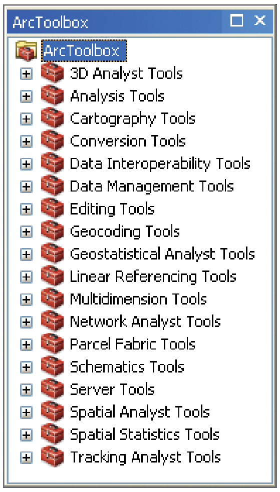
Exercise 2
Geoprocessing in ArcGIS

Examine toolboxes and tools

ArcGIS software contains hundreds of tools, organized in toolboxes and toolsets. There are two main ways to find the tools you need: search and browse. You will practice both.

1 Start ArcMap. On the Standard toolbar, click the ArcToolbox button . You can also open the ArcToolbox window from the menu bar by clicking Geoprocessing > ArcToolbox.

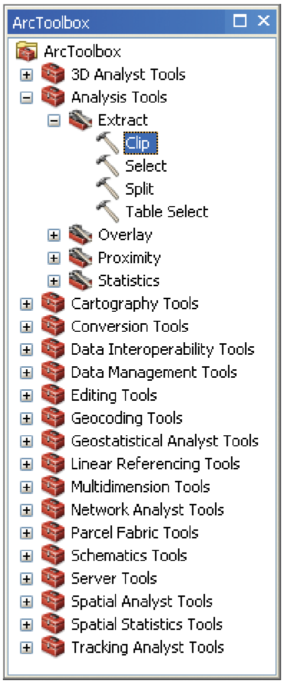


ArcToolbox

ArcToolbox
- ⊞ 3D Analyst Tools
- ⊞ Analysis Tools
- ⊞ Cartography Tools
- ⊞ Conversion Tools
- ⊞ Data Interoperability Tools
- ⊞ Data Management Tools
- ⊞ Editing Tools
- ⊞ Geocoding Tools
- ⊞ Geostatistical Analyst Tools
- ⊞ Linear Referencing Tools
- ⊞ Multidimension Tools
- ⊞ Network Analyst Tools
- ⊞ Parcel Fabric Tools
- ⊞ Schematics Tools
- ⊞ Server Tools
- ⊞ Spatial Analyst Tools
- ⊞ Spatial Statistics Tools
- ⊞ Tracking Analyst Tools

The ArcToolbox window shows a list of all the system toolboxes in ArcGIS and allows you to browse through an organized list of all the tools.

2 Expand the Analysis toolbox and then the Extract toolset to see the list of tools inside this toolset. →



ArcToolbox

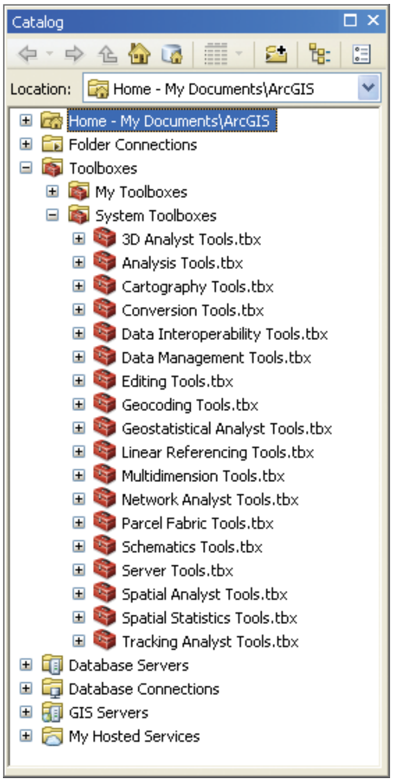
ArcToolbox
- ⊞ 3D Analyst Tools
- ⊟ Analysis Tools
  - ⊟ Extract
    - Clip
    - Select
    - Split
    - Table Select
  - ⊞ Overlay
  - ⊞ Proximity
  - ⊞ Statistics
- ⊞ Cartography Tools
- ⊞ Conversion Tools
- ⊞ Data Interoperability Tools
- ⊞ Data Management Tools
- ⊞ Editing Tools
- ⊞ Geocoding Tools
- ⊞ Geostatistical Analyst Tools
- ⊞ Linear Referencing Tools
- ⊞ Multidimension Tools
- ⊞ Network Analyst Tools
- ⊞ Parcel Fabric Tools
- ⊞ Schematics Tools
- ⊞ Server Tools
- ⊞ Spatial Analyst Tools
- ⊞ Spatial Statistics Tools
- ⊞ Tracking Analyst Tools

Finding the tool you want can be a bit cumbersome if you are not sure where to look, but with repetition you will start to remember where the tools you use most are located. A similar approach to browsing can be accomplished in the Catalog window.

3 Close the ArcToolbox window.

4 On the Standard toolbar, click the Catalog button .

5 Expand the Toolboxes entry, and you can see two folders: one for your custom toolboxes (My Toolboxes) and one for system toolboxes. The organization of the system toolboxes is the same as in the ArcToolbox window. ➔


Catalog

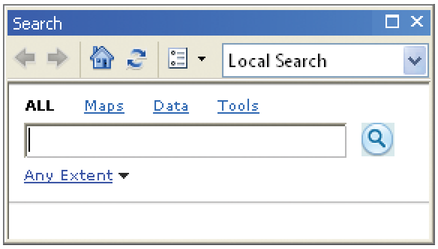
Location: Home - My Documents\ArcGIS

- Home - My Documents\ArcGIS
- Folder Connections
- Toolboxes
  - My Toolboxes
  - System Toolboxes
    - 3D Analyst Tools.tbx
    - Analysis Tools.tbx
    - Cartography Tools.tbx
    - Conversion Tools.tbx
    - Data Interoperability Tools.tbx
    - Data Management Tools.tbx
    - Editing Tools.tbx
    - Geocoding Tools.tbx
    - Geostatistical Analyst Tools.tbx
    - Linear Referencing Tools.tbx
    - Multidimension Tools.tbx
    - Network Analyst Tools.tbx
    - Parcel Fabric Tools.tbx
    - Schematics Tools.tbx
    - Server Tools.tbx
    - Spatial Analyst Tools.tbx
    - Spatial Statistics Tools.tbx
    - Tracking Analyst Tools.tbx
- Database Servers
- Database Connections
- GIS Servers
- My Hosted Services

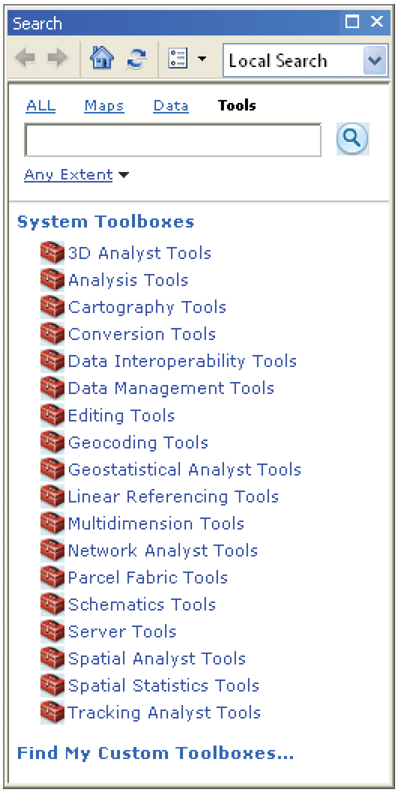
Now you can try using the Search window to find the geoprocessing tools.

6 Close the Catalog window.

7 On the Standard toolbar, click the Search window button . You can also open the Search window from the menu bar by clicking Geoprocessing > Search For Tools.



8 In the Search window, click the Tools hyperlink and increase the size of the Search window to see the complete list of toolboxes. These are exactly the same toolboxes you've seen in the ArcToolbox and Catalog windows. So you can browse to the tools you need in a similar fashion. ➔


Search

Local Search

ALL Maps Data **Tools**

Any Extent

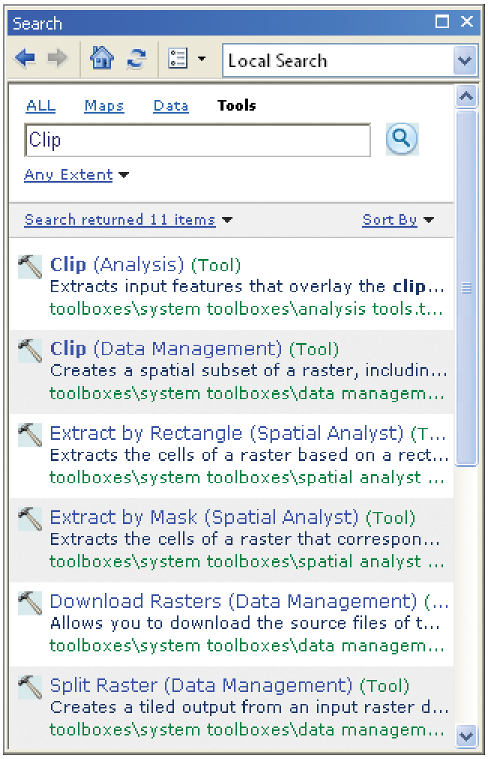
System Toolboxes

- 3D Analyst Tools
- Analysis Tools
- Cartography Tools
- Conversion Tools
- Data Interoperability Tools
- Data Management Tools
- Editing Tools
- Geocoding Tools
- Geostatistical Analyst Tools
- Linear Referencing Tools
- Multidimension Tools
- Network Analyst Tools
- Parcel Fabric Tools
- Schematics Tools
- Server Tools
- Spatial Analyst Tools
- Spatial Statistics Tools
- Tracking Analyst Tools

Find My Custom Toolboxes...

In addition to browsing, you can also search for tools by name.

9 Place your pointer in the Search box, type clip, **and click the Search button 🔍 .** →



Search

Local Search

ALL Maps Data **Tools**

Clip

Any Extent

Search returned 11 items Sort By

Clip (Analysis) (Tool)
Extracts input features that overlay the **clip**...
toolboxes\system toolboxes\analysis tools.t...

Clip (Data Management) (Tool)
Creates a spatial subset of a raster, includin...
toolboxes\system toolboxes\data managem...

Extract by Rectangle (Spatial Analyst) (T...
Extracts the cells of a raster based on a rect...
toolboxes\system toolboxes\spatial analyst ...

Extract by Mask (Spatial Analyst) (Tool)
Extracts the cells of a raster that correspon...
toolboxes\system toolboxes\spatial analyst ...

Download Rasters (Data Management) (...
Allows you to download the source files of t...
toolboxes\system toolboxes\data managem...

Split Raster (Data Management) (Tool)
Creates a tiled output from an input raster d...
toolboxes\system toolboxes\data managem...

The results panel shows the tools that have the search term "clip" in the name or in the description. In this case, notice that there is a Clip tool for vector data such as polylines and polygons in the Analysis toolbox and a Clip tool for raster data in the Data Management toolbox.

You can click the link to the Item description to get a more complete description of how the tools work—this is the same material as provided in the full ArcGIS Desktop Help files.

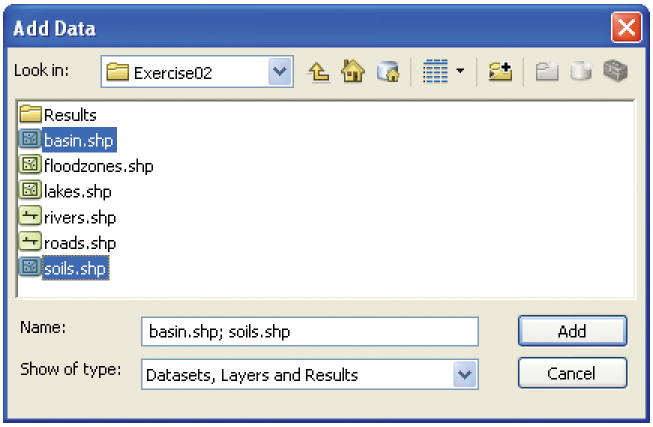
Following the link to the path, toolboxes\system toolboxes, opens the Catalog window. Double-clicking the tool name opens the tool dialog box.

10 Close the Search window.

Run a tool

Next, you will add some data so you can start using some of the geoprocessing tools.

1 With ArcMap open, on the Standard toolbar, click the Add Data button ✚ . On the Add Data dialog box, browse to the folder C:\EsriPress\Python\Data\Exercise02. Hold down CTRL and click basin.shp and soils.shp. Click Add. →



Add Data

Look in: Exercise02

Results
basin.shp
floodzones.shp
lakes.shp
rivers.shp
roads.shp
soils.shp

Name: basin.shp; soils.shp

Show of type: Datasets, Layers and Results

Add

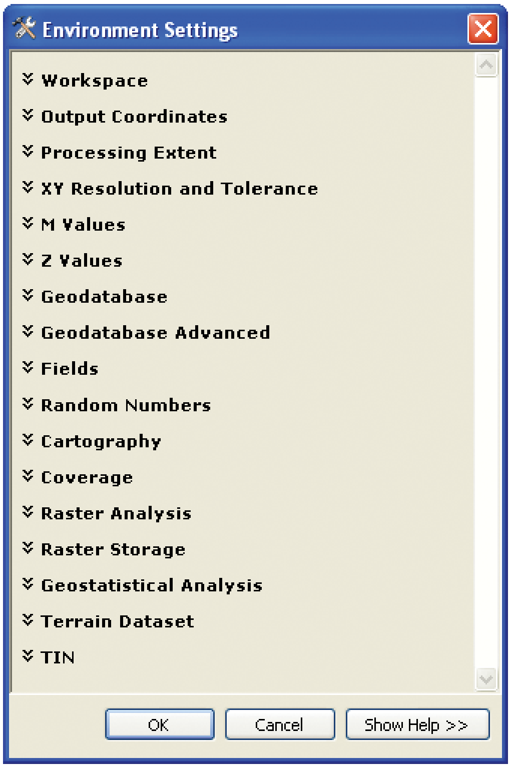
Cancel

This adds two shapefiles to your current data frame in ArcMap. Check to see that the extent of the soils shapefile is much larger than the basin shapefile. You will use the Clip tool to reduce the extent of the soils shapefile to match the basin shapefile.

Prior to running geoprocessing tools, you should set your environments. Environments are parameters that control how a tool is run, but these settings do not appear on the tool dialog box.

2 On the menu bar, click Geoprocessing > Environments. This brings up the Environment Settings dialog box. →

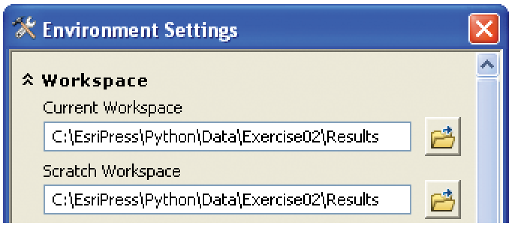
Notice the large number of categories, each with a number of options under it. In this case, you will set only the Workspace.



3 Click the Workspace entry to expand its options.

4 Click the Browse button next to Current Workspace. Navigate to the C:\EsriPress\Python\Data\Exercise02 folder and click Results. Click Add.

5 Repeat step 4 to set the Scratch Workspace.



Current Workspace: C:\EsriPress\Python\Data\Exercise02\Results

Scratch Workspace: C:\EsriPress\Python\Data\Exercise02\Results

By setting the workspace, the outputs of geoprocessing operations will now be saved to the C:\EsriPress\Python\Data\Exercise02\Results folder. Instead of specifying a folder, you can also specify an existing geodatabase. In this case, the current workspace and the scratch workspace are set to the same folder, but they can be set to different ones—this can be useful for separating intermediate results from final results in geoprocessing workflows that have many different steps generating dozens to hundreds of outputs. This is typically the case when using models in ModelBuilder.

6 Click OK to close the Environment Settings dialog box.

There are many other environment settings, some of which are introduced in later exercises.

Now you are ready to run a tool.

7 Open the ArcToolbox window. Expand the Analysis toolbox and then the Extract toolset and double-click the Clip tool.

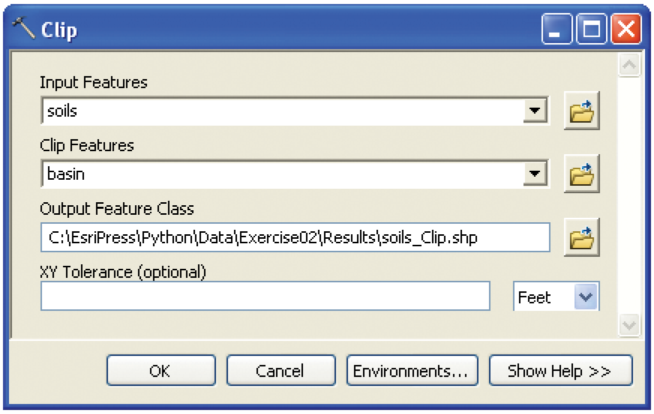
Next, you will specify the parameters of the Clip tool.

__Note:__ These instructions primarily use the ArcToolbox window to find and run tools, but you can also use the Catalog or Search windows.

8 For Input Features, select the soils shapefile from the drop-down list.

9 For Clip Features, select the basin shapefile from the drop-down list. Notice that the Output Feature Class is automatically populated. The path is determined by the environment settings (C:\EsriPress\Python\Data\Exercise02\Results) and the file name is based on the inputs and the name of the tool. You can use the Browse button to navigate to a different path. You can also change the path and output file name by typing in the text box.

10 Leave the XY Tolerance blank.



Clip

Input Features
soils

Clip Features
basin

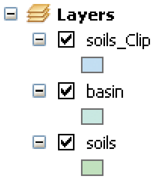
Output Feature Class
C:\EsriPress\Python\Data\Exercise02\Results\soils_Clip.shp

XY Tolerance (optional)

Feet

11 Click OK to run the tool. A progress bar that appears at the bottom of the document on the ArcGIS for Desktop application status bar shows that the tool is running. Once tool execution is complete, a small pop-up notification briefly appears in the notification area, at the far right of the taskbar.

The result of running the tool is a new shapefile called **soils_Clip**, which is added to the ArcMap table of contents.

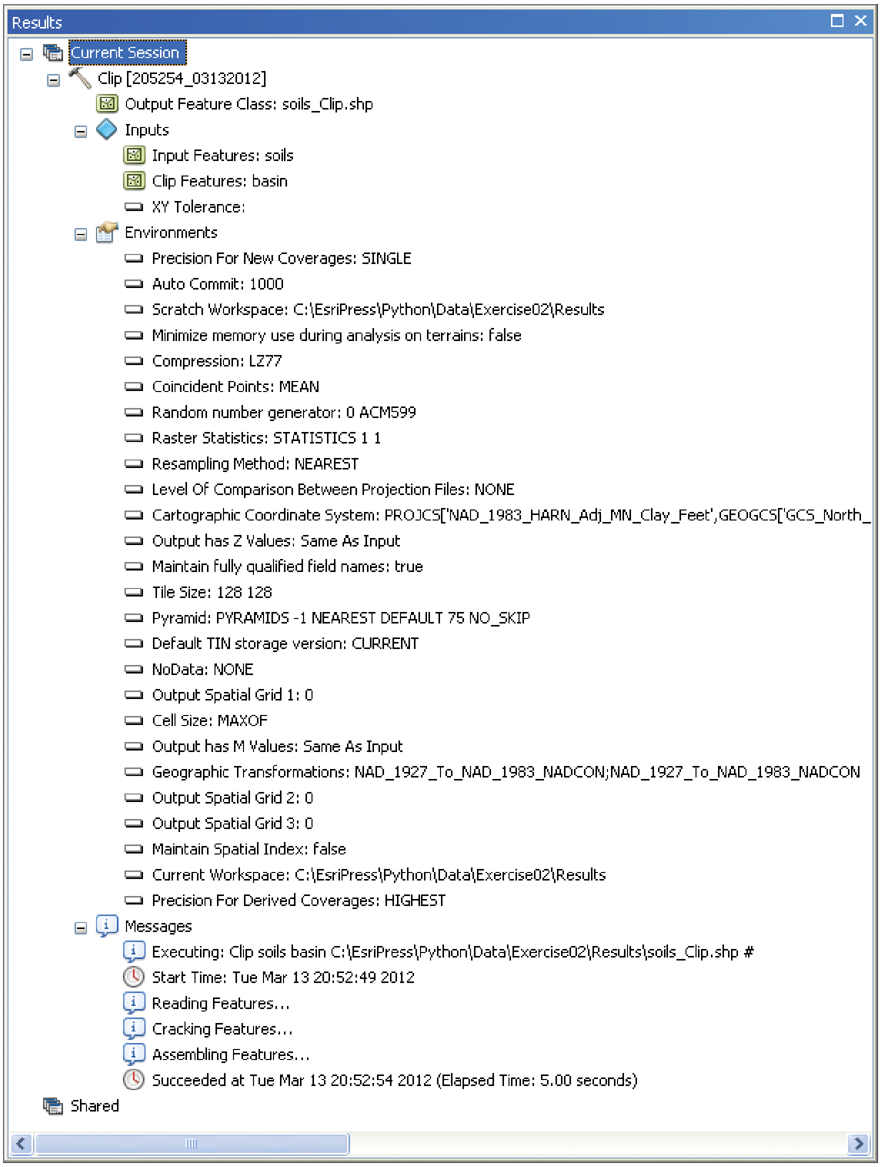


⊟ Layers
⊟ ☑ soils_Clip
⊟ ☑ basin
⊟ ☑ soils

You can now explore the soils_Clip shapefile to confirm the result. You can also review the execution of the tool by exploring the Results window.

12 On the ArcMap menu bar, click Geoprocessing > Results.

13 In the Results window, expand the Current Session entry as well as the entries inside Current Session.



```
Results
⊟ Current Session
    ⊟ Clip [205254_03132012]
          Output Feature Class: soils_Clip.shp
        ⊟ Inputs
              Input Features: soils
              Clip Features: basin
              XY Tolerance:
        ⊟ Environments
              Precision For New Coverages: SINGLE
              Auto Commit: 1000
              Scratch Workspace: C:\EsriPress\Python\Data\Exercise02\Results
              Minimize memory use during analysis on terrains: false
              Compression: LZ77
              Coincident Points: MEAN
              Random number generator: 0 ACM599
              Raster Statistics: STATISTICS 1 1
              Resampling Method: NEAREST
              Level Of Comparison Between Projection Files: NONE
              Cartographic Coordinate System: PROJCS['NAD_1983_HARN_Adj_MN_Clay_Feet',GEOGCS['GCS_North_
              Output has Z Values: Same As Input
              Maintain fully qualified field names: true
              Tile Size: 128 128
              Pyramid: PYRAMIDS -1 NEAREST DEFAULT 75 NO_SKIP
              Default TIN storage version: CURRENT
              NoData: NONE
              Output Spatial Grid 1: 0
              Cell Size: MAXOF
              Output has M Values: Same As Input
              Geographic Transformations: NAD_1927_To_NAD_1983_NADCON;NAD_1927_To_NAD_1983_NADCON
              Output Spatial Grid 2: 0
              Output Spatial Grid 3: 0
              Maintain Spatial Index: false
              Current Workspace: C:\EsriPress\Python\Data\Exercise02\Results
              Precision For Derived Coverages: HIGHEST
        ⊟ Messages
              Executing: Clip soils basin C:\EsriPress\Python\Data\Exercise02\Results\soils_Clip.shp #
              Start Time: Tue Mar 13 20:52:49 2012
              Reading Features...
              Cracking Features...
              Assembling Features...
              Succeeded at Tue Mar 13 20:52:54 2012 (Elapsed Time: 5.00 seconds)
  Shared
```

The Results window records a history of the geoprocessing operations, including all the tool parameters and the environment settings. The messages at the end demonstrate the steps that were part of running the tool and show when the run was completed.

14 Close the Results window.

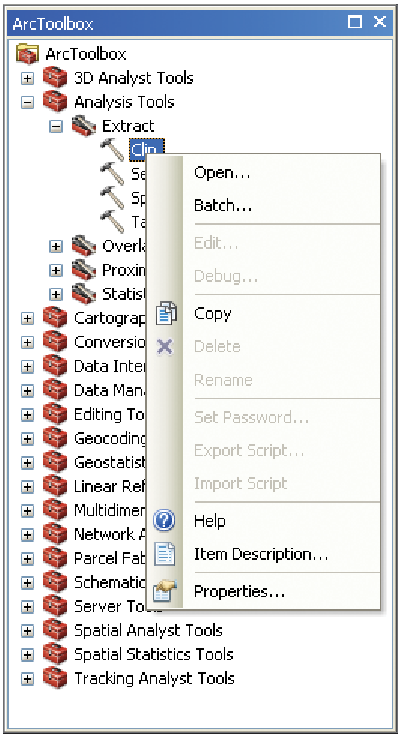
15 Right-click the soils_Clip layer in the ArcMap table of contents and click Remove.

Conduct batch processing

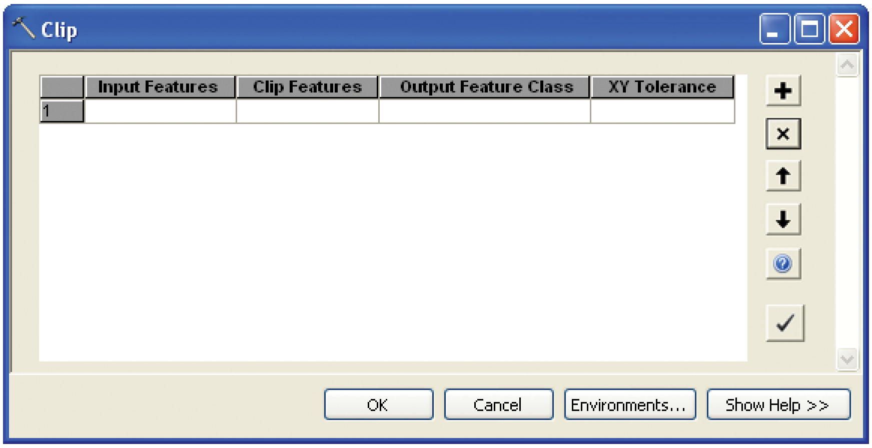
You will next use batch processing to clip the other layers.

1 Add all the other layers in the exercise 2 folder to the data frame: floodzones, lakes, rivers, and roads.

2 Right-click the Clip tool and click Batch.



This brings up the batch grid for the Clip tool.

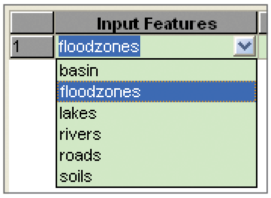


There are several ways to fill in the cells, as follows:

- Double-clicking a row number brings up the Clip tool dialog box for that row.
- Double-clicking in a cell brings up a dialog box for just that cell.
- Clicking on the right side of a cell opens a drop-down list of layers.
- Right-clicking in a cell and clicking Open brings up a dialog box for that cell.
- Right-clicking in a cell and clicking Browse brings up a dialog box for browsing.

Since the layers have already been added to the data frame, using the drop-down option in this case is quite efficient.

3 In the column Input Features, click on the right side of the cell in the first row to open the drop-down list and select the floodzones layer. →

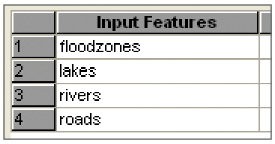


	Input Features
1	floodzones

4 Add a row to the batch grid by clicking the Add Row button + .

5 In the second row, use the drop-down arrow to select the lakes layer.

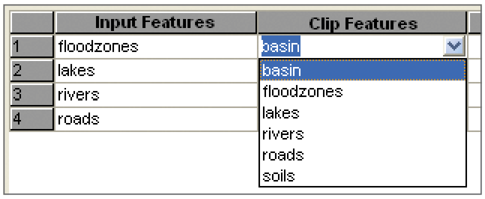
6 Keep adding rows for the remaining layers that need to be clipped so that the grid matches the example in the figure. →



	Input Features
1	floodzones
2	lakes
3	rivers
4	roads

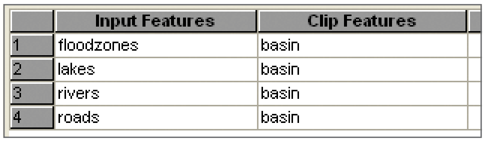
Now you can move on to the Clip Features column.

7 For Clip Features in the first row, use the drop-down list to select the basin layer. →



	Input Features	Clip Features
1	floodzones	basin
2	lakes	
3	rivers	
4	roads	

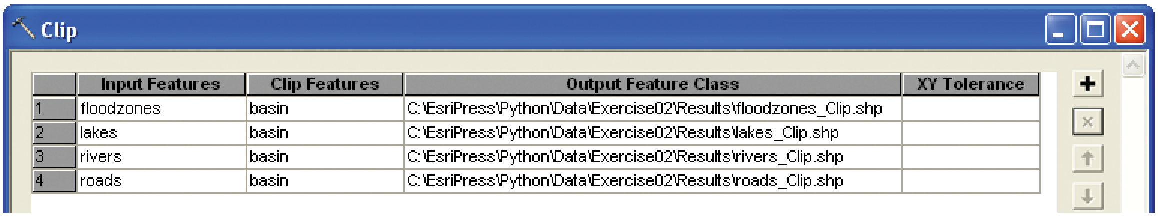
8 In the first Clip Features cell, right-click basin and click Fill. It populates the cells in the remaining rows with the same value—that is, the basin layer. →



	Input Features	Clip Features
1	floodzones	basin
2	lakes	basin
3	rivers	basin
4	roads	basin

The last set of parameters to fill in are for Output Feature Class.

9 Click the Check Values button. It validates the cell values in the batch grid, and in the case of Output Feature Class, populates the file names with defaults based on the current workspace.



Clip

	Input Features	Clip Features	Output Feature Class	XY Tolerance
1	floodzones	basin	C:\EsriPress\Python\Data\Exercise02\Results\floodzones_Clip.shp	
2	lakes	basin	C:\EsriPress\Python\Data\Exercise02\Results\lakes_Clip.shp	
3	rivers	basin	C:\EsriPress\Python\Data\Exercise02\Results\rivers_Clip.shp	
4	roads	basin	C:\EsriPress\Python\Data\Exercise02\Results\roads_Clip.shp	

You can also enter the file names one by one, but for filling in a large number of rows, the Check Values button provides a quick method. You can make changes to these values by modifying a single cell (for example, by double-clicking in the cell or right-clicking in the cell and clicking Open).

The batch tool is now ready to run.

10 Click OK. Once the tool runs, the four output feature classes are added to the ArcMap table of contents.

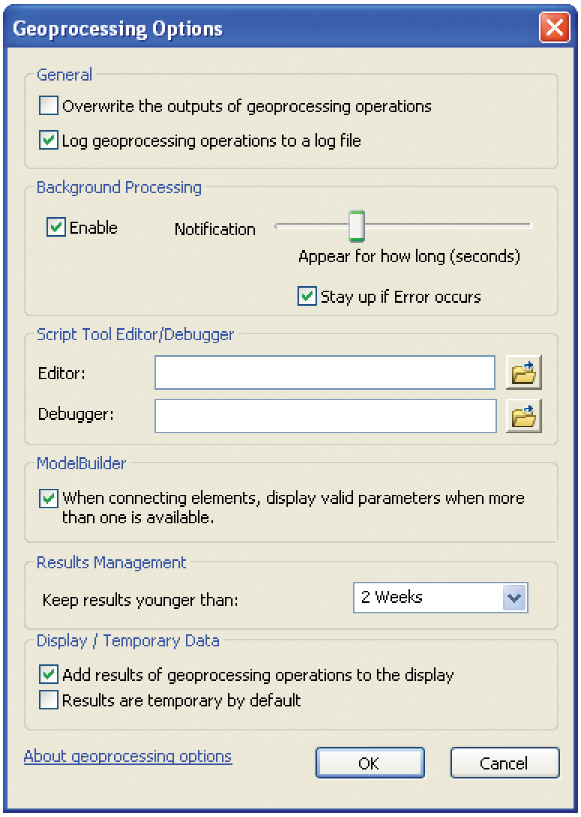
Set the geoprocessing options

Sometimes when you run geoprocessing operations, you encounter a situation where you might want to overwrite the existing data. For example, you may realize you made a mistake and want to run a tool again using the same name you already used for an output feature class. This is a common scenario when running models. By default, geoprocessing tools do not overwrite existing datasets, but this is a setting you can change.

First, you will modify the geoprocessing option to prevent overwriting files.

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

2 Under the General heading, clear the "Overwrite the outputs of geoprocessing operations" check box. ➜



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

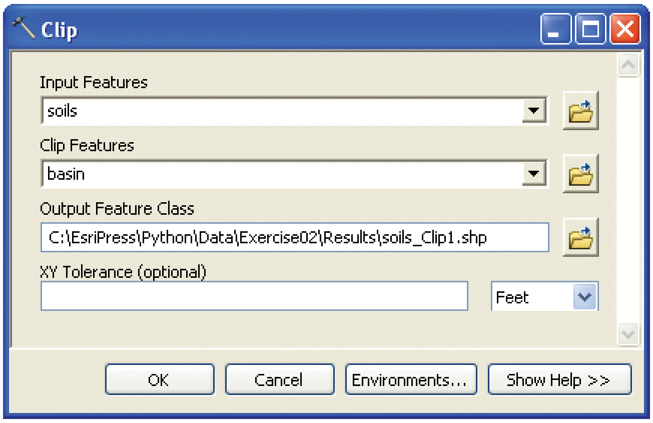
□ Results are temporary by default

3 Click OK.

Next, take a look at the following example.

4 Open the Clip tool dialog box.

5 For Input Features, select soils and for Clip Features, select basin.



Clip

Input Features: soils

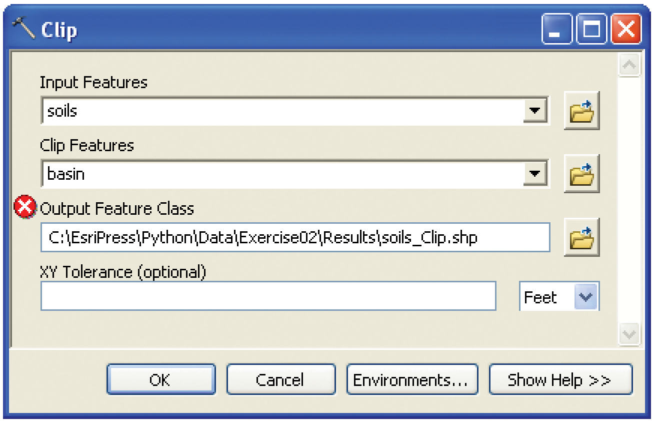
Clip Features: basin

Output Feature Class: C:\EsriPress\Python\Data\Exercise02\Results\soils_Clip1.shp

XY Tolerance (optional): Feet

Notice that, by default, in the naming of the output feature class, ArcMap recognizes that the soils_Clip feature class already exists and therefore the name soils_Clip1 is entered.

6 Change the name of Output Feature Class to soils_Clip**. Then click outside the input box for this parameter.** Since this name already exists, an error icon appears indicating that the tool will not run.



Clip

Input Features: soils

Clip Features: basin

Output Feature Class: C:\EsriPress\Python\Data\Exercise02\Results\soils_Clip.shp

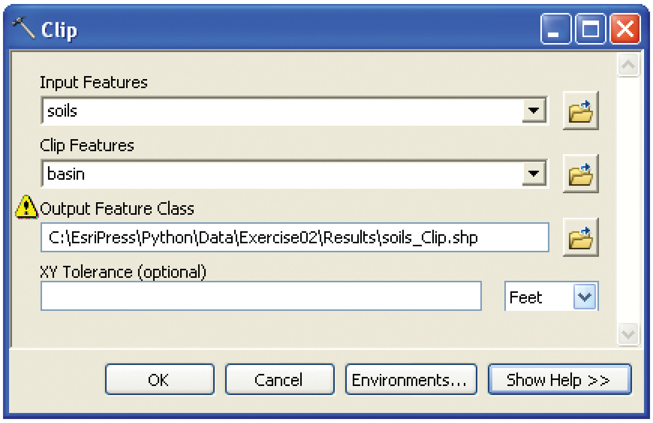
XY Tolerance (optional): Feet

Next, modify the geoprocessing option to allow for overwriting files.

7 On the ArcMap menu bar, click Geoprocessing > Geoprocessing Options.

8 Under the General heading, select the "Overwrite the outputs of geoprocessing operations" check box.

9 **Click OK.** On the Clip tool dialog box, the error icon has become a warning icon. The warning message says that the output feature class soils_Clip already exists, but this will not prevent the tool from running.



Clip

Input Features: soils

Clip Features: basin

⚠ Output Feature Class: C:\EsriPress\Python\Data\Exercise02\Results\soils_Clip.shp

XY Tolerance (optional): Feet

>>> TIP

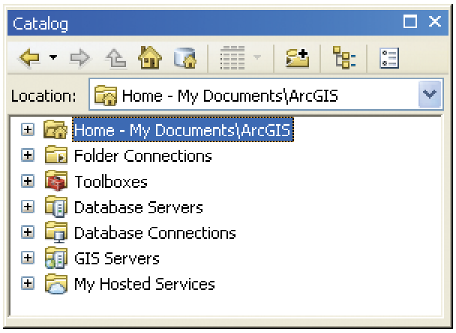
Although the changes to the geoprocessing options take effect immediately, you may need to retype the name of the output feature class on the Clip tool dialog box for the error icon to disappear.

10 **Click Cancel to close the Clip tool.**

Explore models and ModelBuilder

Models are one way to create a sequence of geoprocessing operations in ArcGIS. Like tools, models are organized in toolboxes and toolsets within ArcToolbox. Before creating a model then, you need to create a toolbox to store it.

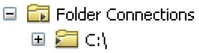
1 **On the ArcMap Standard toolbar, open the Catalog window by clicking the Catalog button.**



Catalog

Location: Home - My Documents\ArcGIS

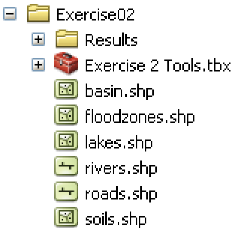
2 **Expand Folder Connections and make a connection to the C drive if this connection does not already exist.**



3 **Navigate to the exercise 2 folder.**

4 Right-click this folder and click New > Toolbox.

5 Name the toolbox Exercise 2 Tools.tbx**.**

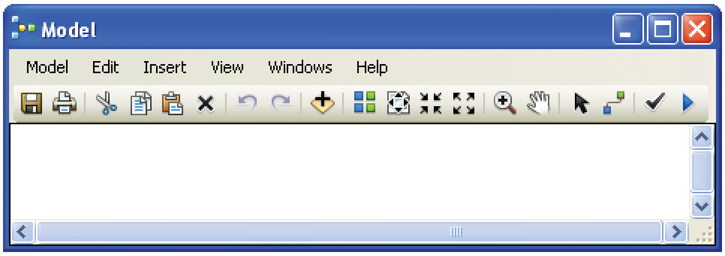


⊟ Exercise02
⊞ Results
⊞ Exercise 2 Tools.tbx
basin.shp
floodzones.shp
lakes.shp
rivers.shp
roads.shp
soils.shp

Note: *You can create your toolbox in any folder or in a geodatabase. The key is that a model needs to be saved within a toolbox, so you must create a toolbox first before creating the model.*

Now you are ready to create a model.

6 In the Catalog window, right-click the Exercise 2 Tools toolbox and click New > Model. This brings up the ModelBuilder interface and a new blank model.



First, you can give the model a name.

7 On the ModelBuilder menu bar, click Model > Model Properties.

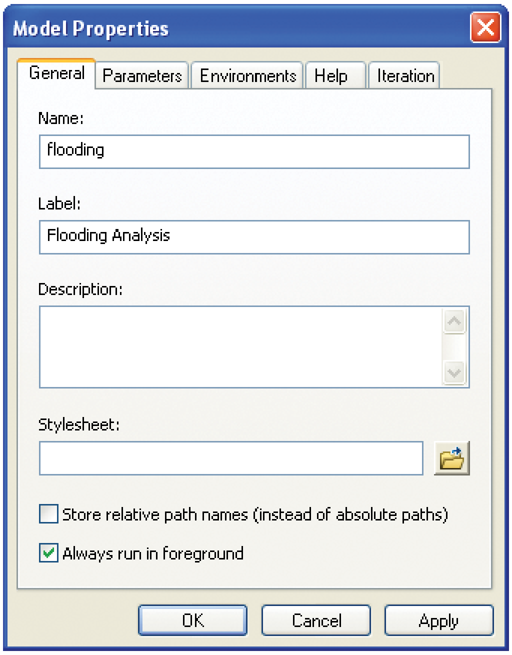
8 For Name, type flooding**. For Label, type** Flooding Analysis**.** →

Note: *The name indicates the actual name of the model. This name is used when calling the model from other tools in ArcGIS. The name cannot contain any spaces. The label indicates the label that will appear next to the model tool in the toolbox. The label can contain spaces.*

9 On the General tab, select the "Store relative path names" check box.

10 Click OK to close the Model Properties dialog box.

11 On the ModelBuilder toolbar, click the Save button.



Name: flooding

Label: Flooding Analysis

Description:

Stylesheet:

☐ Store relative path names (instead of absolute paths)

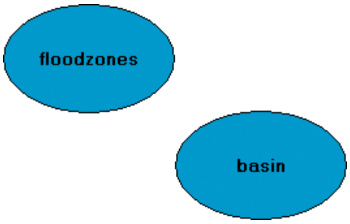
☑ Always run in foreground

Notice that this updates the label of the model in the Exercise02 folder and the heading of the model itself. You are now ready to start adding elements to the model. There are several ways to add data and tools to your model, as follows:

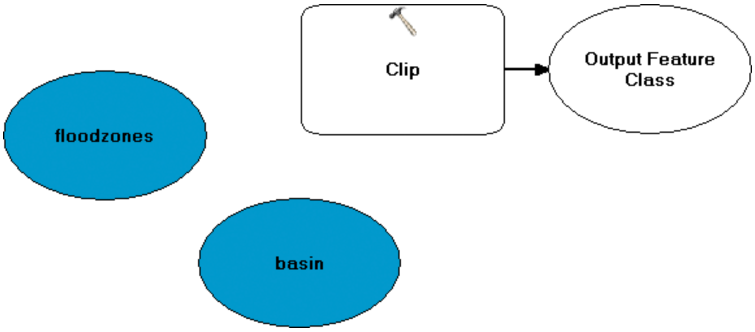
- Use the Add Data or Tool button on the ModelBuilder toolbar.
- Right-click in the model and click Add Data or Tool.
- Drag layers from the table of contents and tools from ArcToolbox into the model.

You will use the drag-and-drop method in the following steps, but the other methods are just as good.

12 Drag the basin and floodzones layers from the ArcMap table of contents into the model.



13 Drag the Clip tool from ArcToolbox into the model.



As the layers were added, their oval symbols were given a fill color (blue) because the file name for these data variables is specified. When the Clip tool was added, its rectangular symbol remained hollow because the tool's parameters have not been specified yet. Hollow symbols indicate that a model is not ready to run. In addition, by its very nature, the Clip tool produces an Output Feature Class, so this data variable is automatically added to the model, even though it is not pointing to an output feature class yet.

Connectors also need to be added to make the model ready to run. In this example, the layers basin and floodzones need to be connected to the Clip tool. There are two ways to create the appropriate connectors:

1. Use the Connect tool . On the toolbar, click this tool, and then click one element and drag the connector to the second element.

2. Open the tool dialog box and specify the tool's parameters.

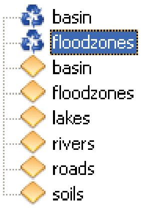
In the following steps, you will use the tool dialog box option.

14 In the model (not in ArcToolbox!), double-click the Clip tool to open the tool dialog box.

You can now specify the tool's parameters as you normally would for a tool.

15 For Input Features, click the drop-down arrow.

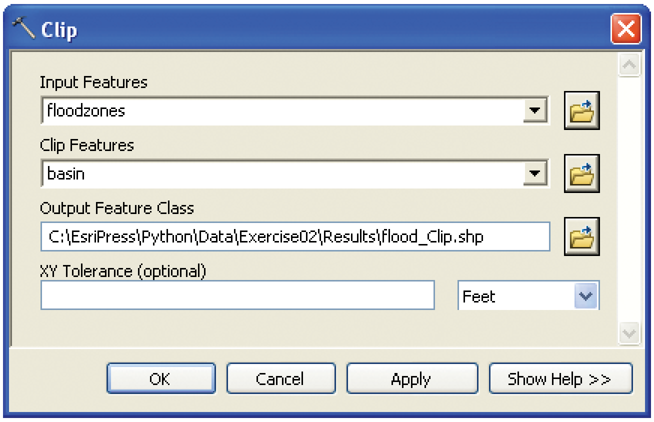
The drop-down list shows all the available layers in the ArcMap table of contents as well as the data variables already added to the model. This explains why floodzones and basin occur twice on the list. →



16 Click the floodzones data variable (the one with a blue symbol in front of it) for Input Features.

17 Click the basin data variable for Clip Features.

18 Set Output Feature Class to C:\EsriPress\Python\Data\Exercise02\Results\flood_Clip.shp**.**


Clip

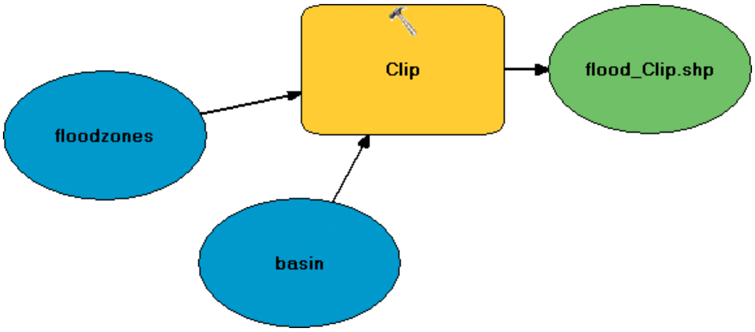
Input Features
floodzones

Clip Features
basin

Output Feature Class
C:\EsriPress\Python\Data\Exercise02\Results\flood_Clip.shp

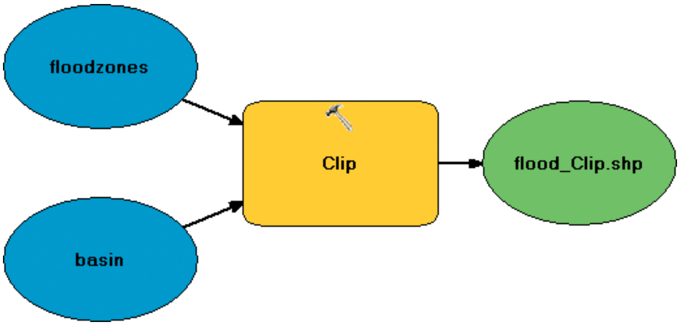
XY Tolerance (optional)
Feet

19 Click OK to close the tool dialog box. When you click OK to close the tool dialog box, the tool does not run as it would if you were using the tool in stand-alone mode—that is, outside a model. Instead, with the tool parameters specified, the appropriate connectors are created in the model. As a result, the symbol for the Clip tool in the model is now given a fill color (yellow) and so is the symbol for the output data variable (green). When all parameters are specified and all elements in the model have a fill color, the model is ready to run.



Before proceeding, try cleaning up the look of the model a bit.

20 On the ModelBuilder toolbar, first click the Auto Layout button and then the Full Extent button. This organizes the model elements into a consistent pattern. Although it has no effect on running the model, it makes it easier to follow the workflow in your model. As a general rule, after every few modifications or additions to a model, it is a good idea to use the Auto Layout and Full Extent buttons to reorganize the model elements.



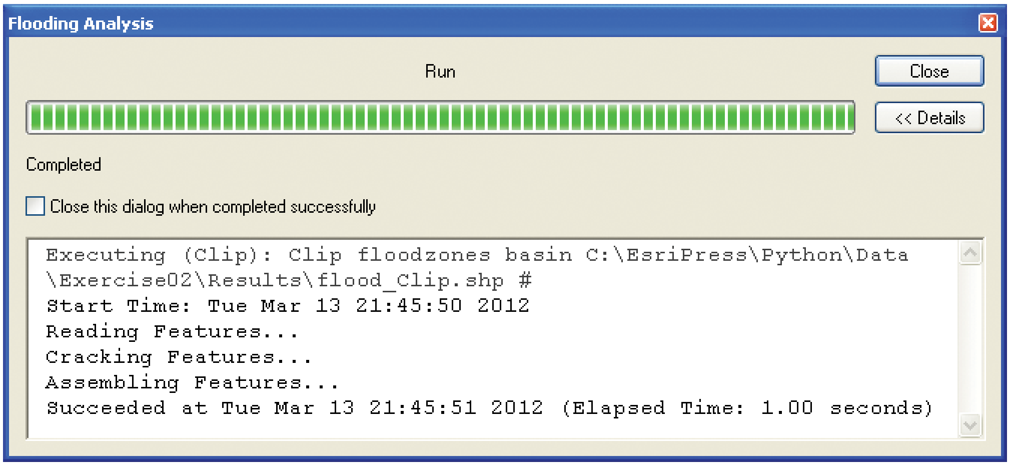
Run your model

Now your model is ready to run. There are several ways to run a model:

- You can click the Run button on the ModelBuilder toolbar to run the entire model.
- You can click Model > Run on the ModelBuilder menu bar to run the entire model.
- You can right-click a particular tool in the model and click Run to run just the selected tool.

Since there is only one tool in the current model, there is no difference between running the entire model or only a single tool, but this option becomes more relevant when your models become more complex.

1 **On the ModelBuilder toolbar, click the Run button to run the entire model.** A model progress dialog box appears that shows the progress and time elapsed in running the tools in the model. The messages are similar to those in the Results window when geoprocessing tools are running.



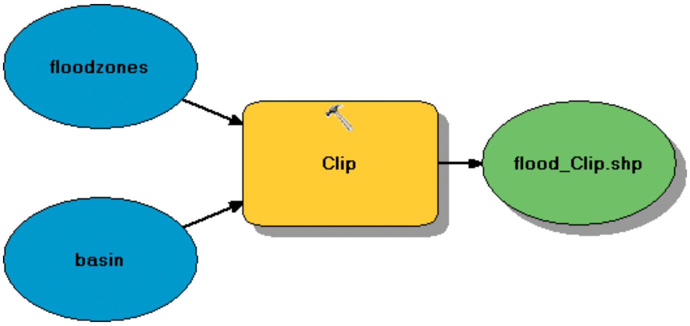
Completed

□ Close this dialog when completed successfully

```
Executing (Clip): Clip floodzones basin C:\EsriPress\Python\Data
\Exercise02\Results\flood_Clip.shp #
Start Time: Tue Mar 13 21:45:50 2012
Reading Features...
Cracking Features...
Assembling Features...
Succeeded at Tue Mar 13 21:45:51 2012 (Elapsed Time: 1.00 seconds)
```

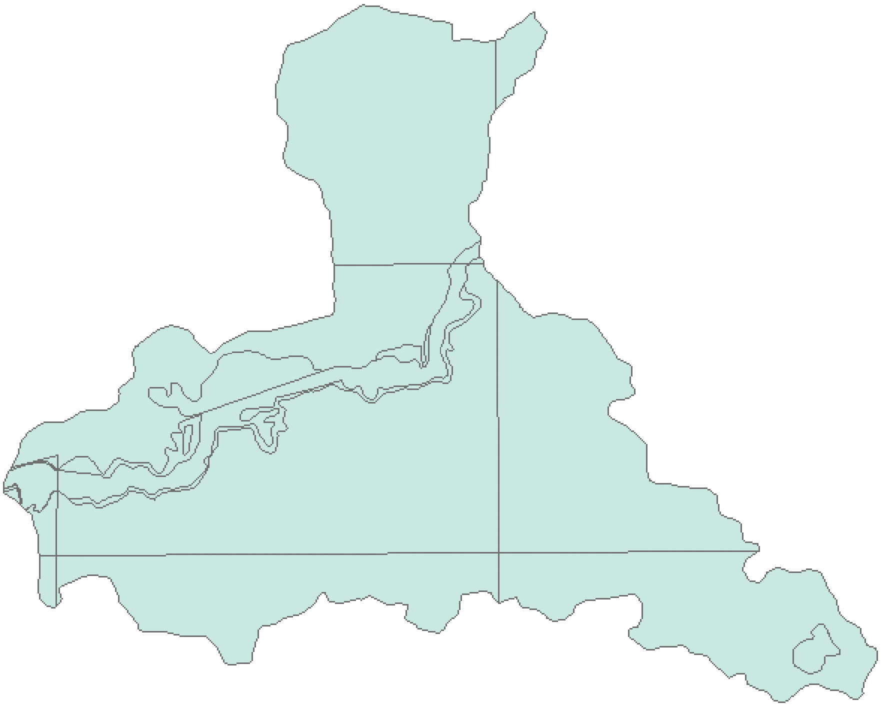
Note: *Since you are running the model from within ModelBuilder, the execution of the model is not recorded in the geoprocessing Results window, but on the model progress dialog box. If you were to save your model and close it, you could run it as a tool, and then the tool execution would be recorded in the Results window.*

2 **Click Close to close the Flooding Analysis dialog box.** When the model run is completed, the model elements (other than the input datasets) have a drop shadow to indicate that the tool has been run and the output datasets have been created—in this case, a shapefile.



Although the output shapefile flood_Clip.shp was created, it has not been added to the ArcMap table of contents. By default, ModelBuilder assumes the model outputs represent intermediate data.

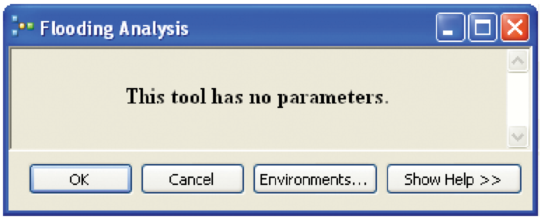
3 **Right-click the flood_Clip.shp element in the model and click Add To Display. Then right-click the flood_Clip layer and click Zoom To Layer.** You can now confirm that the flood_Clip layer has been added to the ArcMap table of contents and that it represents the clipped version of the floodzones layer.



4 On the ModelBuilder toolbar, click the Save button. Then on the menu bar, click Model > Close.

Next, you will run the model.

5 In the Exercise 2 Tools toolbox, double-click the Flooding Analysis model. This brings up the Flooding Analysis tool dialog box with the rather discouraging message, "This tool has no parameters." So what happened? Where is the model?



This tool has no parameters.

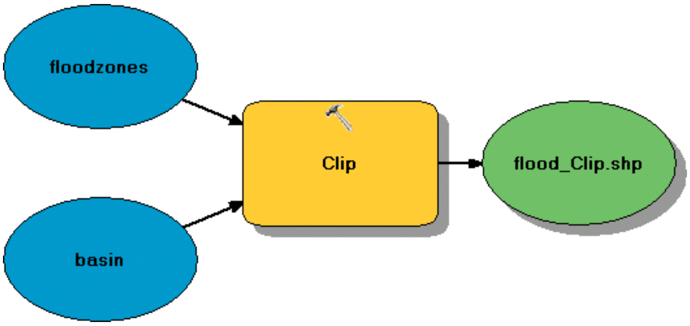
Models are tools, so by creating a model, you automatically create a tool. And tools have tool dialog boxes to specify parameters. However, in the ModelBuilder interface, you only created the model elements without indicating which elements should become parameters. In other words, the model is not yet fully ready to be used as a tool in which the user could specify the tool parameters.

Note: *Creating tool parameters is not covered here but is covered in chapter 13 on custom tools.*

So instead of running the model as a tool, you are going to go back into the model itself.

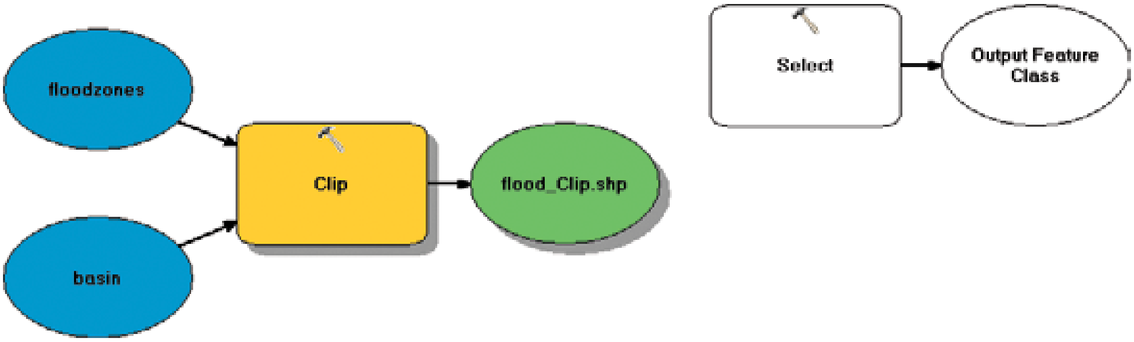
6 Click Cancel to close the Flooding Analysis tool dialog box.

7 In the Exercise 2 Tools toolbox, right-click the Flooding Analysis tool and click Edit. This brings you back into the ModelBuilder interface. Notice that the Clip tool and the output feature class still have drop shadows—the model has already been run and it remembers its processing state.



Next, you can add another step to the model. You may have noticed that the polygons in the floodzones layer cover the entire study area. This is how traditional flood maps are organized: polygons cover the entire study area but are coded as being inside or outside particular flood zone categories. You will add a tool to select just the polygons of interest.

8 In ArcToolbox, drag the Select tool from the Extract toolset in the Analysis toolbox into the model.



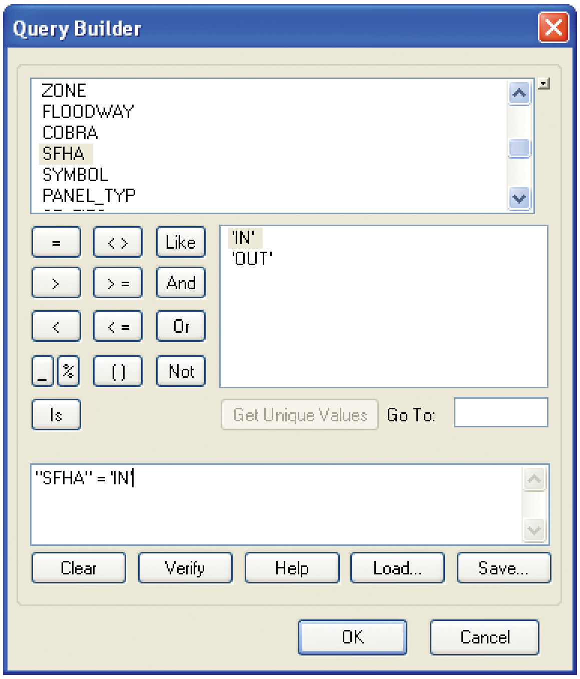
9 In the model (not in ArcToolbox), double-click the Select tool.

10 For Input Features, select flood_Clip.shp from the drop-down list.

11 Set the Output Feature Class to C:\EsriPress\Python\Data\Exercise02\Results\flooding.shp**.**

12 To create the Expression, click the SQL button .

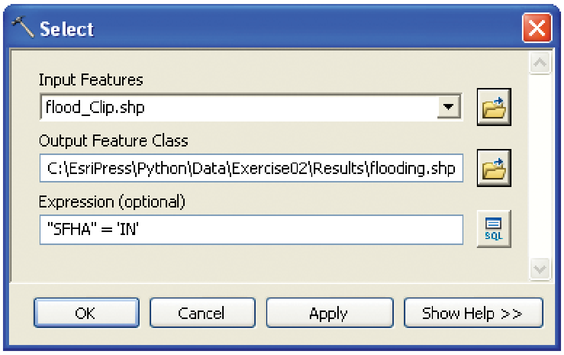
13 On the Query Builder dialog box, create the following expression: `"SFHA" = 'IN'` (SFHA stands for Special Flood Hazard Area). →



'IN'
'OUT'

"SFHA" = 'IN'

14 Click OK to close the Query Builder dialog box. The Select tool dialog box should now look like the example in the figure. →



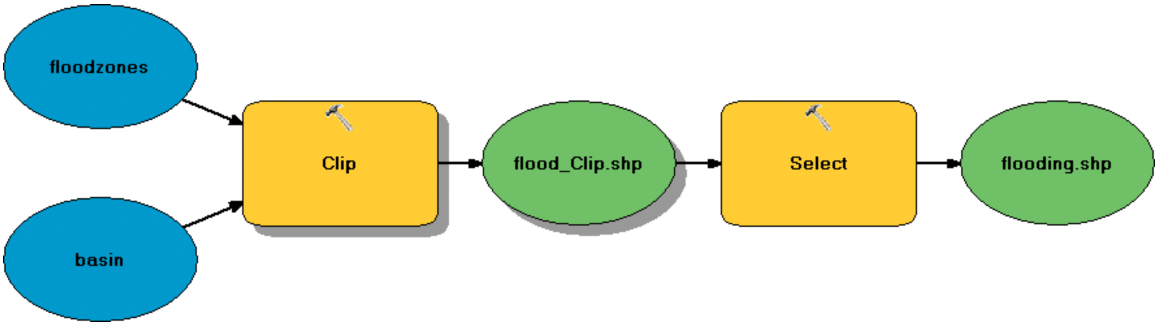
Input Features
flood_Clip.shp

Output Feature Class
C:\EsriPress\Python\Data\Exercise02\Results\flooding.shp

Expression (optional)
"SFHA" = 'IN'

15 Click OK to close the Select tool dialog box.

16 **On the ModelBuilder toolbar, first click the Auto Layout button and then the Full Extent button.** Your model should look like the example in the figure and is now ready to run.



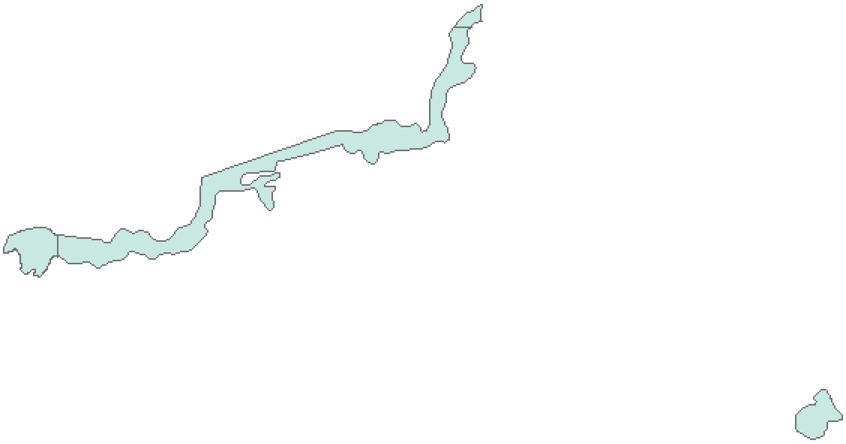
17 **Right-click the flood_Clip.shp element and turn off Add To Display.**

18 **Right-click the flooding.shp element and turn on Add To Display.**

19 **On the ModelBuilder toolbar, click the Run button to run the model.** Because the Clip tool was run previously, only the Select tool needs to be run for the model run to be complete.

20 **When the model run is complete, close the model progress dialog box.**

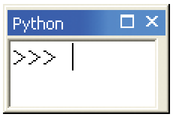
21 **Save and close the model.** The final result of the model is now added to the data frame.



Use scripting for geoprocessing

Scripting represents another way to carry out geoprocessing operations in ArcGIS. A basic Python script is similar to a model, except that it uses code instead of the visual programming language of ModelBuilder. Python is the preferred scripting language for working with ArcGIS, and Python code can be run directly in the Python window.

1 On the ArcMap Standard toolbar, click the Python window button .
This opens the Python window. The prompt (> > >) indicates that the Python window is ready to accept code.



```
>>>
```

To be able to run geoprocessing tools from Python, you first need to import the ArcPy site package, which you'll do next. Importing the ArcPy site package makes all the tools in the geoprocessing framework in ArcGIS available for Python scripting.

2 Following the prompt, type the following:

```
>>> import arcpy
```

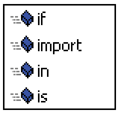
Within the Python window, ArcPy is automatically referenced, so the `import arcpy` statement is in fact not necessary to use the geoprocessing tools from within that window. However, code in the Python window can be converted to a script file (.py) and stand-alone scripts do need the `import arcpy` statement.

***Note:** Do not type the greater-than (>>>) symbols. They are shown here to indicate that Python code should be typed following the prompt. When a Python script is being written (which is discussed later in this section), the prompt is no longer used. In many programming environments, the prompt is referred to as a "command prompt," so you may see either term used in the documentation.*

Notice that the Python window provides prompts to assist in writing proper syntax. For example, when you start typing the letter *i*, a list is provided of the code elements that start with this letter. You can select the option you want by using the arrow keys to point to it, and then press the TAB key.

>>> TIP
Instead of using the prompts, you can also just keep typing. Even if you don't use the prompts, they can be very useful as reminders of the proper syntax.

```
>>> i
```



3 **After your first line of code (`import arcpy`), press ENTER.** Pressing ENTER brings up a new prompt at the next line. Remember that Python is an interpreted language, which means that in the Python window, a single line of code is run as soon as you press ENTER.

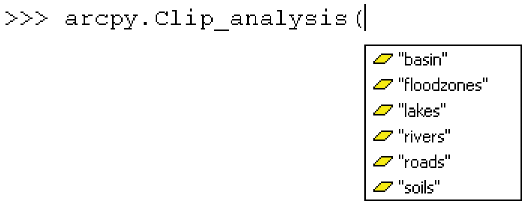
Now you are ready to run a geoprocessing tool.

4 **On the next line of code, enter the following, but do not press ENTER yet:**

```
    arcpy.Clip_analysis
```

This code calls the Clip tool. Python is case sensitive (for the most part), so be sure to type "Clip," not "clip." Calling the Clip tool is equivalent to opening the tool dialog box. The next step is to specify the tool's parameters, as if you were filling out the tool dialog box. As you start typing, the prompts will be helpful to ensure that you use the proper syntax.

When you type an opening left paren [(] after the Clip tool, a drop-down list appears, containing all the layers from the ArcMap table of contents.



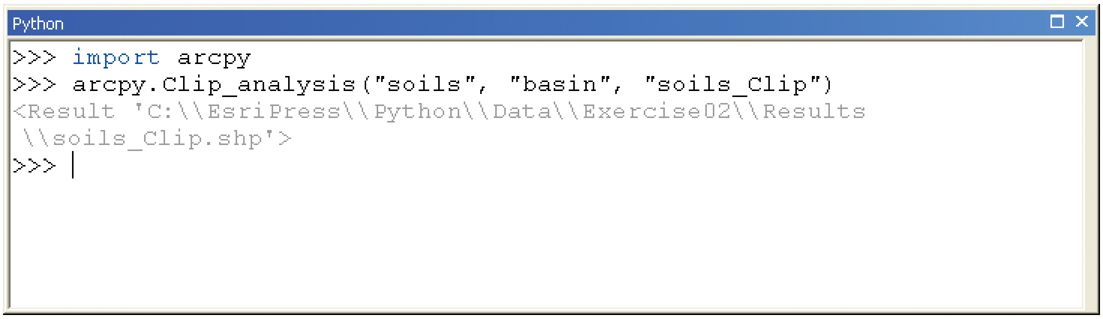
```
>>> arcpy.Clip_analysis(
```

5 **Complete the following line of code:**

```
    arcpy.Clip_analysis("soils", "basin", "soils_Clip")
```

The required tool parameters are listed inside the parens. The optional XY Tolerance is not included, which means that, just like a tool dialog box, the default value will be used.

6 **Press ENTER to run the line of code.** Similar to when a tool is run from ArcToolbox, when background processing is enabled, a progress bar appears on the ArcMap status bar, to show that the tool is running. Once the tool execution is complete, a small pop-up notification appears in the notification area, at the far right of the taskbar. The output feature class is added to the data frame, and the result is printed in the Python window.



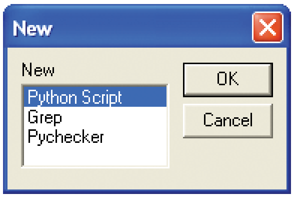
```
>>> import arcpy
>>> arcpy.Clip_analysis("soils", "basin", "soils_Clip")
<Result 'C:\\EsriPress\\Python\\Data\\Exercise02\\Results
 \\soils_Clip.shp'>
>>> |
```

The use of Python code in the Python window is covered in more detail in chapter 3. For now, it is important to remember that you can run geoprocessing tools directly from the Python window. Lines of code are run immediately, and the Python window is highly integrated with the ArcGIS interface.

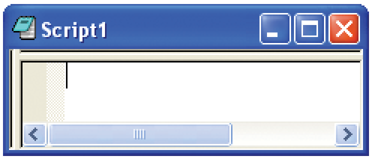
In addition to working with Python code in the Python window, you can write and run code in a Python editor. You will use the PythonWin editor in the next set of steps to create a simple script.

7 **On the taskbar, click the Start button, and then, on the Start menu, click Programs > Python 2.7 > PythonWin.** This brings up the PythonWin application.

8 **On the PythonWin menu bar, click File > New. On the New dialog box, click Python Script and click OK.**



This brings up a new script window.



9 On the menu bar, click File > Save As and save the script as my_clip.py **to the Results folder for exercise 2 (C:\EsriPress\Python\Data\Exercise02\Results\my_clip.py). Python script files are simply text files that have the .py extension.** There is no prompt (> > >) in the script window. Python code in a script is not run until the script is run. So you can enter multiple lines of code before you run the script.

10 Enter the following code in the my_clip script window:

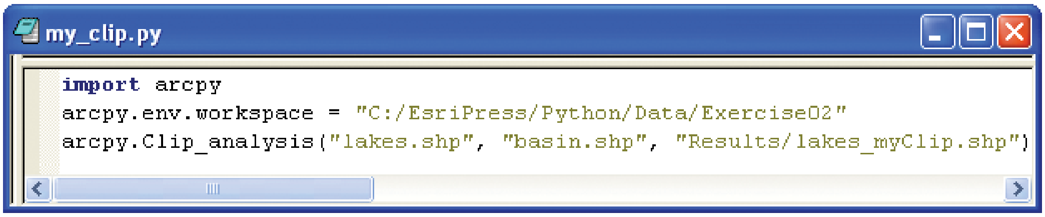
```
    import arcpy
    arcpy.env.workspace = "C:/EsriPress/Python/Data/Exercise02"
    arcpy.Clip_analysis("lakes.shp","basin.shp", "results/lakes_myClip. ➜
➜   shp")
```

Note: This book uses an arrow symbol ➜ to indicate long lines of code that appear all on one line in Python.

The line of code that starts with `arcpy.env.workspace` is equivalent to setting the workspace on the Environment Settings dialog box. This syntax is covered in more detail in the following chapters.

Note: The workspace needs to be set in the Python script, even though the environment settings have been set in the geoprocessing framework in ArcMap (that is, in ArcToolbox). A stand-alone Python script does not inherit the environment settings of ArcGIS for Desktop applications.

Your script window should now look like the example in the figure.



my_clip.py

```
import arcpy
arcpy.env.workspace = "C:/EsriPress/Python/Data/Exercise02"
arcpy.Clip_analysis("lakes.shp", "basin.shp", "Results/lakes_myClip.shp")
```

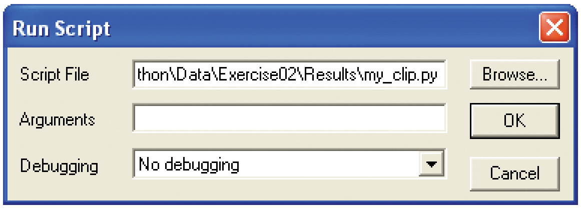
11 On the PythonWin menu bar, click File > Save to save the script.

>>> TIP

After subsequent edits, click the Run button to save the script automatically.

12 On the toolbar, click the Run button.

13 On the Run Script dialog box, leave the default settings (No debugging). Click OK. ➜



Run Script

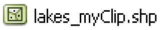
Script File: thon\Data\Exercise02\Results\my_clip.py

Arguments:

Debugging: No debugging

The script now runs. Upon execution of the script, it does not initially appear as if much has happened. Because you are running a stand-alone script, the output is not automatically added to a data frame in ArcMap. In fact, ArcMap doesn't need to be open for a script to run.

14 In ArcMap, open the Catalog window, navigate to the Results folder for exercise 2, and confirm that the lakes_myClip shapefile was created.


lakes_myClip.shp

The Python script accomplishes the same task as the Python code in the Python window: in both places, the Clip tool runs and creates a new dataset—in this case, a shapefile. However, there are a few differences:

- The Python window inherits the environments of the geoprocessing framework in ArcMap, but in the stand-alone Python script, the environments need to be set.

- The Python script can run without having any ArcGIS for Desktop applications open, whereas the Python window is an integral part of ArcGIS for Desktop applications.

- Code in the Python window is run line by line, whereas the standalone Python script is run in its entirety.

These differences between running Python scripts and running code in the Python window are revisited in later chapters.

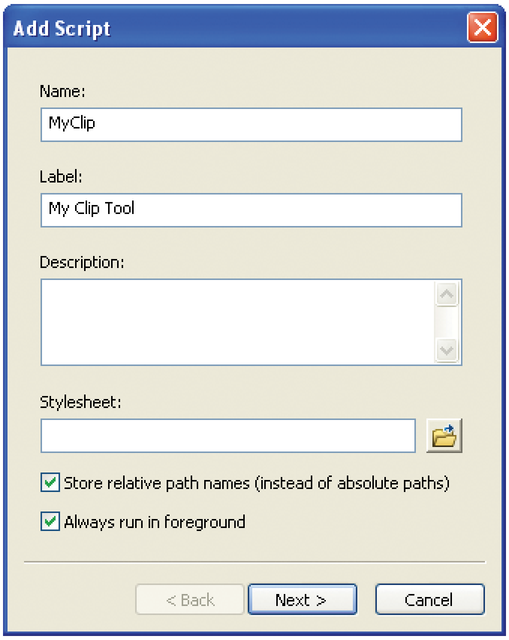
Use scripts as tools

The Python window provides a flexible environment for testing snippets of Python code, but more complex code is typically saved to a script. Python scripts can be integrated into the ArcGIS environment by adding them as tools.

1 In ArcMap, in the Catalog window, right-click the Exercise 2 Tools toolbox in the Exercise02 folder and click Add > Script.

2 On the Add Script dialog box, type MyClip **for Name and** My Clip Tool **for Label.**

3 Select the "Store relative path names" and "Always run in foreground" check boxes.



Add Script

Name: MyClip

Label: My Clip Tool

Description:

Stylesheet:

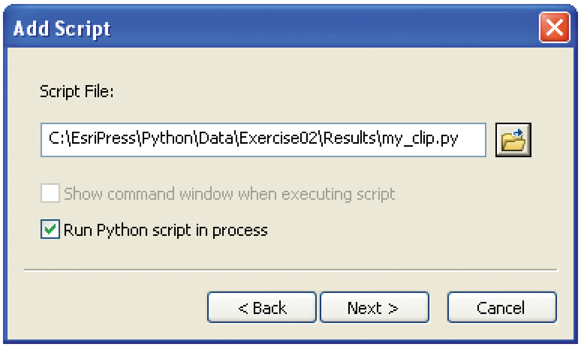
☑ Store relative path names (instead of absolute paths)

☑ Always run in foreground

4 Click Next.

On the next dialog box, you can select the script file that will be attached to the tool.

5 Click the Browse button and navigate to the Results folder for exercise 2. Select the my_clip.py script file.



Add Script

Script File: C:\EsriPress\Python\Data\Exercise02\Results\my_clip.py

□ Show command window when executing script

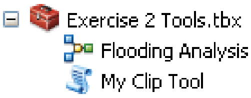
☑ Run Python script in process

6 Click Next.

The next dialog box allows you to specify tool parameters to be displayed in the tool dialog box. You can skip this for now because the simple script you are using contains hard-coded parameters, with the values already in place, instead of user-specified inputs. Creating script parameters for use on a tool dialog box is covered in chapter 13.

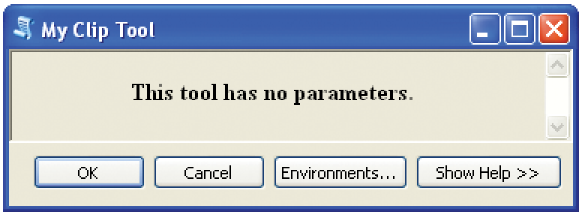
7 Leave the list of parameters blank and click Finish.

This adds a script tool to the toolbox. You can now run the script as a tool.



8 To test the script, in the Catalog window, navigate to the Results folder for exercise 2 and delete the lakes_myClip shapefile.

9 Double-click the My Clip Tool. The tool has no parameters because none were created when the script tool was set up. However, the script will run fine with the hard-coded parameters.



This tool has no parameters.

10 Click OK to run the tool. When the tool execution is complete, close the My Clip Tool progress dialog box.

11 In the Catalog window, right-click the Results folder for exercise 2 and click Refresh.

12 Confirm that the lakes_myClip shapefile was created.

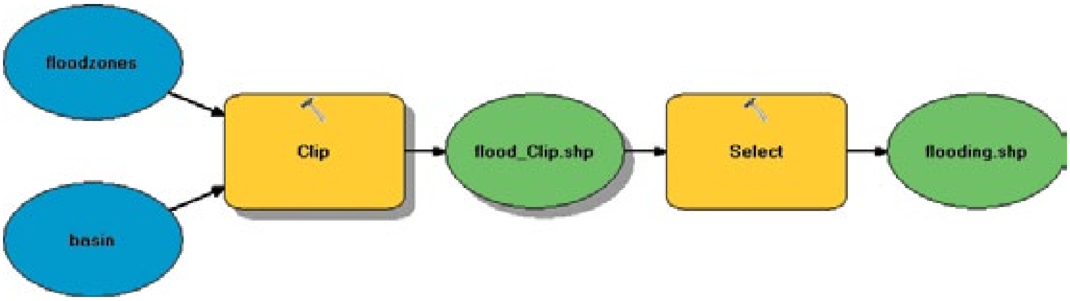
Note: *Because the script now runs as a tool, it inherits the environment settings and geoprocessing options of the current ArcMap document. For example, if you were to run the script tool again without first deleting the lakes_myClip shapefile, the tool would run fine and overwrite the existing dataset. This is because the "Overwrite the outputs of geoprocessing operations" check box was selected under Geoprocessing Options.*

As you have just seen, creating a script tool is fairly easy. However, there is much more to creating robust script tools, including setting tool parameters to obtain user input, validating input parameters, and error handling, to name a few. These topics are covered in chapter 13, after you have had more exposure to Python syntax and writing Python scripts.

Convert a model to a script

Another way to learn about Python scripting is to export a model to a script. This allows you to see what a logical sequence of geoprocessing operations looks like in Python.

For starters, you can revisit the Flooding Analysis model created earlier, as shown in the example in the figure.



You will now convert this model to a script.

1 **In the Catalog window, right-click the Flooding Analysis model in the Exercise 2 Tools toolbox in the Exercise02 folder and click Edit.**

2 **On the ModelBuilder menu bar, click Model > Export > To Python Script.**

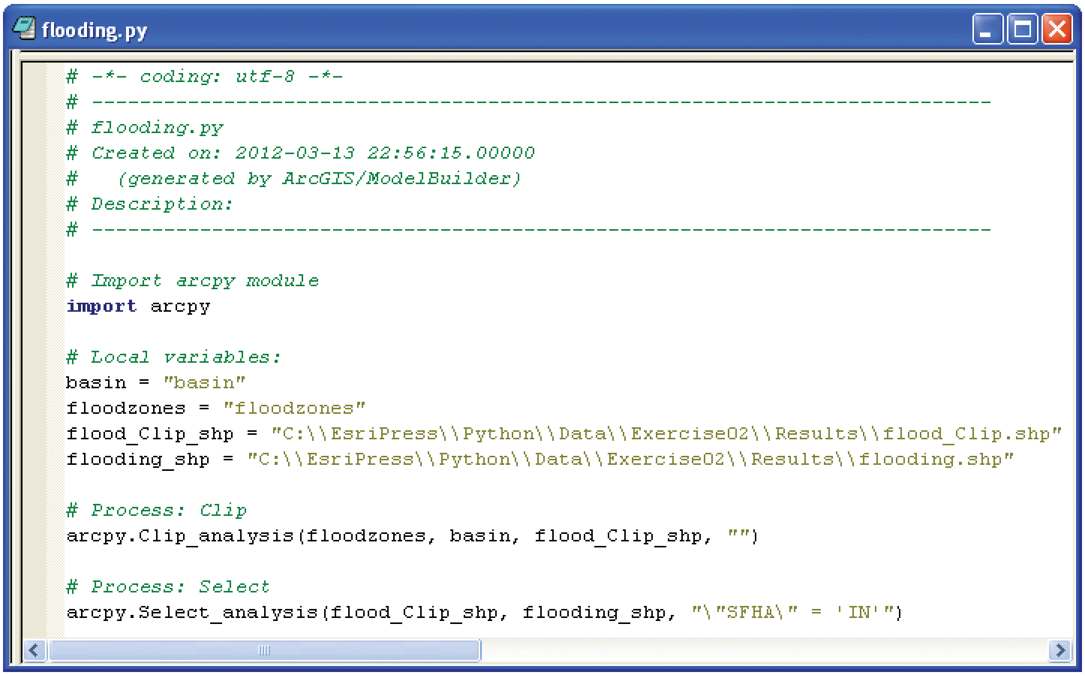
3 **On the Save As dialog box, save the script file as** flooding.py **to the Results folder for exercise 2.**

4 **Close the model.**

5 **Return to the PythonWin application. On the Standard toolbar, click the Open button 📂 .**

6 **On the Open dialog box, browse to the Results folder for exercise 2, click the flooding.py script, and click Open.**

7 Review the contents of the script.



```
# -*- coding: utf-8 -*-
# ---------------------------------------------------------------------------
# flooding.py
# Created on: 2012-03-13 22:56:15.00000
#   (generated by ArcGIS/ModelBuilder)
# Description:
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Local variables:
basin = "basin"
floodzones = "floodzones"
flood_Clip_shp = "C:\\EsriPress\\Python\\Data\\Exercise02\\Results\\flood_Clip.shp"
flooding_shp = "C:\\EsriPress\\Python\\Data\\Exercise02\\Results\\flooding.shp"

# Process: Clip
arcpy.Clip_analysis(floodzones, basin, flood_Clip_shp, "")

# Process: Select
arcpy.Select_analysis(flood_Clip_shp, flooding_shp, "\"SFHA\" = 'IN'")
```

Don't worry for now about being able to understand everything in the script. However, you should be able to recognize each of the model elements, including the data variables floodzones and basin, as well as the Clip and Select tools.

8 Close PythonWin.

9 Close ArcMap. There is no need to save your map document.

In this exercise, you have learned how to run geoprocessing tools and control how they are run by using tool parameters and environment settings. You have created a model using the ModelBuilder interface. You have also run Python code in the Python window as well as running stand-alone Python scripts and script tools from within an ArcGIS for Desktop application.

Challenge exercise

Challenge 1

Create a new model called **Soil Analysis** that accomplishes the following:

1. Clips the soils layer using the basin layer

2. From the clipped version of the soils layer, selects the features that are "Not prime farmland" (field FARMLNDCL)

Convert the model to a script called **soil.py**.