



Esri International Developer Summit
Palm Springs, CA

Using Python with ArcGIS Runtime Desktop SDKs

Mike Branscomb and Elise Acheson

Using Python with ArcGIS Runtime desktop SDKs

- Use Python with ArcGIS Runtime for **geoprocessing**:
 - automating GIS tasks
 - modeling and analysis
- We will focus on **integrating** Python scripts into ArcGIS Runtime apps
 - Rather than on writing scripts
- We assume:
 - A working knowledge of **Python**
 - Some familiarity with **ArcGIS Runtime**

Python in ArcGIS Runtime

- Uses Runtime Local Server (Java, Qt, WPF, .NET)
 - Local (offline) equivalent of geoprocessing services in ArcGIS Server
 - same concepts apply
 - Subset of geoprocessing tools are supported in Local Server
 - anything dependent on MXD not supported (e.g. arcpy.mapping)
 - see list of supported geoprocessing tools in SDK doc
1. **Write** a Python script to consume in your Runtime app
 2. **Package** the script into a geoprocessing package (gpk)
 3. **Consume** the package/script in your Runtime app

DEMO

Writing a script for data management

Mike Branscomb

```
ListGeodatabaseFeatureClasses.py - C:\WORKEVENTS\DevSummit2014_
File Edit Format Run Options Windows Help
# Import necessary modules
import arcpy
import os

# Set variables from input parameters.
workspace = arcpy.GetParameterAsText(0)

# Set the current workspace
arcpy.env.workspace = str(workspace)

# Create the output table.
outTableFc = arcpy.CreateTable_management('in_memory', 'featureclasses')

# Set the field names and definitions for the output table.
arcpy.AddField_management(outTableFc, "name", "TEXT", "", "", 256)

# Use the ListFeatureClasses function to return a list of FCs
featureclasses = arcpy.ListFeatureClasses()

# Create an insert cursor to insert rows into the output table.
cur = arcpy.da.InsertCursor(outTableFc, ('name'))

# For each featureclass insert the name into a new row in the table.
for featureclass in featureclasses:
    name = featureclass
    row = [name]
    cur.insertRow(row)
del cur

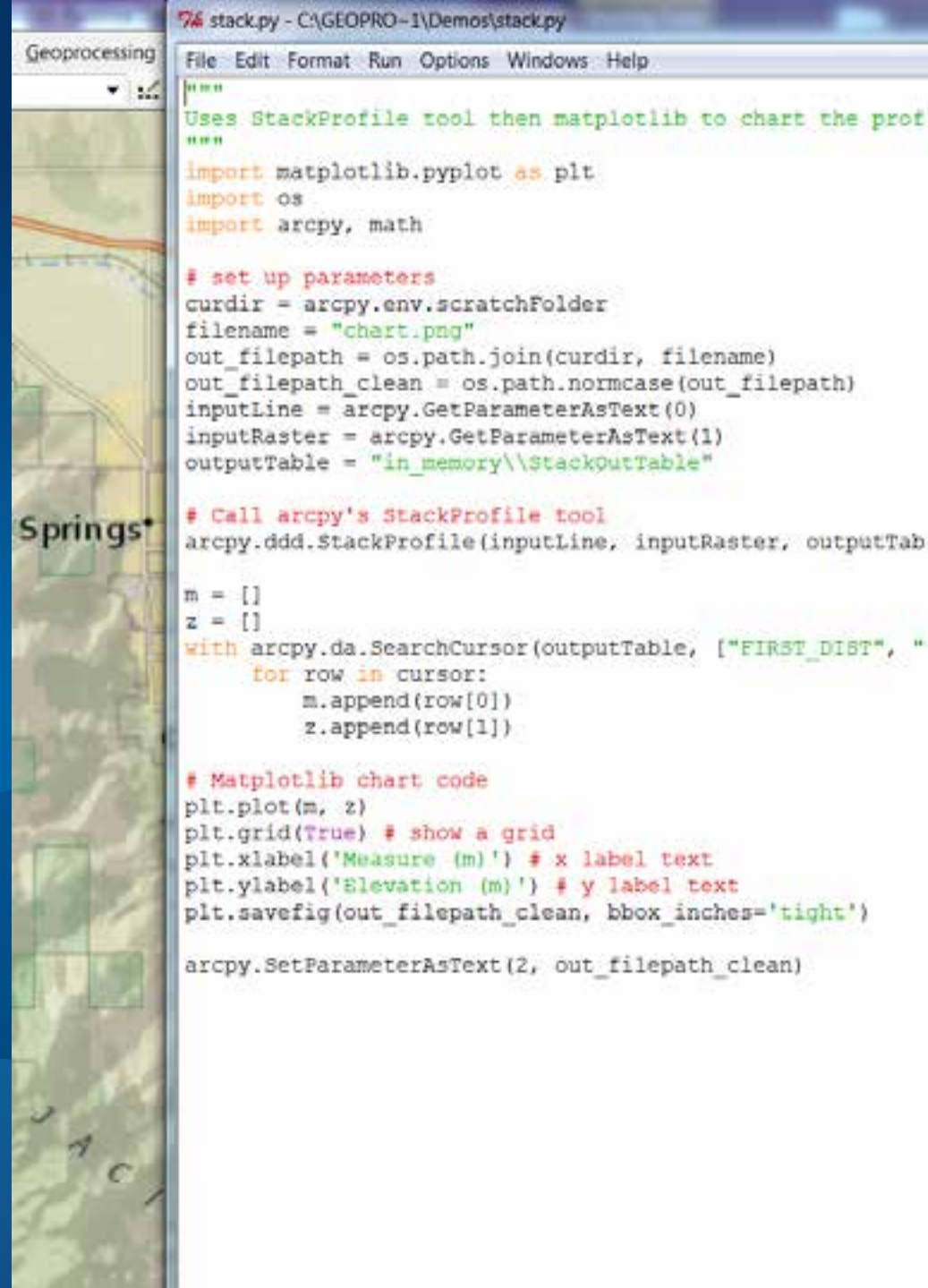
# Set the output parameter:
arcpy.SetParameterAsText(1, outTableFc)

Ln: 30 Col: 26
```

DEMO

Writing a script for analysis

Elise Acheson



```
stack.py - C:\GEOPRO-1\Demos\stack.py
File Edit Format Run Options Windows Help
"""
Uses StackProfile tool then matplotlib to chart the prof
"""
import matplotlib.pyplot as plt
import os
import arcpy, math

# set up parameters
curdir = arcpy.env.scratchFolder
filename = "chart.png"
out_filepath = os.path.join(curdir, filename)
out_filepath_clean = os.path.normcase(out_filepath)
inputLine = arcpy.GetParameterAsText(0)
inputRaster = arcpy.GetParameterAsText(1)
outputTable = "in_memory\\StackOutTable"

# Call arcpy's StackProfile tool
arcpy.ddd.StackProfile(inputLine, inputRaster, outputTab

m = []
z = []
with arcpy.da.SearchCursor(outputTable, ["FIRST_DIST", "
    for row in cursor:
        m.append(row[0])
        z.append(row[1])

# Matplotlib chart code
plt.plot(m, z)
plt.grid(True) # show a grid
plt.xlabel('Measure (m)') # x label text
plt.ylabel('Elevation (m)') # y label text
plt.savefig(out_filepath_clean, bbox_inches='tight')

arcpy.SetParameterAsText(2, out_filepath_clean)
```

Demo Summary

- **Lots of Python samples on ArcGIS Resource Center**
 - Each function has an example
- **Write Python script in editor of your choice**
- **Import Python script to Toolbox**
- **Define parameter names and types**
 - Must match order of inputs/output in script
- **Run tool in Desktop**
 - Needs to have valid inputs and outputs
 - Data may be incorporated into GPK – use small representative datasets
- **Share Result as GPK**
- **Tool will be validated by ArcGIS analyzers**
 - But cannot validate entire python script

Supported input and output data types

- **Supported Input Data Types**

- Feature Set
- Record Set
- Standard Types (Long, Double, Boolean, Date, and String)
- Multivalue

- **Supported Output Data Types**

- Feature Set (Feature Classes)
- Feature Layer and Raster Layers (Map Services)
- Table
- Standard Types (Long, Double, Boolean, Date, and String)
- File
- Multivalue

Writing scripts – Messages

- Can be useful for debugging
- `arcpy.AddError("Uh-oh!")`
- `arcpy.AddWarning("Hello!")`
- `arcpy.AddMessage("Hello?")`
- Remember: you need to programmatically iterate and display these
 - ArcMap displays them in the Results Window

What happens during packaging?



- Script is scanned
- Data identified



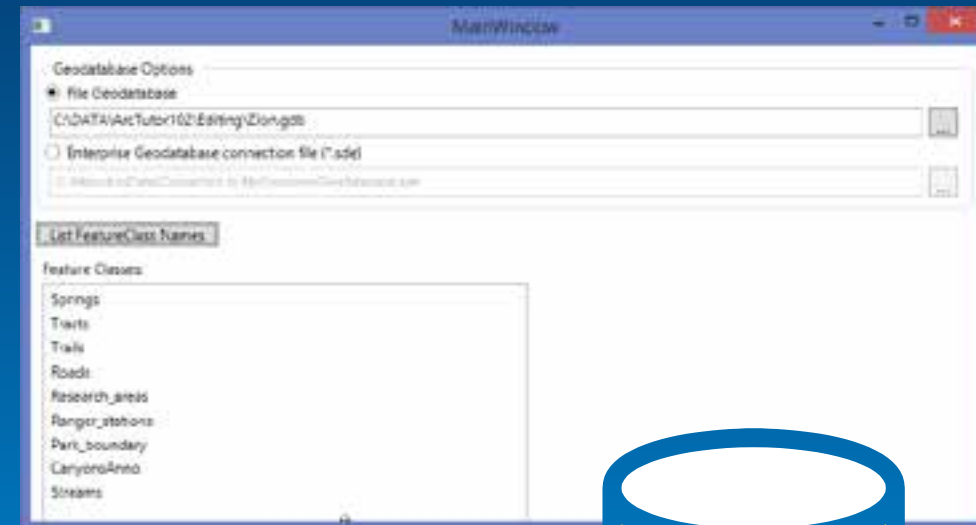
- Packaging consolidates data, updates paths



DEMO

Consume the data management script

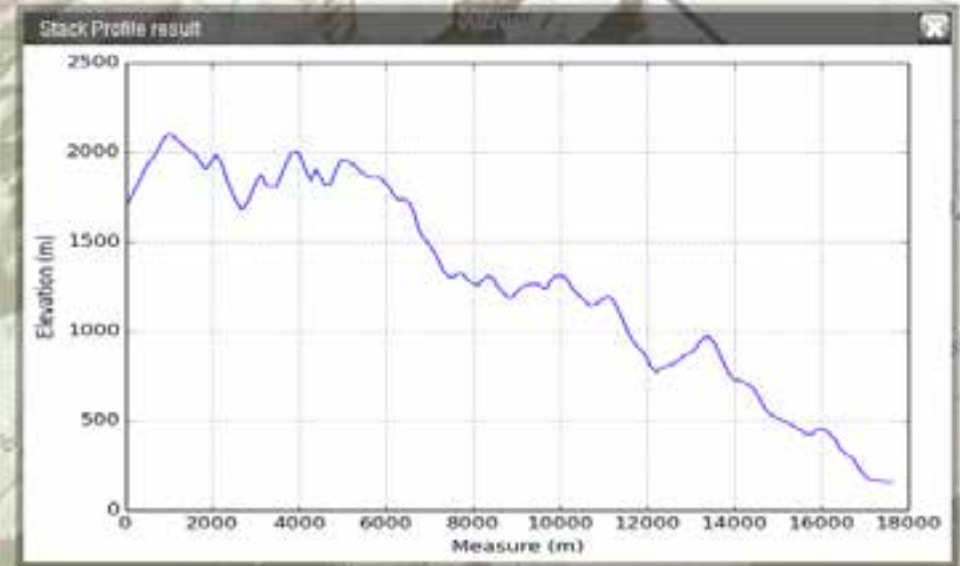
Mike Branscomb



DEMO

Consume the analysis script

Mike Branscomb / Elise Acheson



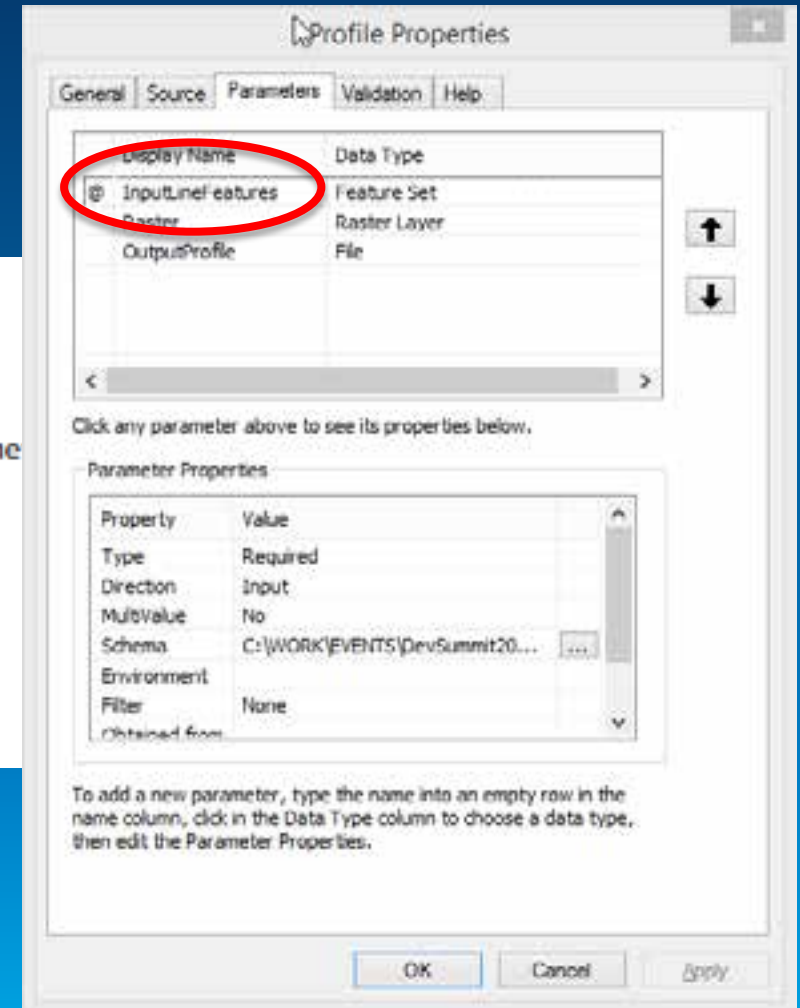
Consuming the script - summary

- Note parameter names defined for imported script
- Must use same parameter names in code

```
GPInputParameter gpInputParameter = new GPInputParameter();
gpInputParameter.GPParameters.Add(new GPFeatureRecordSetLayer
("InputLineFeatures", inputWorldMillerCylindrical));

GPExecuteResult results = await geoprocessorTask.ExecuteAsync(gpInputParameter);

GPDataFile gpDataFile = results.OutParameters[0] as GPDataFile;
BitmapImage bitmapImage = new BitmapImage();
bitmapImage.BeginInit();
bitmapImage.UriSource = gpDataFile.Uri;
bitmapImage.EndInit();
ChartImage.Source = bitmapImage;
```

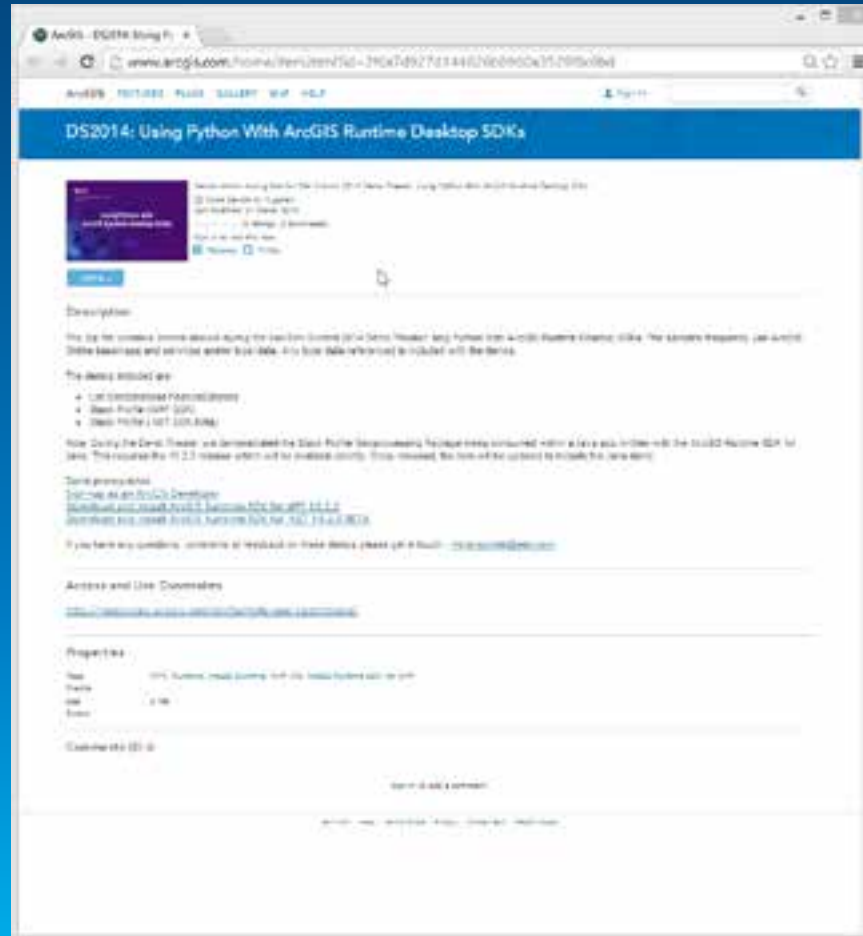


More Resources

- Python ArcGIS homepage on ArcGIS Resources:
 - <http://pro.arcgis.com/en/analysis/python/>
- Code snippets from dev team
 - <http://arcpy.wordpress.com/>
- Blog on choosing a Python IDE
 - <http://blogs.esri.com/esri/arcgis/2013/06/24/choosing-the-right-python-integrated-development-environment/>
- matplotlib site
 - <http://matplotlib.org/>
- Python documentation
 - <http://docs.python.org/>
- Dev Summit Tech Session
 - Creating and Working with Geoprocessing Services

Demos

<http://esriurl.com/7576>





Understanding our world.