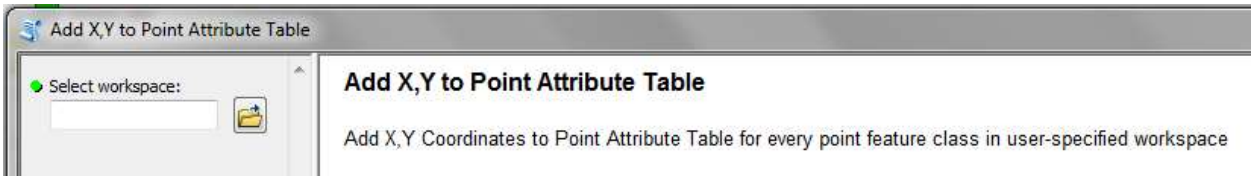
Session 4: ArcGIS Script Tools

In this session we will cover how to create script tools in your personal toolbox**.**

Script tools allow you to have your Python geoprocessing script accessible to others and also allow you to filter and control how the script tool appears to the user.

AddXY To Many Point Feature Classes

Let's take a simple example…a script tool that will add the x,y coordinate of every point feature class in a user-specified workspace.

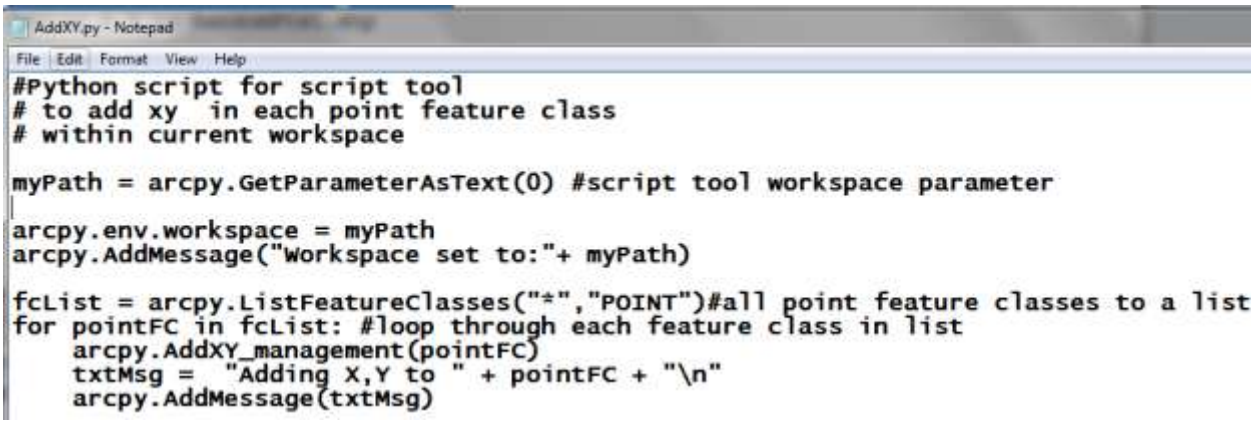


When you build your script tool, you will specify the script tool:

label ("Add X,Y to Point Attribute Table")

Description ("Add X,Y Coordinates to Point Attribute Table for every …")
Parameter display name ("Select workspace:")
Parameter data type (workspace)

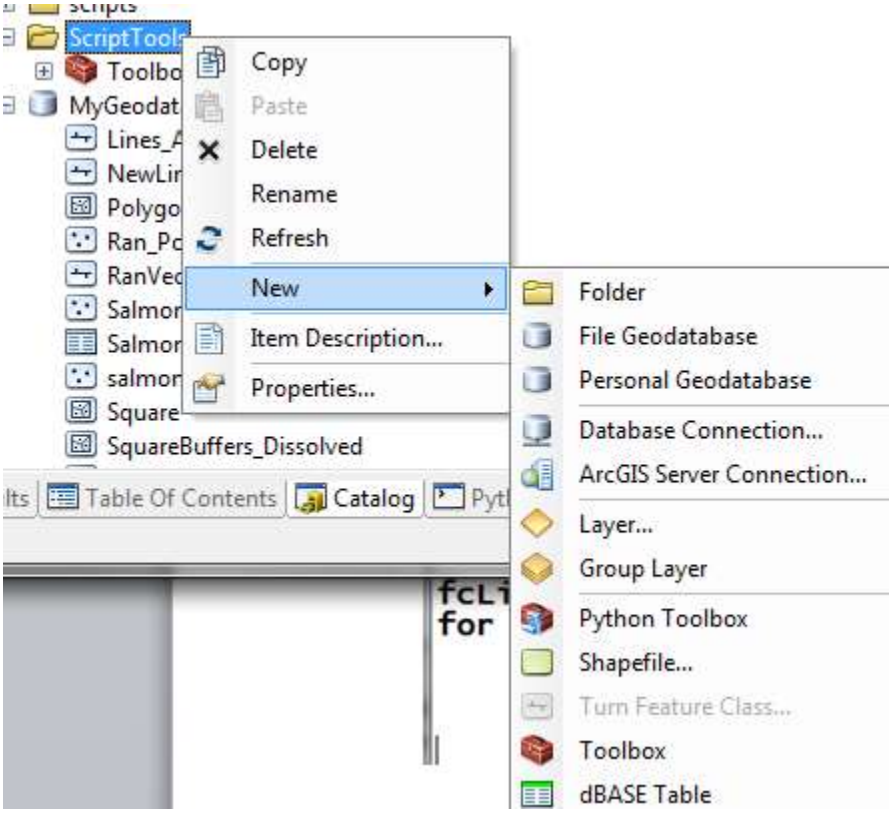
Look at the following Python script in the Session4_ScriptTools\Python folder



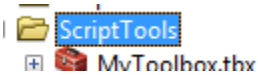
arcpy.GetParameterAsText(0) is our first and only script tool parameter which takes user-specified input text and stores this text in the variable myPath

arcpy.AddMessage() allows us to send messages as the script tool executes

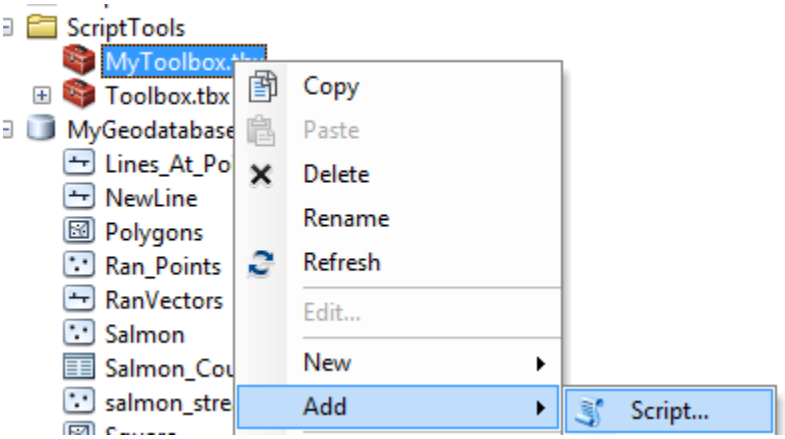
Create a personal toolbox using your Arcmap Catalog window…

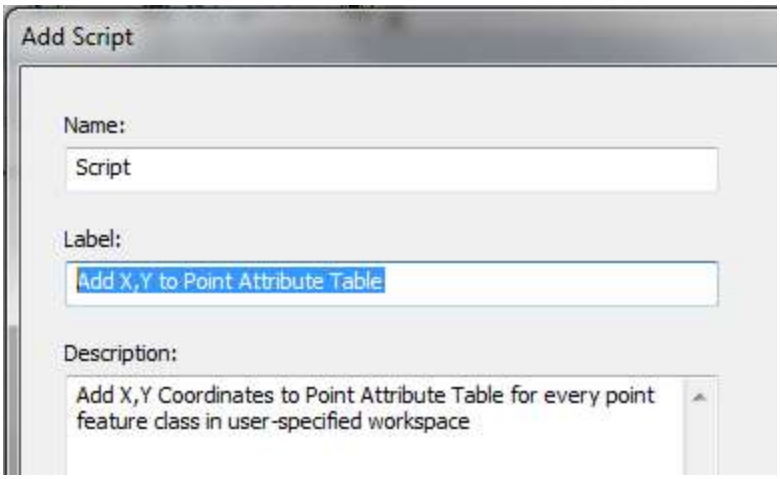


Create a Toolbox (***not*** Python Toolbox)

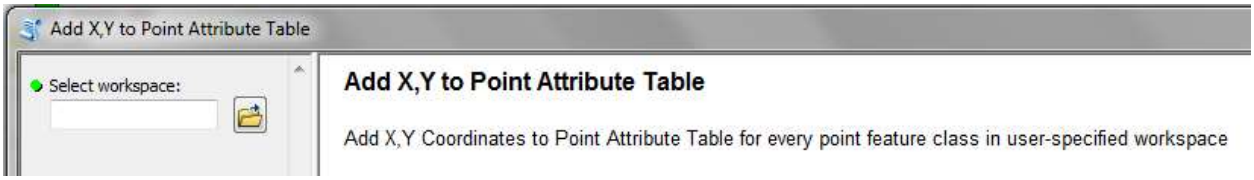


Next, right mouse-click on your personal toolbox and Add your Python script…

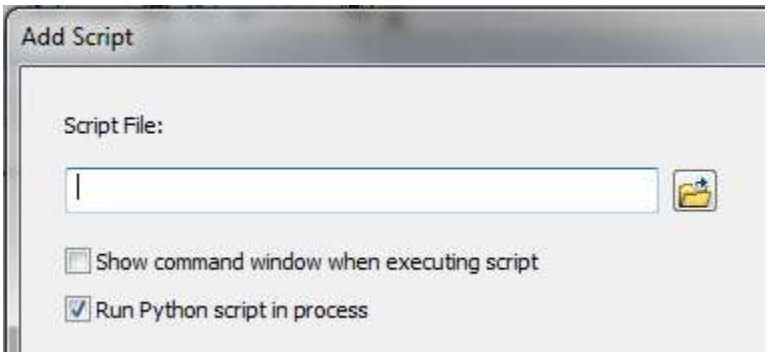


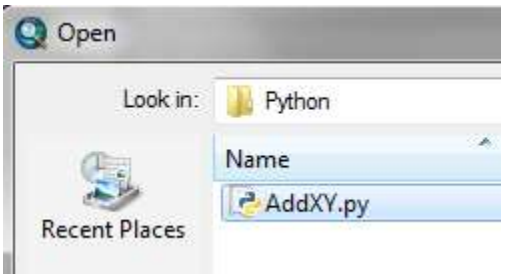


The label will be what you see the script tool titled as and the description will be in the dialog…

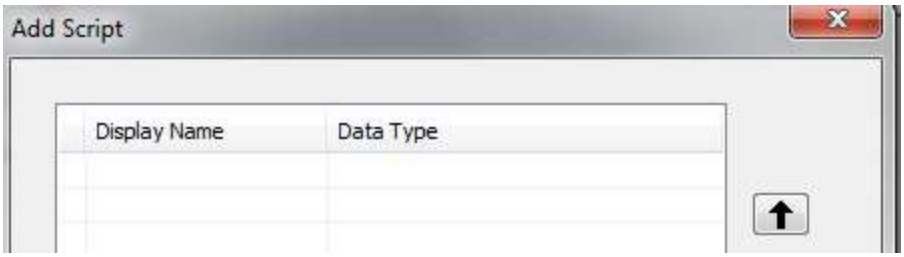


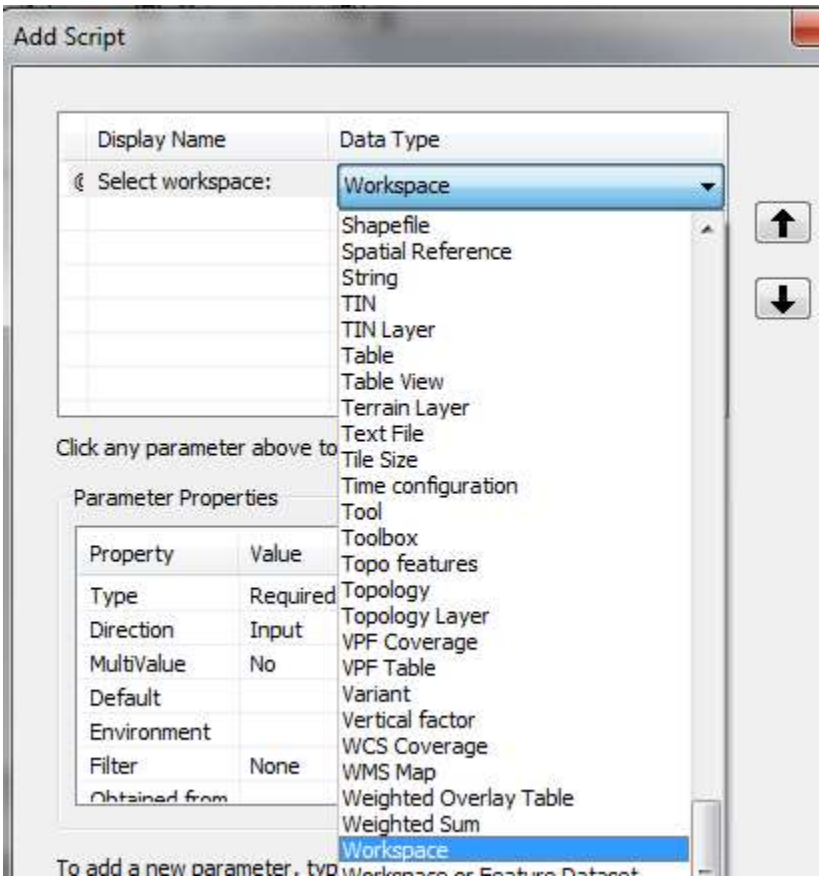
Click on next and browse to your Python script file named AddXY.py



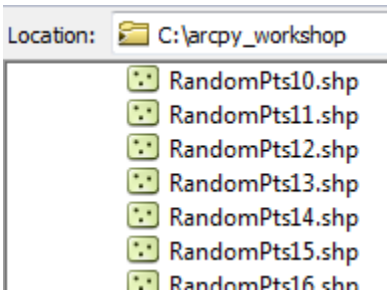


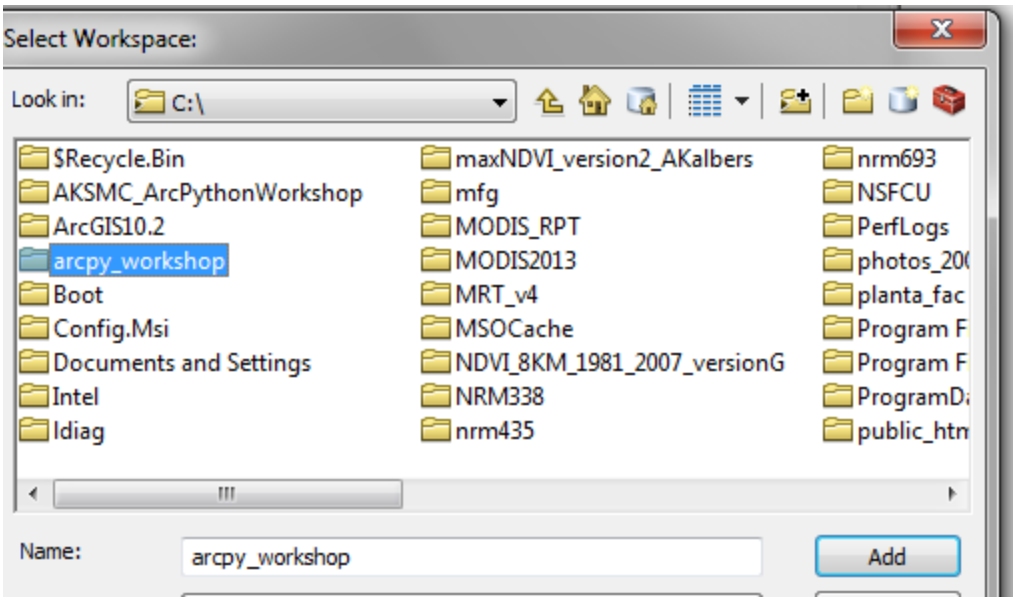
Click on next, and define your input parameter…

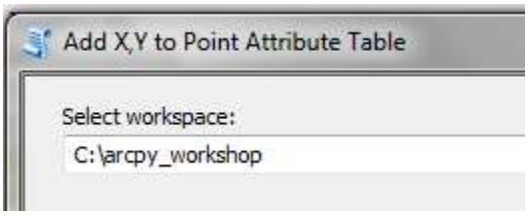




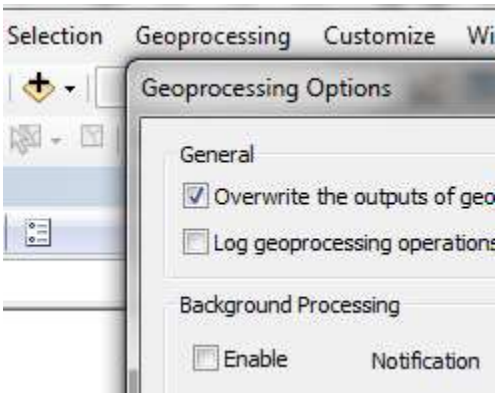
And test drive your script tool on a folder containing some point feature classes…



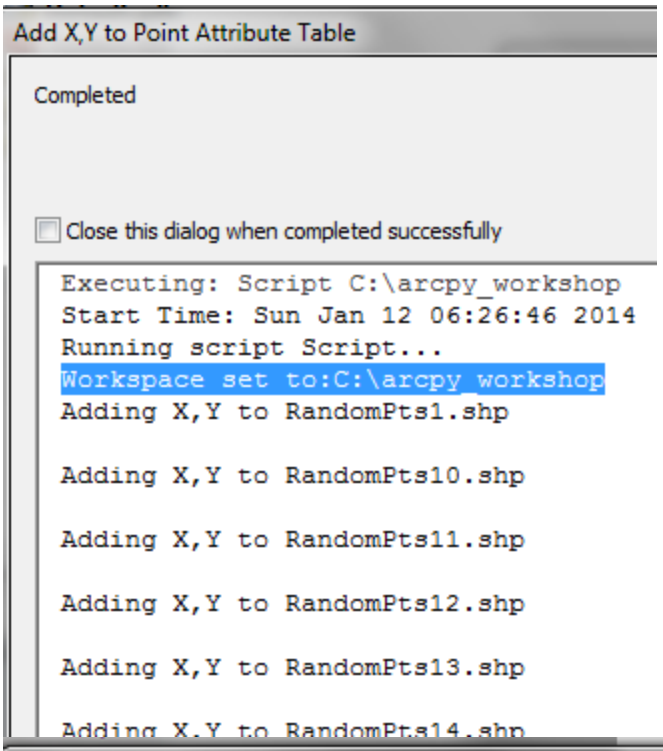


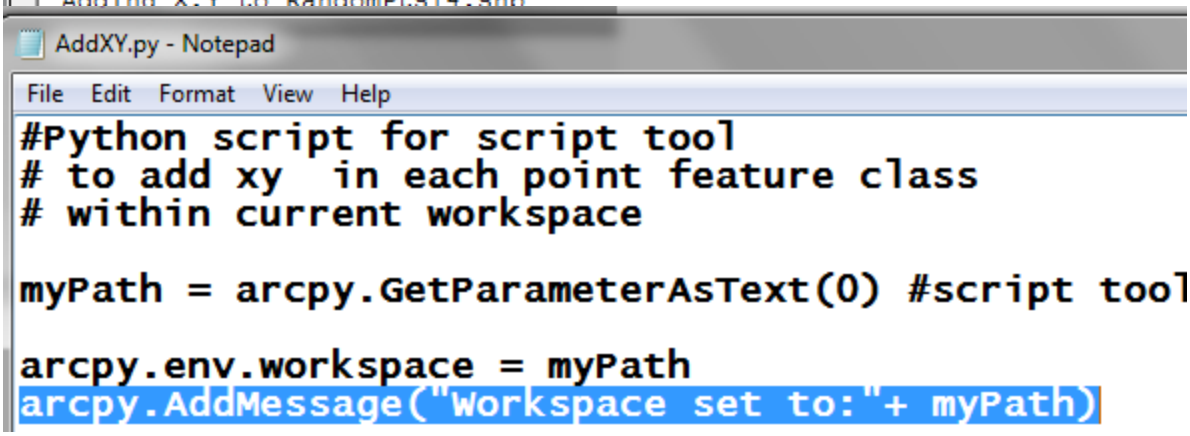


Disable background geoprocessing….

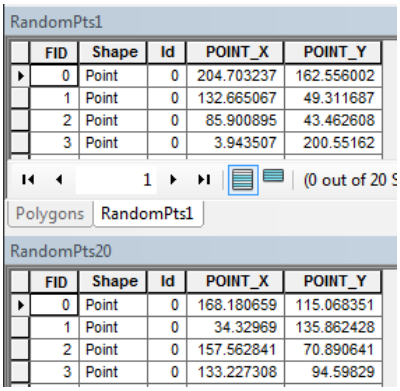


Notice the output from the arcpy.AddMessage() lines





Check out the results…



RandomPts1

FID	Shape	Id	POINT_X	POINT_Y
0	Point	0	204.703237	162.556002
1	Point	0	132.665067	49.311687
2	Point	0	85.900895	43.462608
3	Point	0	3.943507	200.55162

1 (0 out of 20 S

Polygons RandomPts1

RandomPts20

FID	Shape	Id	POINT_X	POINT_Y
0	Point	0	168.180659	115.068351
1	Point	0	34.32969	135.862428
2	Point	0	157.562841	70.890641
3	Point	0	133.227308	94.59829

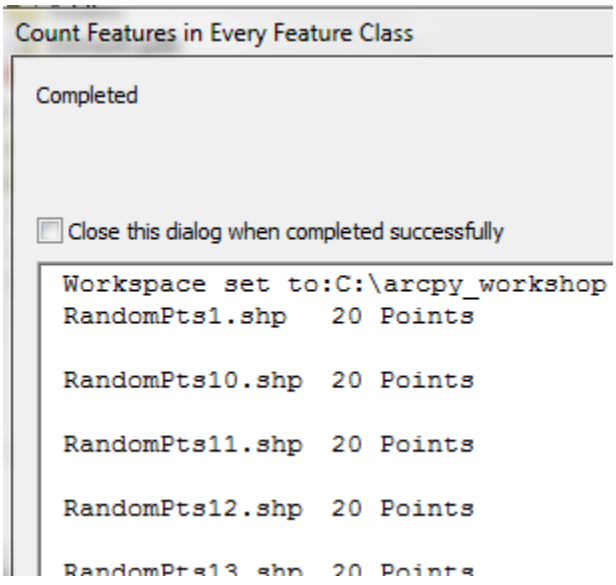
Feature Counts For Many Feature Classes

See if you can create a script tool that works with the following Python script:

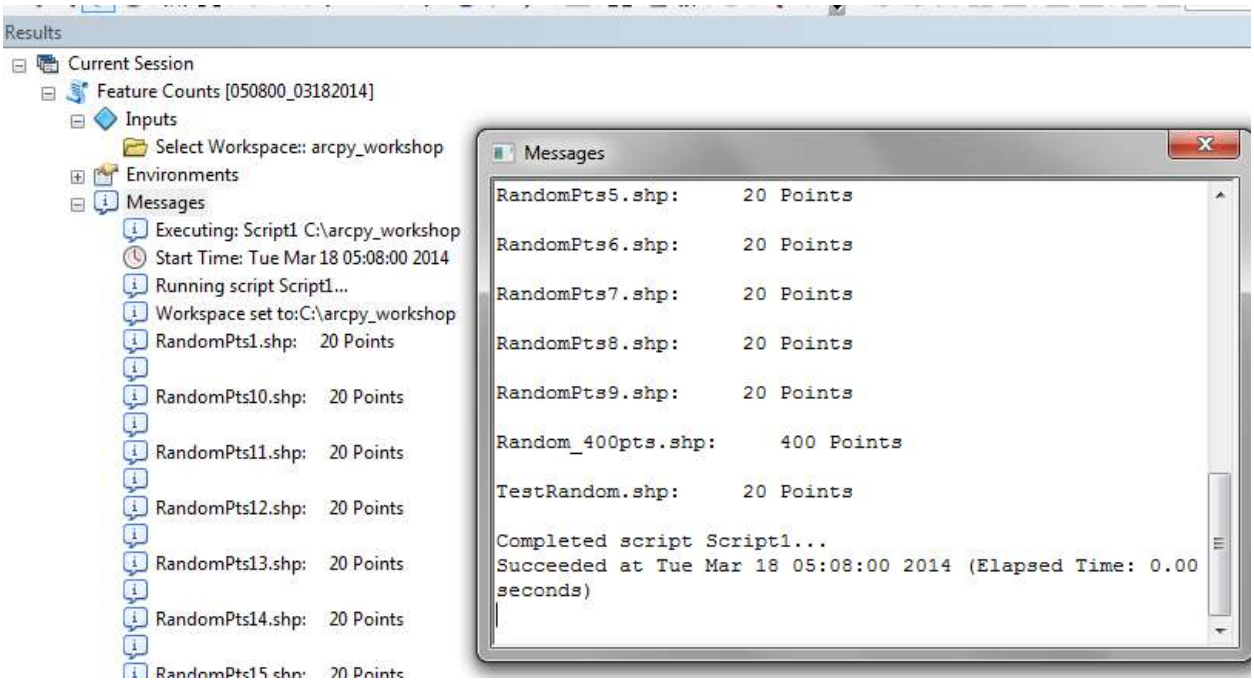
FeatureCountsScriptTool.py - Notepad

File Edit Format View Help

```
#Python script for script tool
# to report number of features in each feature class
# within current workspace

myPath = arcpy.GetParameterAsText(0) #script tool workspace parameter
arcpy.env.workspace = myPath
arcpy.AddMessage("Workspace set to:"+ myPath)

fcList = arcpy.ListFeatureClasses()
for fc in fcList: #loop through each feature class in list
    strFCount = str( arcpy.GetCount_management(fc) ) #get feature count as string
    featCount = int(strFCount) #convert string to integer value
    strShape = arcpy.Describe(fc).ShapeType #shape type is Point, Polyline, Polygon

    if(featCount == 1):
        strTxt = strFCount + " " + strShape + "\n" #\n for new line
    else:
        strTxt = strFCount + " " + strShape + "s\n" #\n for new line
    txtMsg =  fc + '\t' + strTxt
    arcpy.AddMessage(txtMsg)
```



You can also access the geoprocessing message in your Geoprocessing Results window…



Script Tool Reporting to Text File

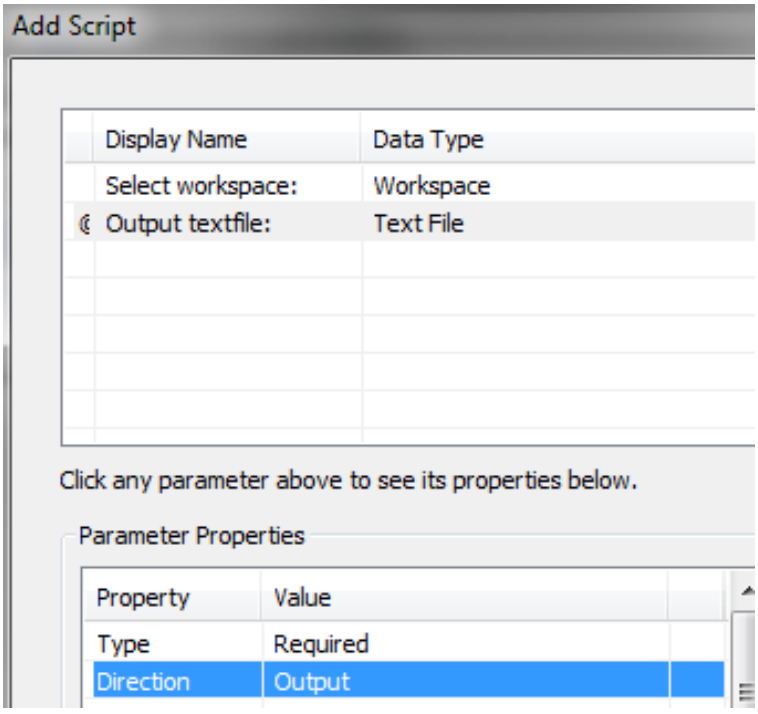
Next, see if you can create a script tool that works with the following Python script:

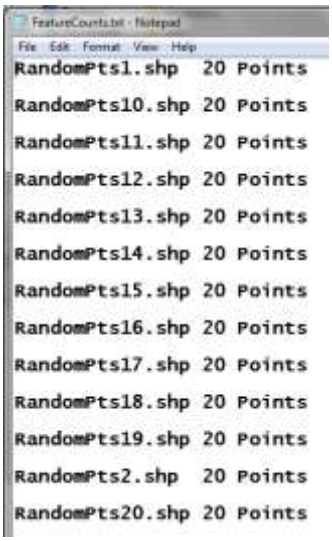
FeatureCounts_TextFile.py - Notepad

File Edit Format View Help

```
#Python script for script tool
# to output to a text file the
# number of features in each feature class
# within current workspace

myPath = arcpy.GetParameterAsText(0) #script tool workspace
txtOutFile = arcpy.GetParameterAsText(1)#output text file

arcpy.env.workspace = myPath
arcpy.AddMessage("Workspace set to:"+ myPath)
```

Note that your output text file direction should be set to output…

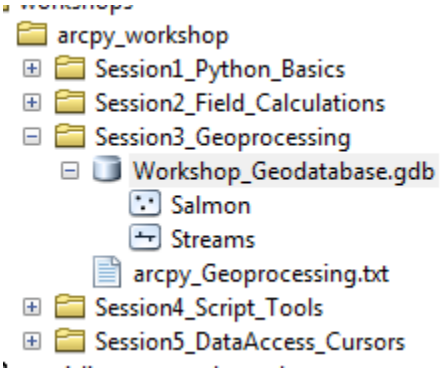


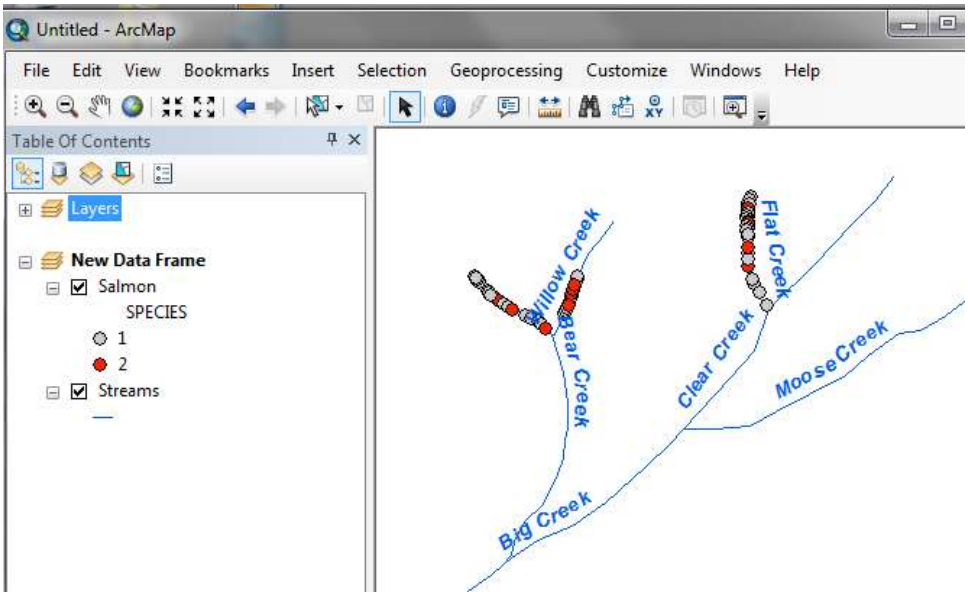

FeatureCounts.txt - Notepad
File Edit Format View Help
RandomPts1.shp 20 Points
RandomPts10.shp 20 Points
RandomPts11.shp 20 Points
RandomPts12.shp 20 Points
RandomPts13.shp 20 Points
RandomPts14.shp 20 Points
RandomPts15.shp 20 Points
RandomPts16.shp 20 Points
RandomPts17.shp 20 Points
RandomPts18.shp 20 Points
RandomPts19.shp 20 Points
RandomPts2.shp 20 Points
RandomPts20.shp 20 Points

Salmon Counts By Stream and Species

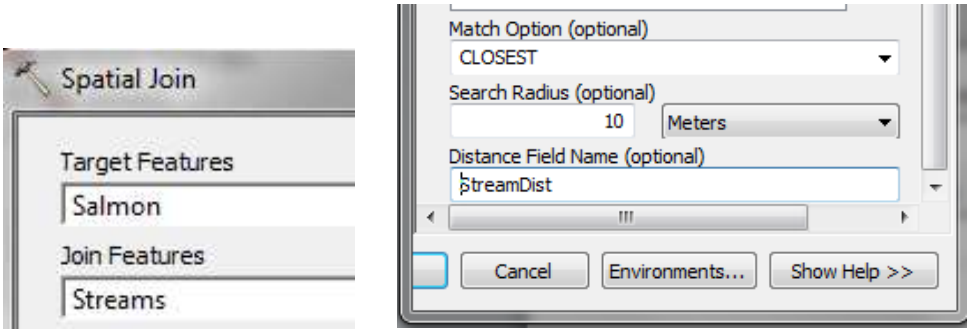
This will be a more complex script tool, so first we will execute the geoprocessing tools.

We want to create a table of salmon counts for each stream…create a new data frame and add the 2 feature classes from the workshop file geodatabase:





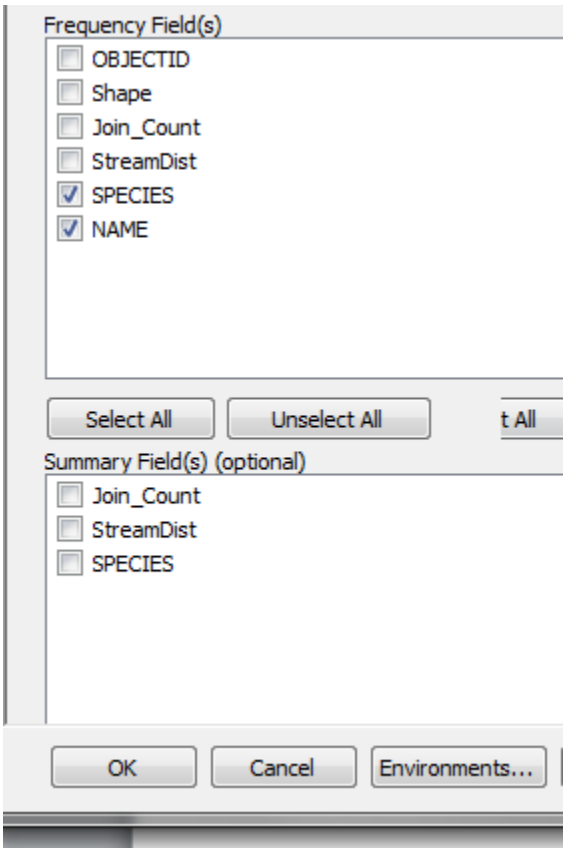
For each salmon point, find the ***closest*** stream via the ***Spatial Join*** geoprocessing tool…



salmon_stream_points

Join_Count	StreamDist	SPECIES	NAME
1	1.7	1	Ptarmigan Creek
1	2.3	1	Ptarmigan Creek
1	1.2	1	Ptarmigan Creek
1	2.5	1	Ptarmigan Creek

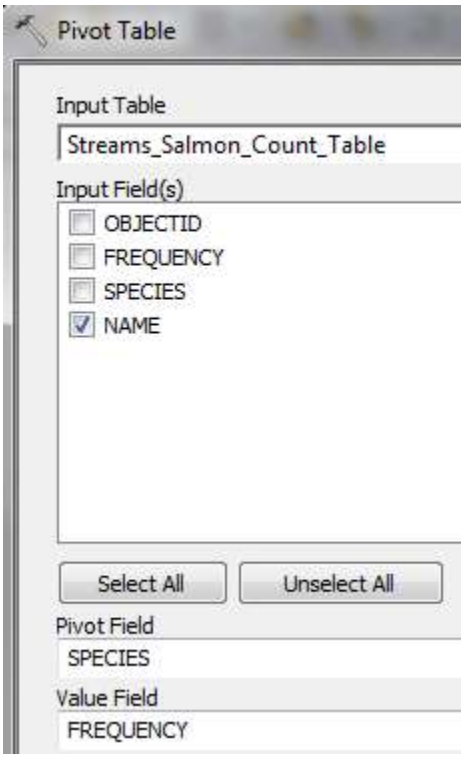
Next use the ***Frequency*** geoprocessing tool to get the count by stream name and salmon species…



Streams_Salmon_Counts_Table

FREQUENCY	SPECIES	NAME
27	1	Flat Creek
17	1	Ptarmigan Creek
11	1	Willow Creek
5	2	Flat Creek
4	2	Ptarmigan Creek
6	2	Willow Creek

Finally, use the ***Pivot Table*** geoprocessing tool to create an output table with salmon species as columns (pivot field), stream name as rows (input field):



NAME	SPECIES1	SPECIES2
Flat Creek	27	5
Ptarmigan Creek	17	4
Willow Creek	11	6

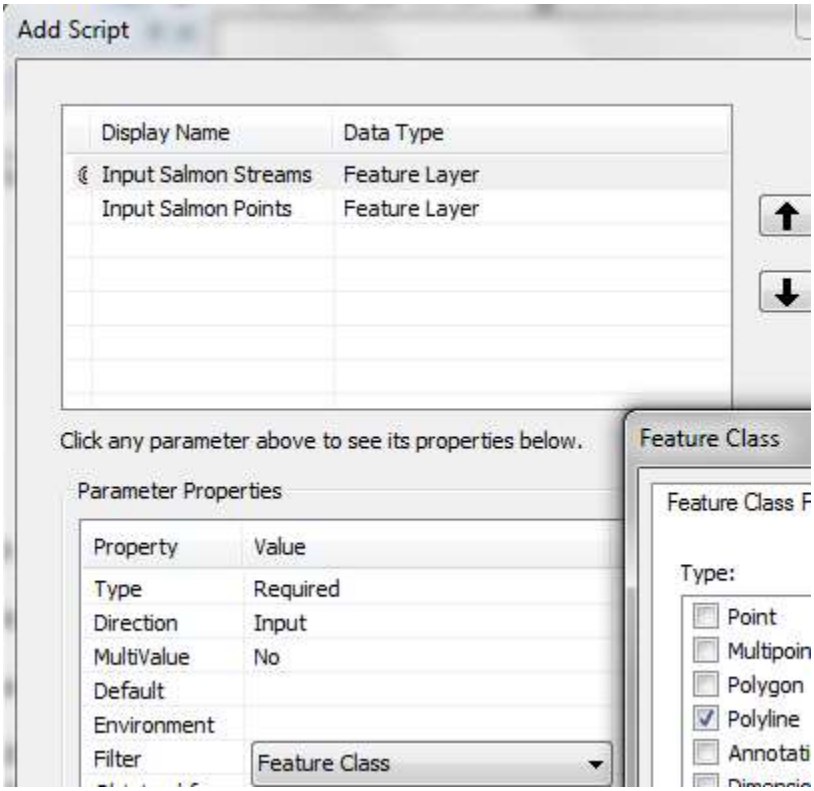
Now create a script tool to accomplish this geoprocessing using the following Python script:

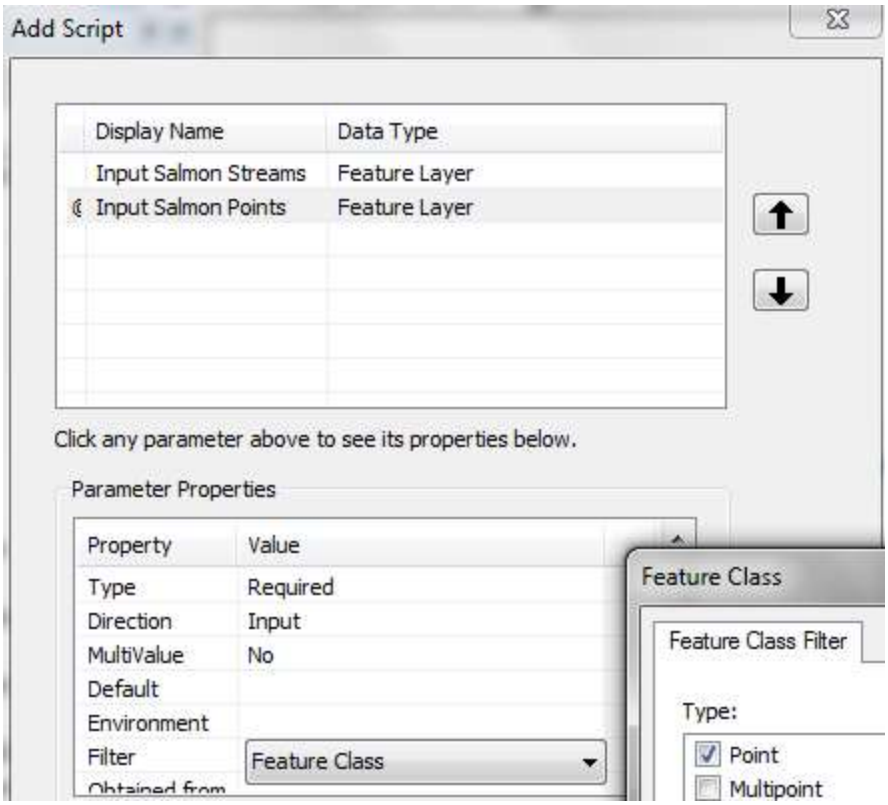
SalmonCountByStreamName.py - Notepad

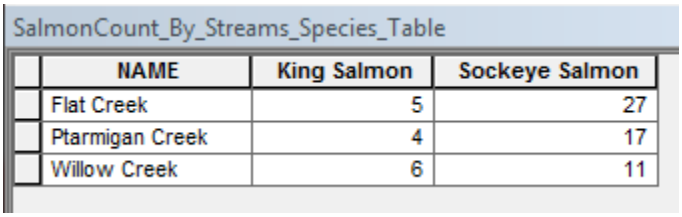
File Edit Format View Help

```
#Python script to report fish counts by stream and salmon species
arcpy.env.overwriteOutput = True

fcStreams = arcpy.GetParameterAsText(0)
fcSalmon = arcpy.GetParameterAsText(1)
outTable = arcpy.GetParameterAsText(2)

#for each salmon point, determine closest stream:
arcpy.SpatialJoin_analysis(fcSalmon,fcStreams,"salmon_stream_points

#Count salmon by species and stream name:
arcpy.Frequency_analysis("salmon_stream_points" "Salmon_Counts_Tab
```

You can apply a filter so the user can only select a line feature class as your salmon streams and only a point feature class in the input for salmon locations







The Python script used an arcpy data access cursors command to replace the 1 with “Sockeye Salmon” 2 with “King Salmon”. We cover arcpy data access cursors in the next and final session…

Take a 10-minute break