# Exercise 13

## Creating custom tools

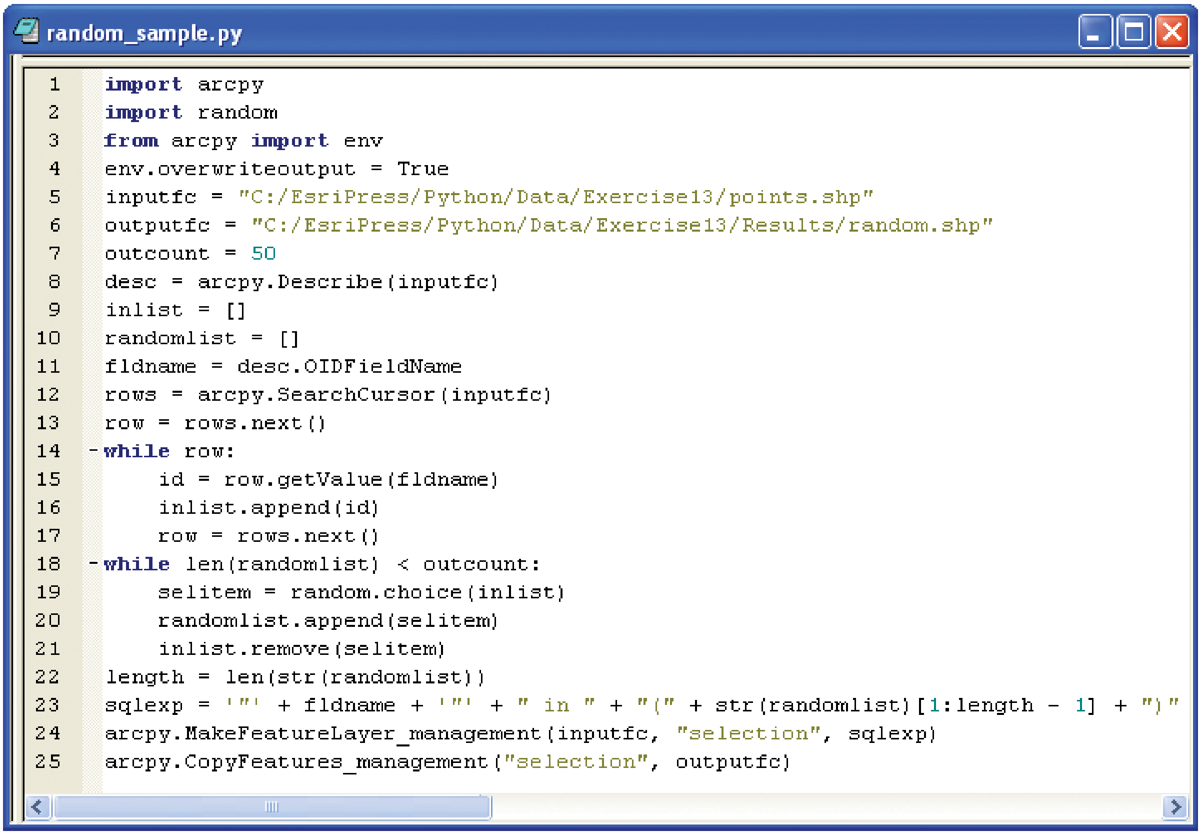
### Examine the script

In this exercise, you will create a script tool that creates a random selection of features from an existing feature class and saves the result as a new feature class. First, examine the script.

1 **Start PythonWin. On the menu bar, click the Open button and browse to the script random_sample.py in your C:\EsriPress\Python\Data\Exercise13 folder.**

2 **Click OK.**

## 3 Examine the script.



```
import arcpy
import random
from arcpy import env
env.overwriteoutput = True
inputfc = "C:/EsriPress/Python/Data/Exercise13/points.shp"
outputfc = "C:/EsriPress/Python/Data/Exercise13/Results/random.shp"
outcount = 50
desc = arcpy.Describe(inputfc)
inlist = []
randomlist = []
fldname = desc.OIDFieldName
rows = arcpy.SearchCursor(inputfc)
row = rows.next()
while row:
    id = row.getValue(fldname)
    inlist.append(id)
    row = rows.next()
while len(randomlist) < outcount:
    selitem = random.choice(inlist)
    randomlist.append(selitem)
    inlist.remove(selitem)
length = len(str(randomlist))
sqlexp = '"' + fldname + '"' + " in " + "(" + str(randomlist)[1:length - 1] + ")"
arcpy.MakeFeatureLayer_management(inputfc, "selection", sqlexp)
arcpy.CopyFeatures_management("selection", outputfc)
```

Notice that the names of the input and output feature classes and the count of random features to be selected are hard-coded into the script.

A few points to note about the script:

- The script creates a new empty list, `inlist`, and this is populated with the OID (object ID) values of all the features in the input feature class.
- The script creates a new empty list, `randomlist`, and this is populated with 50 OID values randomly selected from `inlist`.
- The output feature class is created by using the OID values in an SQL expression to select features from the input feature class. The SQL expression looks a bit complicated but essentially uses the `in` operator to query each feature in the input feature class as to whether its OID value is part of `randomlist`.

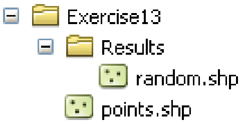
First, you will run the script to see what the output looks like.

**4 Run the script.**

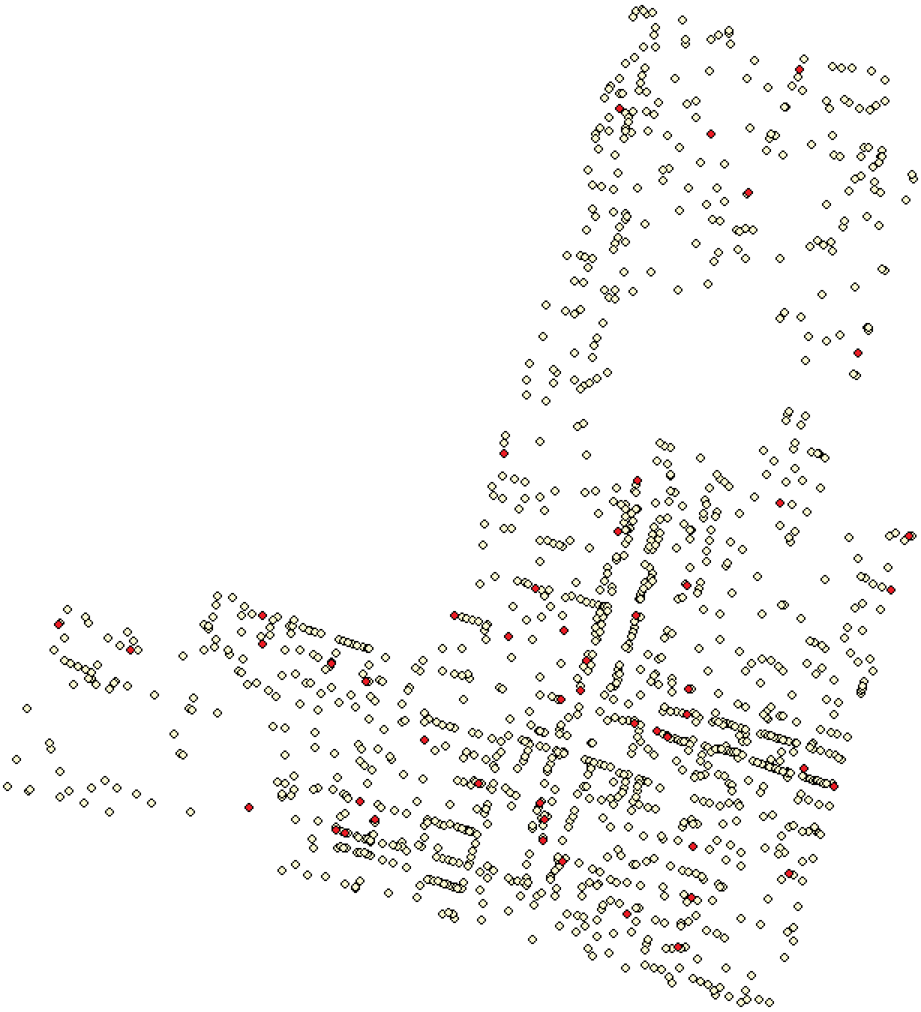
**5 Close PythonWin.**

Next, you will examine the results.

**6 Start ArcMap. Open the Catalog window. Navigate to the C:\EsriPress\Python\Data\Exercise13\Results folder.** →



**7 Drag the shapefiles points.shp (from the Exercise13 folder) and random.shp (from the Results folder for exercise 13) into the data frame.** When you examine the contents of the two shapefiles, you will notice that the points.shp file contains 1,314 features, and the random.shp file contains only 50 features created through a random selection. If you were to run the script again, you would get a different random selection.



Next, instead of running the script from PythonWin, you will create a script tool so that it can be run from within an ArcGIS for Desktop application.

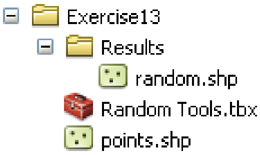
## Create a custom tool

Script tools are located inside a toolbox. First, you will create a new custom toolbox. Then within that toolbox, you will create a new tool that makes use of the preceding script.

**1 In ArcMap, make sure the Catalog window is open and browse to the Exercise13 folder.**

**2 Right-click the Exercise13 folder and click New > Toolbox.**

**3 Name the new toolbox** Random Tools.tbx**.**



⊟ Exercise13
⊟ Results
random.shp
Random Tools.tbx
points.shp

**4 Right-click the new toolbox and click Add > Script.**

**5 On the Add Script dialog box in the first panel, enter the following information: →**

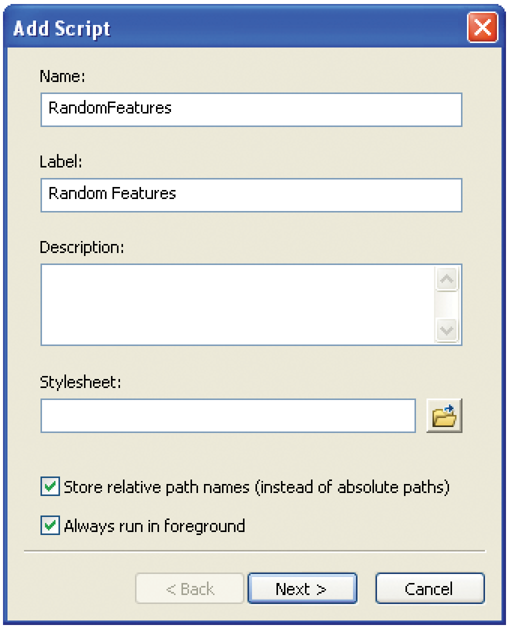
a. For Name, type **RandomFeatures**.

b. For Label, type **Random Features**.

c. Select the check box "Store relative path names".

d. Select the check box "Always run in foreground".

**6 Click Next.**



Add Script

Name: RandomFeatures

Label: Random Features

Description:

Stylesheet:

☑ Store relative path names (instead of absolute paths)

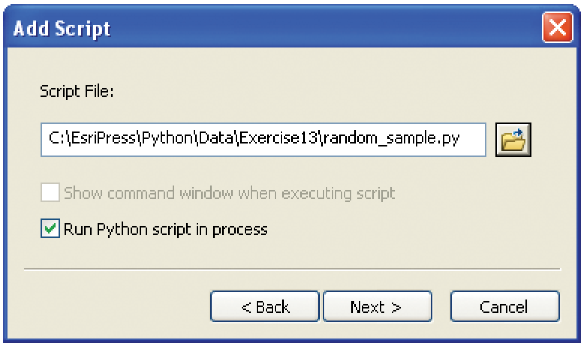
☑ Always run in foreground

**7 On the Add Script dialog box in the second panel, enter the following information:**

a. For Script File, click the Browse button, navigate to the Exercise13 folder, and double-click the script random_sample.py.

b. Leave the check box "Show command window when executing script" cleared. This check box typically appears shaded.

c. Select the check box "Run Python script in process".



Add Script

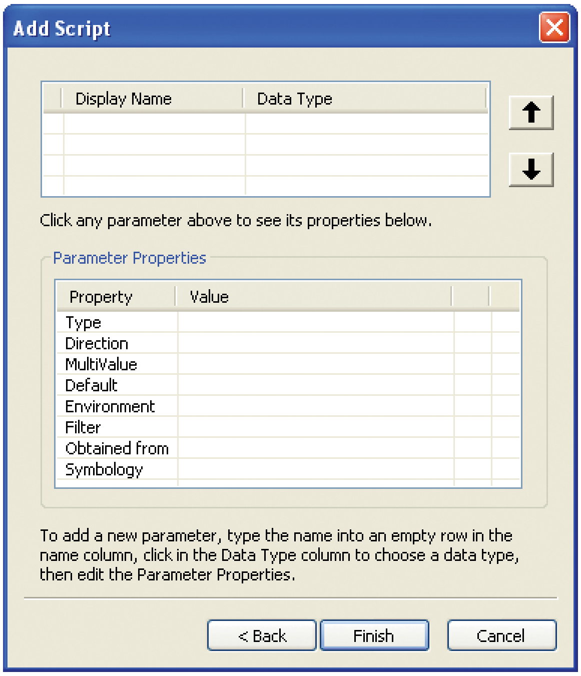
Script File:

C:\EsriPress\Python\Data\Exercise13\random_sample.py

□ Show command window when executing script

☑ Run Python script in process

**8 Click Next.** On the Add Script dialog box in the third panel, you can add the script parameters. For now, you will leave it blank and return to it later.



Add Script

| Display Name | Data Type |
|---|---|
| | |
| | |
| | |

Click any parameter above to see its properties below.

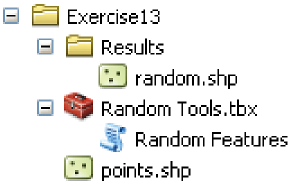
Parameter Properties

| Property | Value |
|---|---|
| Type | |
| Direction | |
| MultiValue | |
| Default | |
| Environment | |
| Filter | |
| Obtained from | |
| Symbology | |

To add a new parameter, type the name into an empty row in the name column, click in the Data Type column to choose a data type, then edit the Parameter Properties.

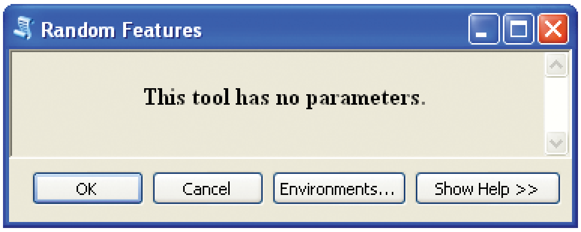
9 **Click Finish.** The script tool is now created but has no parameters yet, because this step was skipped during the initial creation of the tool.

10 **In the Catalog window, navigate to the Random Tools toolbox. Notice that the Random Features script tool has been added to the toolbox.**



```
⊟ Exercise13
    ⊟ Results
        random.shp
    ⊟ Random Tools.tbx
        Random Features
    points.shp
```

11 **Double-click the Random Features tool.** This opens the dialog box of the newly created tool. The tool does not have any parameters so the dialog box is not very informative.



Random Features

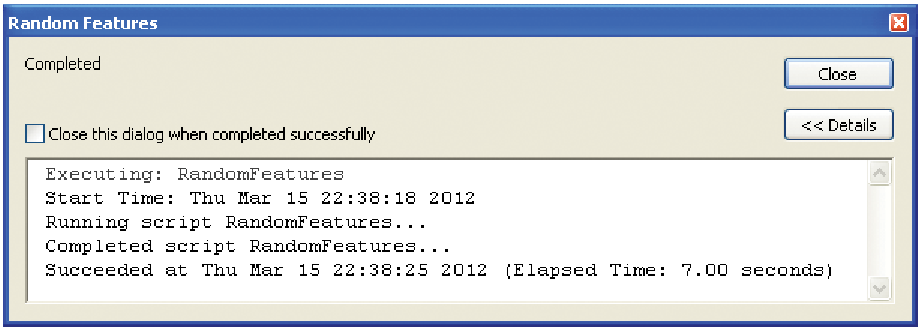
This tool has no parameters.

The tool can be executed, however—that is, the script can be run from the new tool dialog box.

12 **Leave the tool dialog box open. In the ArcMap table of contents, right-click the random layer and click Remove.**

13 **In the Catalog window, browse to the Results folder for exercise 13. Right-click the feature class random.shp and click Delete.**

14 **On the Random Features tool dialog box, click OK.** This runs the script, and the progress dialog box appears.



Random Features

Completed

□ Close this dialog when completed successfully

```
Executing: RandomFeatures
Start Time: Thu Mar 15 22:38:18 2012
Running script RandomFeatures...
Completed script RandomFeatures...
Succeeded at Thu Mar 15 22:38:25 2012 (Elapsed Time: 7.00 seconds)
```

15 **Click Close to close the tool progress dialog box.**

**16 In the Catalog window, browse to the Results folder for exercise 13 and confirm that a new feature class,** random.shp**, has been created.**

Running the script tool this way is possible, but the script still uses the original hard-coded values. The next step is to add parameters to the tool and pass them to the script so the user can choose the values.

## Set the tool parameters

The new tool will have three parameters:

1. An input features class

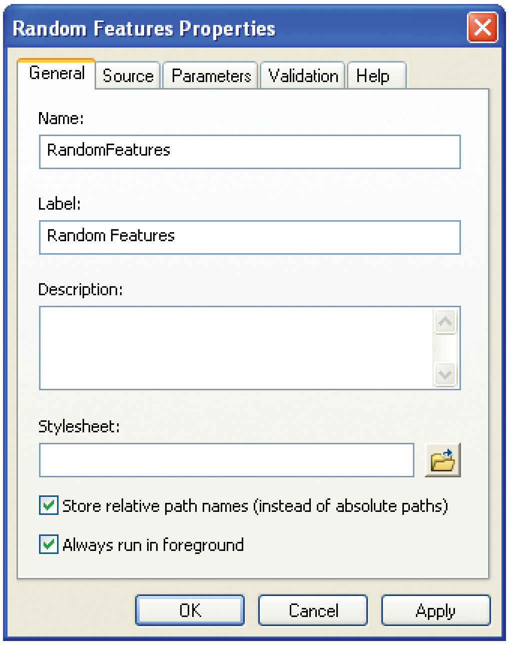
2. An output feature class containing randomly selected features

3. The number of features to be selected

It is possible, of course, to select features based on other criteria, such as a percentage of the input features, but using the number of features will suffice for this exercise.

You will start by setting the tool parameters of the script tool.

**1 In the Catalog window, right-click the Random Features tool and click Properties.** This brings up the tool properties dialog box.



Random Features Properties

Name: RandomFeatures

Label: Random Features

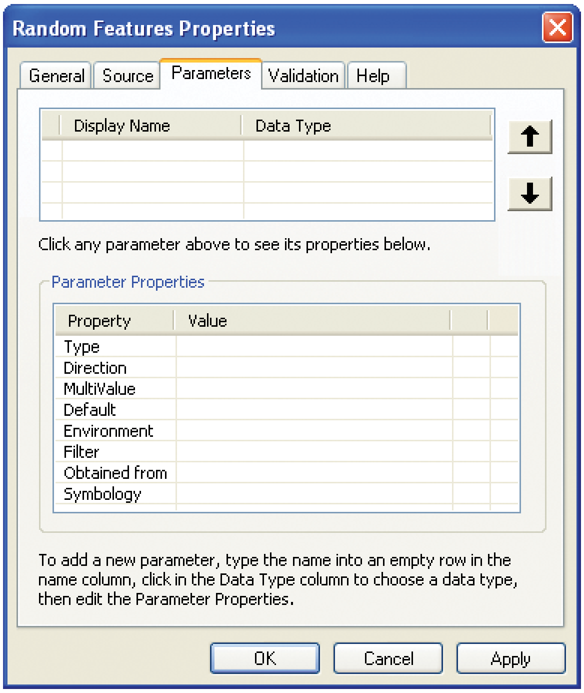
Description:

Stylesheet:

☑ Store relative path names (instead of absolute paths)

☑ Always run in foreground

**2 Click the Parameters tab.** →



| Display Name | Data Type |
|---|---|

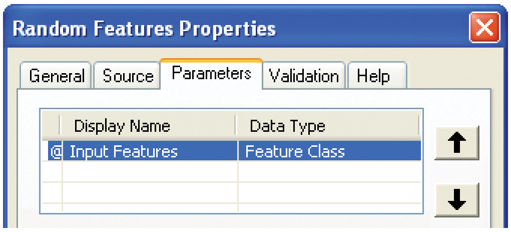
Click any parameter above to see its properties below.

Parameter Properties

| Property | Value |
|---|---|
| Type | |
| Direction | |
| MultiValue | |
| Default | |
| Environment | |
| Filter | |
| Obtained from | |
| Symbology | |

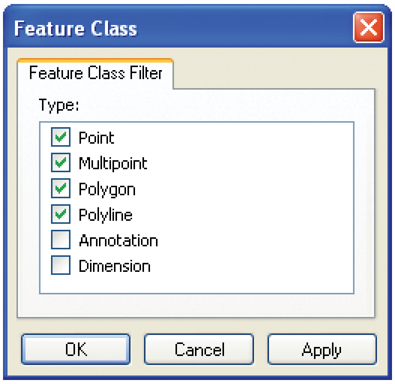
To add a new parameter, type the name into an empty row in the name column, click in the Data Type column to choose a data type, then edit the Parameter Properties.

**3 On the tab's uppermost panel, click in the first empty row in the Display Name column and type** Input Features**. For Data Type, select Feature Class.** →



| Display Name | Data Type |
|---|---|
| @ Input Features | Feature Class |

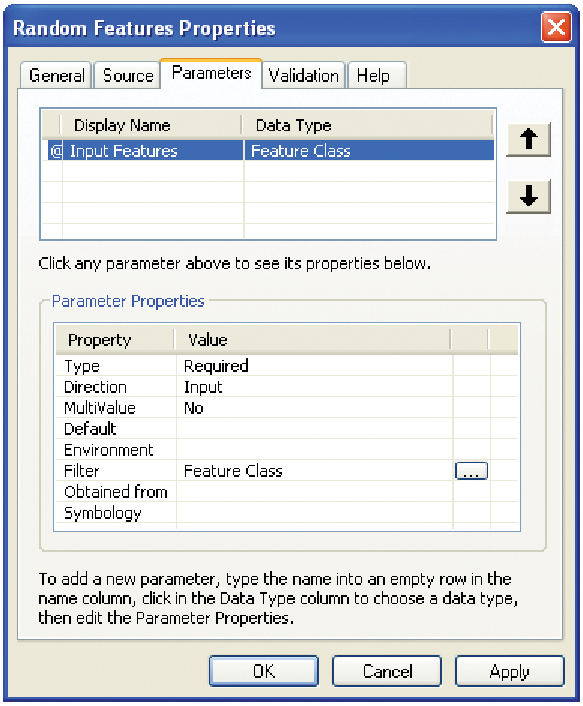
**4 In the Parameter Properties panel, click in the cell to the right of the Filter parameter and select Feature Class. On the Feature Class dialog box, clear the Annotation and Dimension check boxes and click OK.** →



Type:

☑ Point
☑ Multipoint
☑ Polygon
☑ Polyline
□ Annotation
□ Dimension

This completes the first parameter. →



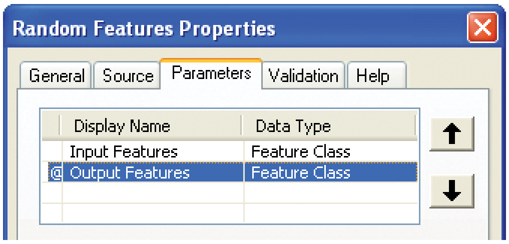
| Display Name | Data Type |
|---|---|
| Input Features | Feature Class |

Click any parameter above to see its properties below.

| Property | Value |
|---|---|
| Type | Required |
| Direction | Input |
| MultiValue | No |
| Default | |
| Environment | |
| Filter | Feature Class |
| Obtained from | |
| Symbology | |

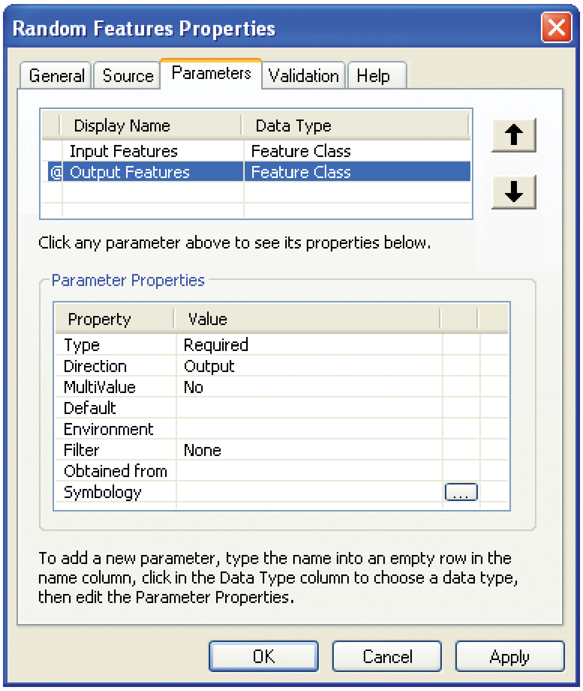
To add a new parameter, type the name into an empty row in the name column, click in the Data Type column to choose a data type, then edit the Parameter Properties.

**5** **Click in the second row in the Display Name column and type** Output Features**. For Data Type, select Feature Class.**



| Display Name | Data Type |
|---|---|
| Input Features | Feature Class |
| Output Features | Feature Class |

**6** **In the Parameter Properties panel, click in the cell to the right of the Direction property and select Output.** This completes the second parameter. →



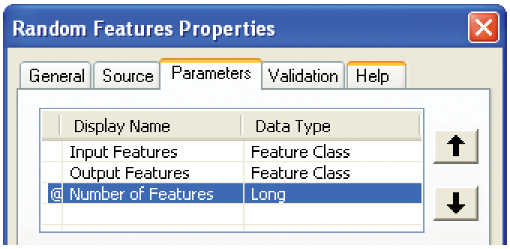
| Display Name | Data Type |
|---|---|
| Input Features | Feature Class |
| Output Features | Feature Class |

Click any parameter above to see its properties below.

| Property | Value |
|---|---|
| Type | Required |
| Direction | Output |
| MultiValue | No |
| Default | |
| Environment | |
| Filter | None |
| Obtained from | |
| Symbology | |

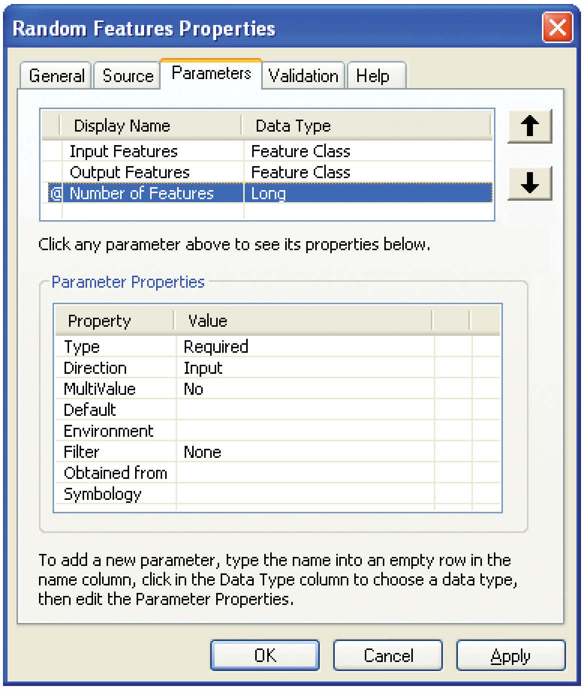
To add a new parameter, type the name into an empty row in the name column, click in the Data Type column to choose a data type, then edit the Parameter Properties.

**7** **Click in the third row in the Display Name column and type** Number of Features**. For Data Type, select Long.**



| Display Name | Data Type |
|---|---|
| Input Features | Feature Class |
| Output Features | Feature Class |
| Number of Features | Long |

8 **Leave the Parameter Properties at the defaults.** This completes the third parameter.



Random Features Properties

| | Display Name | Data Type |
|---|---|---|
| | Input Features | Feature Class |
| | Output Features | Feature Class |
| @ | Number of Features | Long |

Click any parameter above to see its properties below.

Parameter Properties

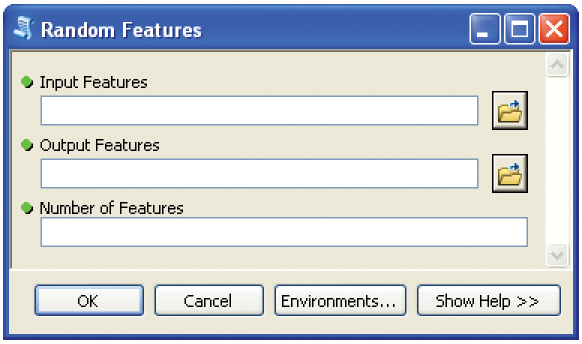
| Property | Value |
|---|---|
| Type | Required |
| Direction | Input |
| MultiValue | No |
| Default | |
| Environment | |
| Filter | None |
| Obtained from | |
| Symbology | |

To add a new parameter, type the name into an empty row in the name column, click in the Data Type column to choose a data type, then edit the Parameter Properties.

9 **Click OK.**

Next, you will view the parameters on the tool dialog box.

10 **In the Catalog window, navigate to the Random Tools toolbox. Double-click the Random Features tool.** This opens the tool dialog box with its newly created parameters.



Random Features

● Input Features

● Output Features

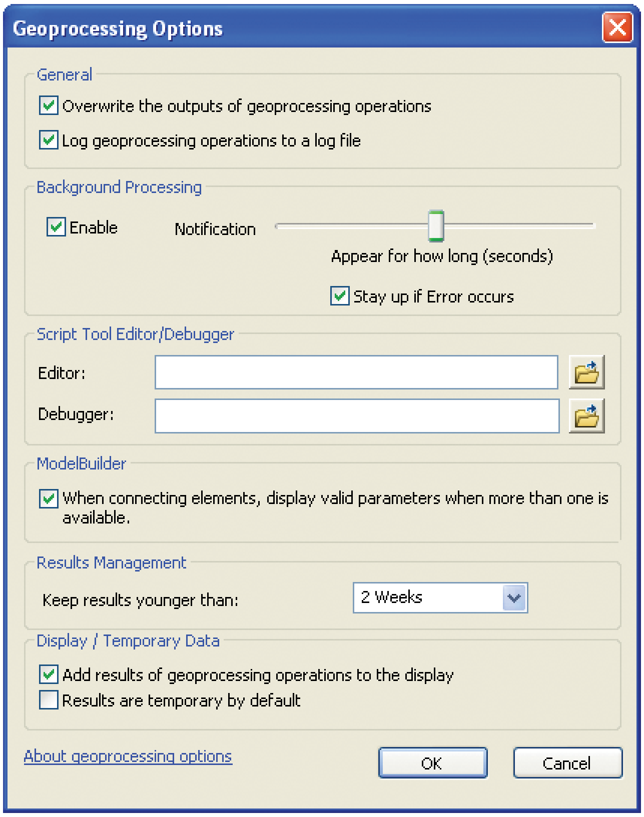
● Number of Features

11 **Click Cancel to close the tool dialog box without running the tool.** Now that the parameters are specified, the script needs to be modified to read these parameters.

## Select a script editor

To edit the code of the script, you can open a Python editor and browse to the location of the script file. However, the script can also be opened directly from the script tool. The default editor is IDLE. So, you will first change the default script editor from IDLE to PythonWin.

**1 On the ArcMap menu bar, click Geoprocessing > Geoprocessing Options.**



Geoprocessing Options

General

☑ Overwrite the outputs of geoprocessing operations

☑ Log geoprocessing operations to a log file

Background Processing

☑ Enable Notification

Appear for how long (seconds)

☑ Stay up if Error occurs

Script Tool Editor/Debugger

Editor:

Debugger:

ModelBuilder

☑ When connecting elements, display valid parameters when more than one is available.

Results Management

Keep results younger than: 2 Weeks

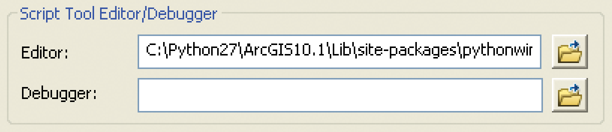
Display / Temporary Data

☑ Add results of geoprocessing operations to the display

□ Results are temporary by default

**2 In the Script Tool Editor/Debugger panel, click the Browse button to the right of the Editor box.**

**3 Browse to the location of the PythonWin application. Typically, the path to this application is: C:\Python27\ArcGIS10.1\Lib\site-packages\PythonWin\PythonWin.exe.**



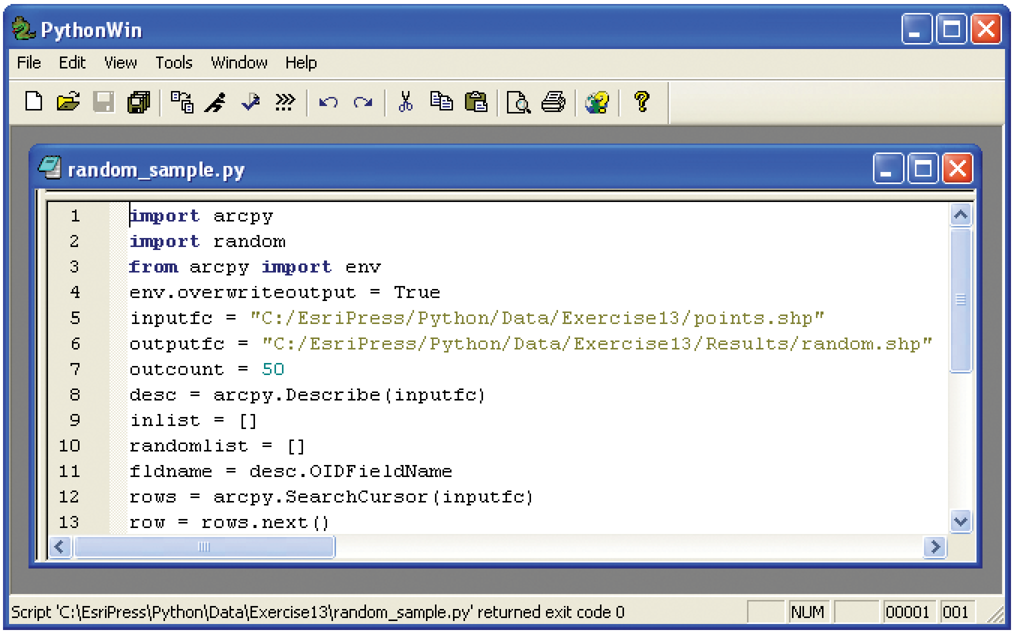
Script Tool Editor/Debugger

Editor: C:\Python27\ArcGIS10.1\Lib\site-packages\pythonwir

Debugger:

**4 Click OK to close the Geoprocessing Options dialog box.**

5 **In the Catalog window, navigate to the Random Tools toolbox. Right-click the Random Features tool and click Edit.** This opens the random_sample.py script in PythonWin.



```
1   import arcpy
2   import random
3   from arcpy import env
4   env.overwriteoutput = True
5   inputfc = "C:/EsriPress/Python/Data/Exercise13/points.shp"
6   outputfc = "C:/EsriPress/Python/Data/Exercise13/Results/random.shp"
7   outcount = 50
8   desc = arcpy.Describe(inputfc)
9   inlist = []
10  randomlist = []
11  fldname = desc.OIDFieldName
12  rows = arcpy.SearchCursor(inputfc)
13  row = rows.next()
```

Script 'C:\EsriPress\Python\Data\Exercise13\random_sample.py' returned exit code 0 00001 001

Next, you will make changes to the code, save the script, and run the tool to see if it works correctly.

## Edit the tool code to read the parameters

The random_sample.py script needs to be modified to read the parameters from the tool.

1 **With the script open in PythonWin, modify lines 5–7 as follows:**

```
inputfc = arcpy.GetParameterAsText(0)
outputfc = arcpy.GetParameterAsText(1)
outcount = int(arcpy.GetParameterAsText(2))
```

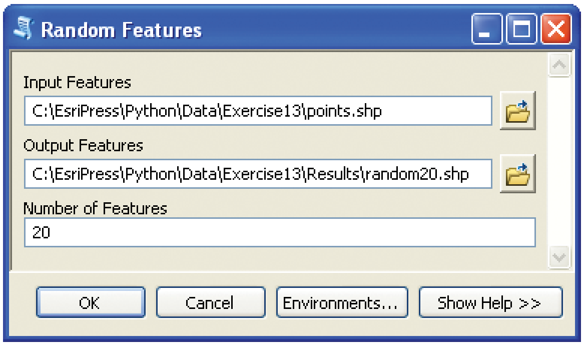
This effectively replaces the hard-coded values in the script with the parameters passed by the tool. The third parameter is received as a string but is converted to an integer.

2 **Save the script.**

3 **Close PythonWin.**

4 **Return to ArcMap. In the Catalog window, navigate to the Random Tools toolbox. Double-click the Random Features tool.**

5 **Fill in the tool parameters as shown in the figure.**



Random Features

Input Features
C:\EsriPress\Python\Data\Exercise13\points.shp

Output Features
C:\EsriPress\Python\Data\Exercise13\Results\random20.shp

Number of Features
20

6 **Click OK to run the tool.** The tool runs and a new feature class is created with a random selection of 20 point features.

## Challenge exercise

### Challenge 1

Make a copy of the random_sample.py script and call it **random_percent.py**. Modify the script so that the third parameter is a percentage of the number of input records as an integer between 1 and 100. Modify the script tool settings so that the input for this parameter is validated on the tool dialog box.