Making a Script Tool

Readings

- Zandbergen 2.10 - 2.13 to reinforce what you learned during this lesson about scripts and script tools.
- ArcGIS help topic: Adding a script tool

User input variables that you retrieve through GetParameterAsText() make your script very easy to convert into a tool in ArcGIS. We’ll take the “creating buffer” script and make it into a tool that can easily be run in ArcGIS.

Follow these steps to make a script tool:

1. Copy the code - “Creating Buffers” into a new PythonWin script and save it as buffer_user_input.py. (Note: located at end of this document)
2. Open ArcMap and display the Catalog window.
3. Expand the nodes Toolboxes > My Toolboxes.
5. Give your toolbox a name, such as "MyScriptTools".
6. Right-click your new toolbox and click Add > Script.
7. Fill in the Name, Label, and Description properties for your Script tool as shown below:

![Add Script](image)

Figure 1.10 Entering information for your script tool.
8. Click **Next** and supply the **Script File**. To do this, click the folder icon and browse to your buffer_user_input.py file.

9. Click **Next** and examine the dialog that appears. This is where you can specify the parameters of your script. The parameters are the values for which you used `arcpy.GetParameterAsText()` in your script, namely inPath, outPath, and bufferMiles. You will use this dialog to list those parameters in the same order, except you can give the parameters names that are easier to understand.

10. In the **Display Name** column that you see at the top of this wizard, click the first empty cell and type “Input Feature Class”.

11. Immediately to the right, click the first empty cell in the **Data Type** column and choose **Feature Class**. Here is one of the huge advantages of making a script tool. Instead of accepting any string as input (which could contain an error), your tool will now enforce the requirement that a feature class be used as input. ArcGIS will help you by confirming that the value entered is a path to a valid feature class. It will even supply the users of your tool with a browse button so they can browse to the feature class.

![Figure 1.11 Choosing "Feature Class."](image)

12. Just as you did in the previous steps, add a second parameter named “Output Feature Class”. The data type should again be **Feature Class**.

13. With the **Output Feature Class** parameter still highlighted, look down at the **Parameter Properties** portion of the dialog. Change the **Direction** property to **Output**.

14. Add a third property named “Buffer Distance”. Choose **Linear Unit** as the data type. This data type will allow the user of the tool to select both the distance value and the units (for example, miles, kilometers, etc.).

15. With the **Buffer Distance** parameter still highlighted, look down at the **Parameter Properties** section again. Set the **Default** property to “5 Miles” (do not include the quotes). Your dialog should look like what you see below:
16. Click Finish and, in the Catalog window, open your new script tool by double-clicking it.  
17. Try out your tool by buffering any feature class on your computer. Notice that once you  
supply the input feature class, an output feature class path is suggested for you. This is  
because you specifically set Output Feature Class as an output parameter. Also, when  
the tool is complete, examine the Results window for the custom message "All done!"  
that you added in your code.
This is a very simple example and obviously you could just run the out-of-the-box Buffer tool with similar results. Normally when you create a script tool, it will be backed with a script that runs a combination of tools and applies some logic that makes those tools uniquely useful.

There’s another benefit to this example, though. Notice the simplicity of our script tool dialog compared to the main Buffer tool:
At some point you may need to design a set of tools for beginning GIS users where only the most necessary parameters are exposed. You may also do this to enforce quality control if you know that some of the parameters must always be set to certain defaults and you want to avoid the scenario where a beginning user (or a rogue user) might change the required values. A simple script tool is effective for simplifying the tool dialog in this way.
Creating Buffers Python Script:

# This script runs the Buffer tool. The user supplies the input
# and output paths, and the buffer distance.

import arcpy
arcpy.env.overwriteOutput = True

try:
    # Get the input parameters for the Buffer tool
    inPath = arcpy.GetParameterAsText(0)
    outPath = arcpy.GetParameterAsText(1)
    bufferDistance = arcpy.GetParameterAsText(2)

    # Run the Buffer tool
    arcpy.Buffer_analysis(inPath, outPath, bufferDistance)

    # Report a success message
    arcpy.AddMessage("All done!")

except:
    # Report an error message
    arcpy.AddError("Could not complete the buffer")

    # Report any error messages that the Buffer tool might have encountered
    arcpy.AddMessage(arcpy.GetMessages())