

Useful ArcPy Commands and Code Snippets

Nick Eubank

May 16, 2014

Individual Functions

Data Manipulation

- Create Geodatabase: `CreatePersonalGDB_management()`
- Delete things: `Delete_management()`
- Import GPS Points: `MakeXYEventLayer_management()`
Creates a Layer – frequently need to then copy into featureclass for storage
- Copy Layer to FeatureClass / Move Featureclass: `CopyFeatures_management()`
- Copy Raster to FeatureClass: `CopyRaster_management()`
- Export Attribute Table: `ExportXYv_stats()`
Note: Will not label columns, and will include some leading columns, so be careful with use!
- Create a layer from featureclass: `MakeFeatureLayer_management()`
Some tools only act on layers, and selections only work for layers

Projections and Coordinate Systems

- Define Projection or Coordinate System: `DefineProjection_management()`
This is for SETTING the coordinate system, not RE-PROJECTING into NEW coordinate system
- Re-Project to New Projection or Coordinate System: `Project_management()`

Working with Fields

- Add a field (prior to calculating): `AddField_management()`
- Calculate Field: `CalculateField_management()`
- Delete Field: `DeleteField_management()`

Selections

Selections can be applied to either one of the two types of layers in ArcPy (see Tutorial, Section 8 for a discussion of layers). *Selections can only be applied to layers, not feature classes!*

- Create a Feature Layer from featureclass: `MakeFeatureLayer_management()`
- Manipulate Selections in Layer: `SelectLayerByAttribute_management()`
Note this is a multifaceted function – the second argument in function allows you to specify whether you’re creating new selection, clearing selection, subsetting, etc.
- Manipulate Selections by Location: `SelectLayerByLocation_management()`

Cartography

You can also use ArcPy to manipulate Map Documents (ending in `.mxd`). This is useful for cartography, and has it’s own set of tools.

When working with cartography, one often also works with selections. However, this is an area I don’t know well, so this is admittedly thin!

- Create Map Document Variable: `mapping.MapDocument(“file.mxd”)`
e.g. `mxd = arcpy.mapping.MapDocument(“file.mxd”)`
- Set the extent of the Map Document (i.e. zoom) to extent of selected elements of a layer (called `myLayer`):

```
lyrExtent=myLayer.getSelectedExtent()  
df.extent=lyrExtent
```

Idioms and Code Snippets

Create New Database, Delete Old if Present, Report Back

```
# Target geodatabase name  
Output="NewDatabaseName.gdb"  
# Home Folder  
HomeFolder ="Z:/Documents/VAM"  
  
try:  
    arcpy.Delete_management(Output,HomeFolder)  
    print "Deleted Old Database"  
except:  
    # Output.gdb did not exist, pass  
    print "Did not delete"  
    pass  
arcpy.CreateFileGDB_management(HomeFolder, Output, "CURRENT")  
print "Created New Database"
```

Cartography: Zoom to different constituencies and print PDFs of each constituency

```
mxd = arcpy.mapping.MapDocument("pc_maps_forexport.mxd")
mxd.author="Nick Eubank"
#mxd.save()

# Now I want to draw out the first item in the current data frame --
# so call for list then subscript for item "0" Of course I only have one
# data frame, but this precludes the possibility I'll get back a list
# instead of a single item.

df=arcpy.mapping.ListDataFrames(mxd, "Layers")[0]

# Call list of layers. If you add a [0] to the end, you get a single
# object. I just pull out the objects in the next few lines. My first
# layer (lyrList[0]) is the layer of PC locations. The second (lyrList[1])
# is a constituency map.
lyrList=arcpy.mapping.ListLayers(mxd)
print lyrList
pcsLayer=lyrList[0]
constLayer=lyrList[1]

# List of target constituencies

consts=["1206", "1207", "1303", "2105", "2201", "2301", "2304",
"2305", "2408", "2505", "3104", "3304", "3305", "4102"]

for item in consts:
    selectorString="[const_no]= "
    print item
    arcpy.SelectLayerByAttribute_management(pcsLayer, "NEW_SELECTION",
selectorString+item)
    print "Selections Made"
    lyrExtent=pcsLayer.getSelectedExtent()
    print "LyrExtent Run"
    df.extent=lyrExtent
    print "DF extent set"
    fileNameA="Z:/Dropbox/Elections_Sierra_Leone/pc_maps/map_"
    fileNameB=".pdf"
    # Now clear selections so cleaner looking
    arcpy.SelectLayerByAttribute_management(pcsLayer, "CLEAR_SELECTION")
    arcpy.mapping.ExportToPDF(mxd, fileNameA + item + fileNameB)
    print "Exported"

print "Total Success!"
```