Exercise: Create a script tool

In this exercise, you will create a script tool. This will allow you to run your script much like any other geoprocessing tool in ArcToolbox.

Estimated completion time: 10 minutes

To complete exercises, you need the following:

ArcGIS Desktop 10.0 or ArcGIS Desktop 10.1 or ArcGIS Desktop 10.2 or ArcGIS Desktop 10.3 or ArcGIS Desktop 10.4 (Basic, Standard, or Advanced)

PythonWin 2.6 Or PythonWin 2.7

Note: PythonWin 2.6 and 2.7 are available as a free download. If you do not have PythonWin installed, the first course exercise provides instructions for downloading the appropriate version.

- Step 1: Download the data

To complete the exercise, you must download the data. If you have already downloaded and installed the data, continue to the next step.

- Step 2: Make your script dynamic

In this step, you will load your BufferWater.py script into PythonWin and modify your script to make it dynamic.

Open PythonWin.

Can't find PythonWin?

Click the Open button  .

Browse to and open the **BufferWater.py** script in the **..\PythonGP10_0\ScriptTool** folder.

Note: If your data is not located in the default C:\Student folder location, be sure to change the workspace path in your code.

Take a moment to look through the script, reading the comments and the Python code.

Notice that all of your paths and feature classes are hard-coded, meaning they are set to a specific path or feature class. To make your script dynamic, you will allow these hard-coded values to accept an input value when you use your script as a geoprocessing tool.

You will use the `arcpy.GetParameterAsText` function to store the input from your script tool as variables.

In the bufferList variable, remove the hard-coded "Lakes" value and replace it with `arcpy.GetParameterAsText(0)`.

BufferWater.py

```
1  import arcpy
2  #Create list of feature classes to Buffer
3  bufferList = [arcpy.GetParameterAsText(0), "Streams"]
```

Step 2a: Make your script dynamic.

The values inside the parentheses are index values. You will have several of these functions in your script: two for the Lakes and Streams feature classes, and one for the output from the Union.

Each one will have a separate index value. These values allow Python to assign the correct input value from your script tool to each of the variables in your script when you run your tool. Index values for `GetParameterAsText` are zero-based, meaning the first index is index (0), the second index is (1), and so on.

On your own, change the "Streams" to accept input from the script tool.

Tip: Remember to increment your index number.

BufferWater.py

```
1  import arcpy
2  #Create list of feature classes to Buffer
3  bufferList = [arcpy.GetParameterAsText(0), arcpy.GetParameterAsText(1)]
```

Step 2b: Make your script dynamic.

In the `Buffer_analysis` function, notice that the dissolve option is set to "ALL." This removes the boundaries between overlapping polygon features.

In your script tool, you will also specify the output name of your Union feature class.

Modify the `Union_analysis` function in your script to accept dynamic input from the script tool.

BufferWater.py

```
1  import arcpy
2  #Create list of feature classes to Buffer
3  bufferList = [arcpy.GetParameterAsText(0), arcpy.GetParameterAsText(1)]
4  #Initialize a new Python list of feature classes to be Unioned together
5  unionList = []
6  for fc in bufferList:
7      #Buffer each feature class and dissolve any overlapping polygons
8      arcpy.Buffer_analysis(fc, fc + "Buffer", "1000 meters", "", "", "ALL")
9      #Add each buffer feature class to a list of feature classes to Union
10     unionList.append(fc + "Buffer")
11 #Union the buffered feature classes
12 arcpy.Union_analysis(unionList, arcpy.GetParameterAsText(2))
```

Step 2c: Make your script dynamic.

Save your script and close PythonWin.

Step 3: Create a new toolbox

Before you can create your script tool, you will need to create a toolbox.

Start ArcMap and open your **SanJuan.mxd**.

Open the Catalog window and browse to the **..\Student\PythonGP10_0** folder.

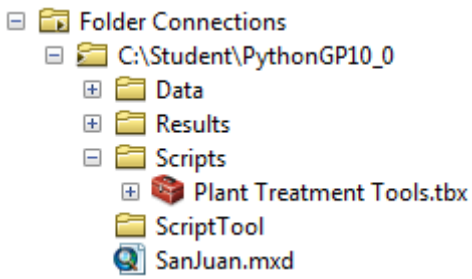
Right-click the Scripts folder, choose New, then click Toolbox.

Right-click your new toolbox and choose Properties.

On the General tab, for Name, type **Plant Treatment Tools**.

The Name field is the name of the toolbox that appears next to its icon in the ArcToolbox window or the Catalog window.

Click OK.



Your new toolbox is renamed in the Catalog window.

Step 4: Add a script tool to your toolbox

In this step, you will create your script tool in your new Plant Treatment Tools toolbox.

Right-click the Plant Treatment Tools toolbox, choose Add, then click Script.

For Name, type **CreateWaterBuffers**.

This is the actual name of the tool with which it is identified in the Python window and in scripting. This name cannot include any spaces.

To specify the name that appears next to the tool in the ArcToolbox window or the ArcCatalog window, use the Label field instead.

For Label, type **Create Water Buffers**.

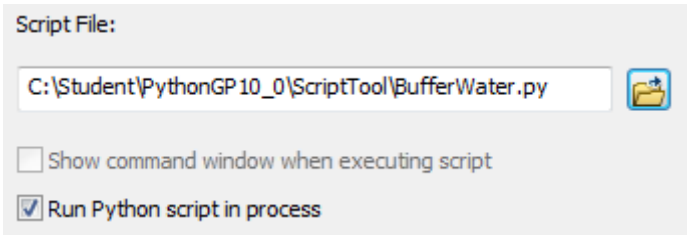
Step 4a: Add a script tool to your toolbox.

Note: You may also add a description of your tool in the Description field. When you display the tool's dialog box, the description will appear when you click the Show Help button.

Click Next.

For Script File, browse to your modified BufferWater.py script in your **..\PythonGP10_0\ScriptTool** folder and double-click it.

Confirm that the checkbox next to Run Python script in process is checked.



Step 4b: Add a script tool to your toolbox.

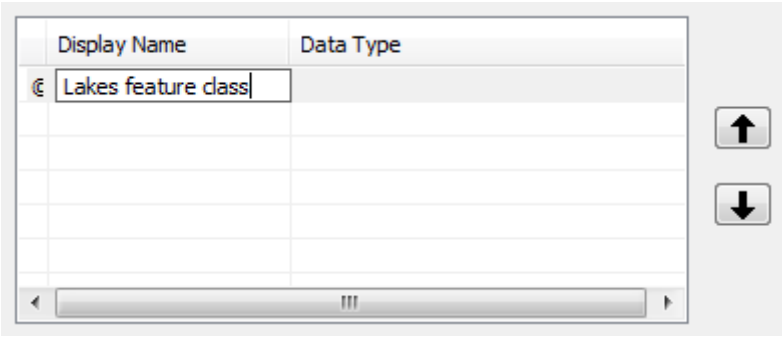
Specifying this option is more memory efficient and provides better performance for your script tool.

Click Next.

You will now enter user-input fields that you will see when you run your script tool. You will also match each field to the `GetParameterAsText` index number in your BufferWater.py script.

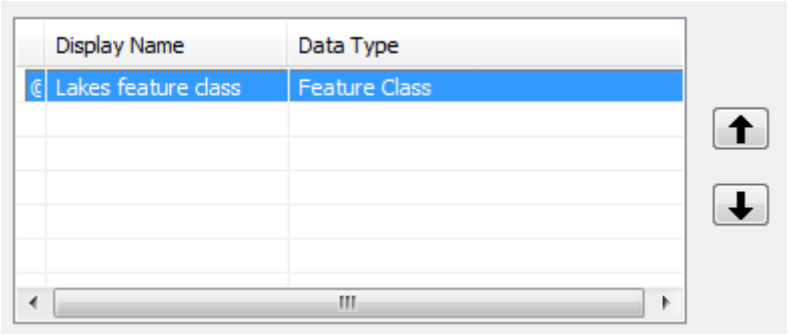
? What is the `GetParameterAsText` index number of the first parameter in your script?

Under Display Name, click the first empty field and type **Lakes feature class**.



Step 4c: Add a script tool to your toolbox.

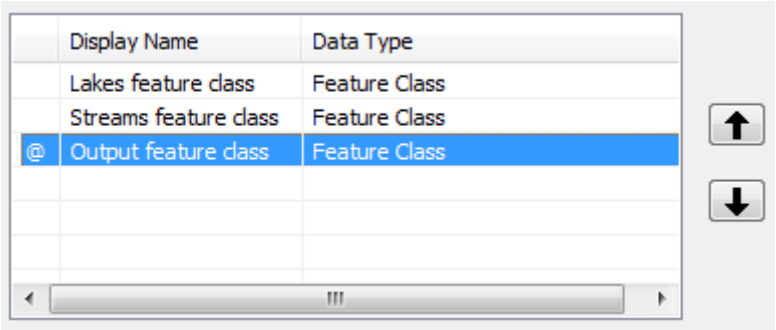
For Data Type, click in the first empty row, scroll down the list and choose Feature Class.



Step 4d: Add a script tool to your toolbox.

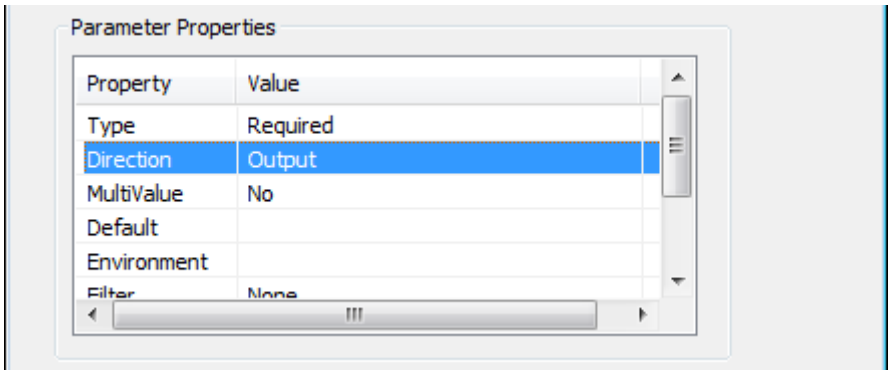
Use the following table as a guide to populate your script tool with the remaining user-input fields.

Index	Display Name	Data Type
1	**Streams feature class**	Feature Class
2	**Output feature class**	Feature Class



Step 4e: Add a script tool to your toolbox.

Under Parameter Properties, change the value of the Direction property from Input to Output.



Step 4f: Add a script tool to your toolbox.

Click Finish.

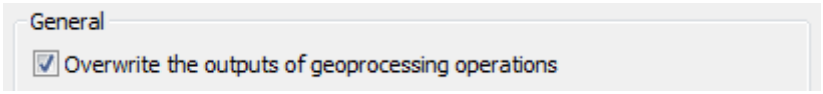
Step 5: Run your script tool

In this step, you will run your new script tool.

Before you run your new tool, you will set a geoprocessing environment.

From the Geoprocessing menu, open Geoprocessing Options.

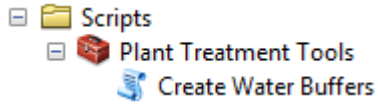
Confirm that the checkbox next to Overwrite the outputs of geoprocessing operations is checked.



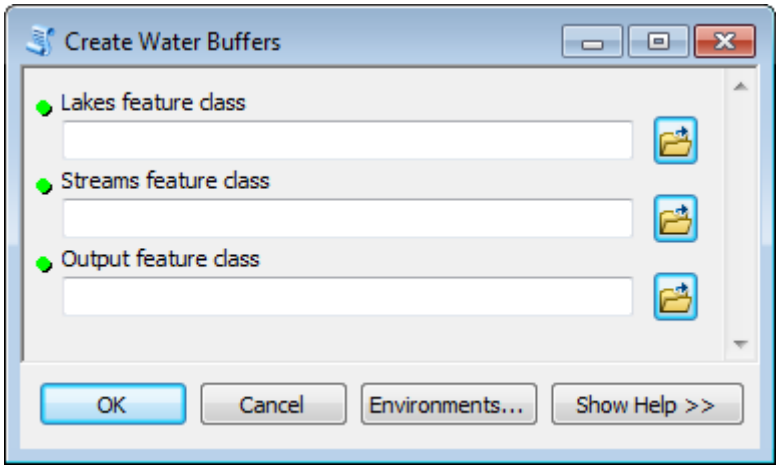
Rather than hard-coding this setting into the script (as you did in the first exercise), a script tool allows the end user to take advantage of the ArcMap Geoprocessing Options to choose whether to overwrite outputs or not.

Click OK.

In the Catalog window, expand your Plant Treatment Tools toolbox.



Double-click the Create Water Buffers tool.



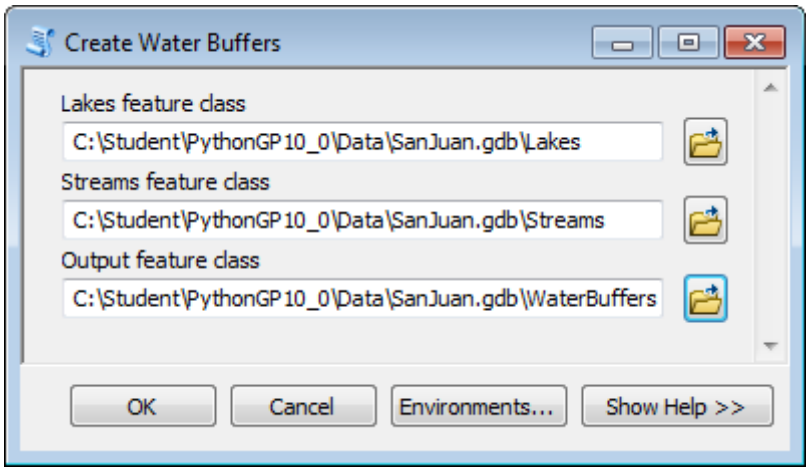
Step 5a: Run your script tool.

Browse to the feature classes in the Create Water Buffers tool:

..\Data\SanJuan.gdb**Lakes**

..\Data\SanJuan.gdb**Streams**

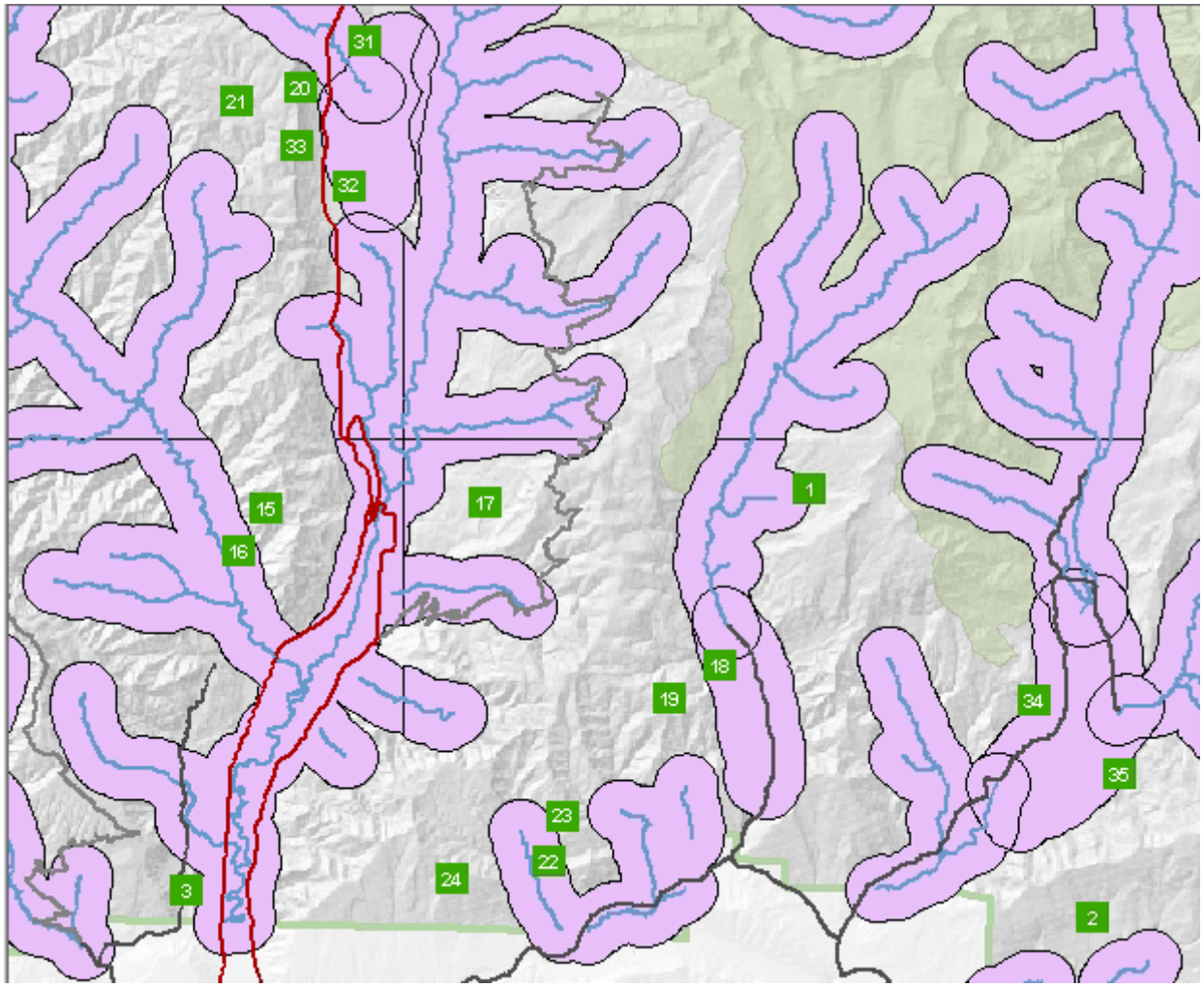
..\Data\SanJuan.gdb**WaterBuffers**



Step 5b: Run your script tool.

Click OK to run your script.

If you completed previous exercises, turn off the NonChemical layer.



Step 5c: Run your script tool.

Your new dissolved WaterBuffers layer is added to the map.

Note: Your map will look slightly different if you completed the challenge step (in the first exercise), "Dissolve polygons."

Save your map document.

Close ArcMap.