.mapping.ListLayoutElements(map_document, "TEXT_ELEMENT", "MapDate")[0]

mapDate = arcpy.mapping.ListLayoutElements(mxd, "TEXT_ELEMENT", "MapDate")[0]

if monthParam == '':
    newMapDate = 'Map Date: ' + now.strftime('%d') + ' ' + now.
calendar Object

Update Maps Script

- methods
  - month_name[]
  - day_name[]

```python
str(calendar.month_name[int(monthParam)])
calendar.month_name[int(variable)]
```

```python
ate: ' + now.strftime('%d') + ' ' + now.strftime('%B') + ' ' + now.strftime('%Y')
ate: ' + dayParam + ' ' + str(calendar.month_name[int(monthParam)])
te ' ate changed'
```
Multi-Map Export Tool

Input Map Table
N:\Tools\MBI_MapExport_Settings.gc

Map Selection Query (optional)

Export Format (optional)
- PDF
- JPG
- EPS

Export Resolution (optional)
- 96 dpi
- 300 dpi
- 600 dpi
- 1200 dpi
MassBroadband 123 Network Progress Tracker

Build Phase 1

Operational Date: Expected June 2013

Poles Licensed: 110 of 10,148 poles (1%)

Fiber Miles Deployed: 57 of 300 miles (19%)
Online Build Phase Map

MassBroadband 123 Network Progress Tracker

Build Phase 1
Build Phase Tracker

Build Phase 1

Operational Date: Expected June 2013

Poles Licensed: 110 of 10,148 poles (1%)

Fiber Miles Deployed: 57 of 300 miles (19%)

CAIs Connected: 0 of 466 CAIs (0%)
Progress Statistics Tool

**Build Phase Statistics**

- **feature dataset**
  - N:\Data\MassBroadband123\gdb\Net

- **output folder**
  - N:\Data\MassBroadband123\MBI_We

- **map document**
  - N:\Workspaces\IN_PROGRESS\Network_Tra

- **data frame**
  - Layers

- **Run statistics on the following feature classes:**
  - CAI
  - Poles
  - Fiber

**WARNING:** The script file with today's date:

This script creates MB123 network data.

It calculates the number of CAI and miles of fiber (planned and under construction) using Spatial Join and other methods.
Statistics Analysis

Online Build Phase Tracker Script

```python
arcpy.Statistics_analysis(input_data, output_table, [[field, "COUNT"]], case_field)
```

```python
arcpy.Statistics_analysis(asset, buildPhaseSJ + "_stats", [[assetPrimaryKey, "COUNT"]], BuildPhaseField)
```

```python
arcpy.AddMessage(buildPhaseSJ + "_stats" + " Calculated statistics for " + asset)```
## Statistics Analysis Output

### Online Build Phase Tracker Script

<table>
<thead>
<tr>
<th>G4SPhase</th>
<th>FREQUENCY</th>
<th>SUM_{LenMiles}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build Phase 1</td>
<td>1078</td>
<td>309.6468340</td>
</tr>
<tr>
<td>Build Phase 2</td>
<td>407</td>
<td>179.9728937</td>
</tr>
<tr>
<td>Build Phase 3</td>
<td>416</td>
<td>141.7151424</td>
</tr>
<tr>
<td>Build Phase 5</td>
<td>266</td>
<td>101.8440182</td>
</tr>
<tr>
<td>Build Phase 6</td>
<td>450</td>
<td>215.4502940</td>
</tr>
<tr>
<td>Build Phase 7</td>
<td>288</td>
<td>122.1585009</td>
</tr>
</tbody>
</table>
rows = arcpy.SearchCursor(table_name)

rows = arcpy.SearchCursor(buildPhaseSJ + "_stats")

buildPhaseFieldStats = arcpy.ListFields(buildPhaseSJ + "_stats")
statsField = arcpy.ListFields(buildPhaseSJ + "_stats")

arcpy.AddMessage("list field variables defined")
getValue of Row

Online Build Phase Tracker Script

```python
for row in rows:
    if row.getValue(myField.name) == "Build Phase 1":
        bp1 = row.getValue(statsField.name)
    elif row.getValue(buildPhaseFieldStats.name) == "Build Phase 2":
        bp2 = row.getValue(statsField.name)
    elif row.getValue(buildPhaseFieldStats.name) == "Build Phase 3":
        bp3 = row.getValue(statsField.name)
```

row.getValue(field.name)
Replace Text in File

Online Build Phase Tracker Script

```python
fileread = open(JSFileTEMPLATE, 'r')
filetext = fileread.read()
filetext = filetext.replace("BP1CCT",str(bp5))
filetext = filetext.replace("BP2CCT",str(bp2))
filetext = filetext.replace("BP3CCT",str(bp3))
filetext = filetext.replace("BP5CCT",str(bp5))
```

```python
text.replace(placeholder,new_number)
```
Review

• Met Python
• Learned how to create and document tools from Python scripts
• Saw Python script examples, such as defining variables and parameters, handling errors, exporting datasets, adding fields, getting field values, summarizing data and using extra modules such as calendar and os
• Saw Output example online
Resources

www.massbroadband.org
www.broadbandmap.gov
www.geospatialtraining.com
http://docs.python.org/library

E-mail: grace@masstech.org