


A summary of converting a Python script into an ArcTool

Creating ArcScripts

This video will provide a summary of the steps that should be taken to convert a Python script into an ArcTool.

Review: creating an ArcScript

1.  Write script using “hard-wired” parameters and debug script
2. Make sure `sys` module has been imported
3. Use `sys.argv[n]` to change appropriate “hard-wired” variables to user-defined variables
4. Add any error catching code (`try / except`)
5. Use `arcpy.AddMessage`, `arcpy.AddWarning`, or `arcpy.AddError` to send messages to ArcGIS progress window
6. In ArcGIS, create toolbox and add script to toolbox
 - Step through wizard to set up script parameters

2

In this slide, we'll create a checklist of the steps needed to convert a Python script into an ArcScript. The first step is to develop and debug the script in Python.

The second step is to make sure the **sys** module has been imported.

Use the `sys` module's **argv** function to change the appropriate hard-wired variables to user-defined variables.

Add `try/except` statements as needed to catch errors.

Use `arcpy`'s `AddMessage`, `AddWarning`, and `AddError` tools, as needed, to print information to ArcGIS's progress window.

In ArcGIS, add the script to a new or existing toolbox using the script import wizard.

For this video, we'll assume that the script has been developed and tested.

Step 2: sys module

```
# import modules...  
import arcpy, sys
```

Import sys module if
not imported already



3

Step 2 in converting a Python script to an ArcScript is to make sure the **sys** module has been imported.

Step 3: script arguments

```
31 # input feature class (argument 1)
32 inputFC = sys.argv[1]
33 # input feature class (argument 2)
34 expression = sys.argv[2]
35 # input feature class (argument 3)
36 outputFC = sys.argv[3]
```

Number arguments in order that they will listed in ArcGIS dialog window.

4

The next step is to replace hard-wired variables with user-defined parameters as needed.

The parameter numbers in the **sys.argv** functions should be in the order that the parameters will be requested in ArcMap.

Step 4: handling errors

```
# import modules...  
import arcpy, sys, traceback
```

Import traceback module



5

Import the traceback module to allow error messages to be retrieved.

```
14 import arcpyscripting, sys, traceback
15
16 try
17
18     # create the geoprocessor...
19     gp = arcpyscripting.create()
20
21     # set license level...
22     gp.SetProduct("ArcView")
23
24     # load toolboxes...
25     gp.AddToolbox("C:\Program Files\ArcGIS\ArcToolbox\Toolboxes\Analysis Tools.tbx")
26
27     # allow output to be overwritten...
28     gp.OverwriteOutput = 1
29
30     #####
31     ## Script variables that can be defined by the user (script arguments)...
32
33     # input feature class (argument 1)
34     inputFC = sys.argv[1]
35     # input feature class (argument 2)
36     field = sys.argv[2]
37
38 except:
39
40     tb = sys.exc_info()[2]
41     tbinfo = traceback.format_tb(tb)[0]
42     pymsg = tbinfo + "\n" + str(sys.exc_type) + ": " + str(sys.exc_value)
43
44     # Select Tool: use inputFC, outputFC, and expression variables defined previously...
45     gp.Select_analysis(inputFC, outputFC, expression)
46
47
48 except:
49     tb = sys.exc_info()[2]
50     tbinfo = traceback.format_tb(tb)[0]
51     pymsg = tbinfo + "\n" + str(sys.exc_type) + ": " + str(sys.exc_value)
```

try

Include entire script in try part

Append except part to script

6

The try/except statement should be used to catch any errors in the script.

The entire script should be placed in the try part of the statement.

The except part of the statement should be appended to the end of the script and include the code shown here.

This code will extract Python error information from the traceback module.

Step 5: send messages to ArcGIS

except:

```
tb = sys.exc_info()[2]
tbinfo = traceback.format_tb(tb)[0]
pymsg = tbinfo + "\n" + str(sys.exc_type) + ": " + str(sys.exc_value)
arcpy.AddError(pymsg)
arcpy.AddError(arcpy.GetMessages(2))
```

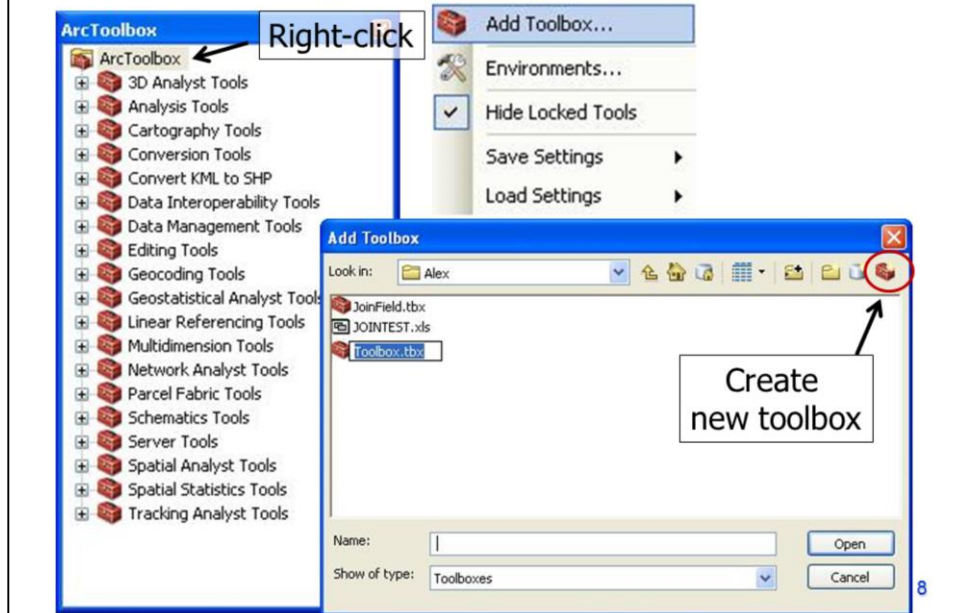
ArcTool error
message

Failed line number
and type of error

7

The AddError methods should be used to print the error information to the ArcGIS progress window. This information should include error messages from any failed ArcTools. The error information will be helpful in troubleshooting the problem.

Adding/creating a toolbox

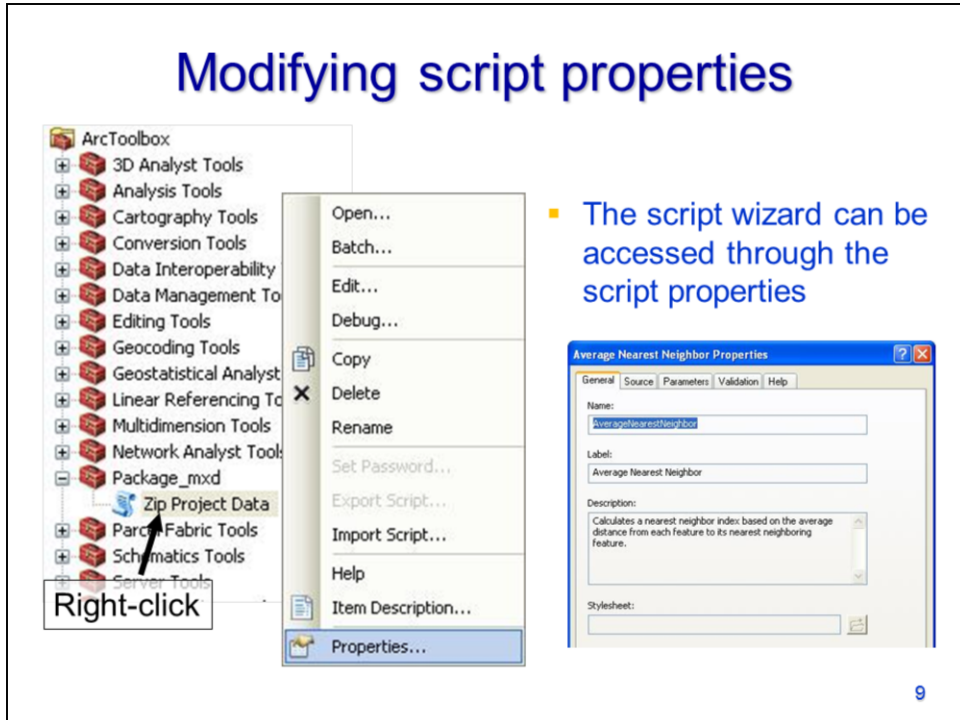


To import a script into ArcToolbox, an editable toolbox must be available.

An toolbox can be added to ArcToolbox by right-clicking on ArcToolbox and then clicking on **Add Toolbox**.

Navigate to an existing toolbox or click on the **Create toolbox** button to create a new toolbox.

Modifying script properties



The screenshot illustrates the process of modifying script properties. On the left, the ArcToolbox is shown with a right-click context menu open over the 'Zip Project Data' tool. The 'Properties...' option at the bottom of the menu is highlighted. A callout box labeled 'Right-click' points to the context menu. To the right, a blue square bullet point states: 'The script wizard can be accessed through the script properties'. Below this, the 'Average Nearest Neighbor Properties' dialog box is shown, featuring tabs for 'General', 'Source', 'Parameters', 'Validation', and 'Help'. The 'General' tab is active, displaying fields for 'Name' (AverageNearestNeighborTool), 'Label' (Average Nearest Neighbor), and a 'Description' field containing the text: 'Calculates a nearest neighbor index based on the average distance from each feature to its nearest neighboring feature.' A 'Stylesheet' field is also visible at the bottom.

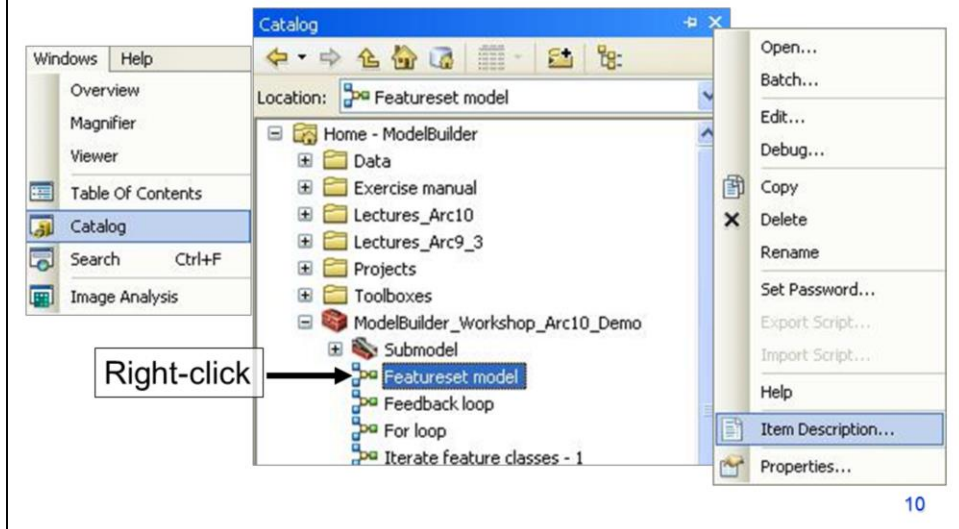
- The script wizard can be accessed through the script properties

9

Add a script to the toolbox using the import wizard.

If an ArcScript's properties needs to be modified after it has been imported, then access the import wizard pages by right-clicking on the script and selecting properties. The tabs at the top of the Script Properties window will allow you to navigate through the pages of the wizard.

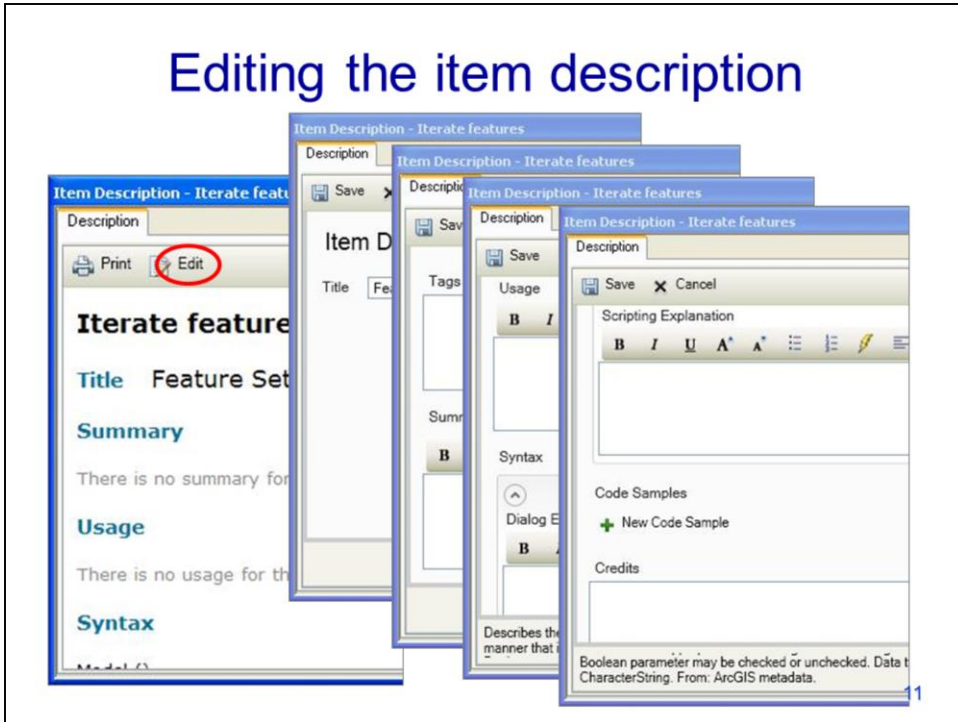
Documenting an ArcScript: accessing item description page



To add documentation to an ArcScript, locate the tool in the Catalog or ArcToolbox.

Click on the **Item Description** to access the tool's documentation.

Editing the item description



Click on Edit in the Item Description page to add documentation.