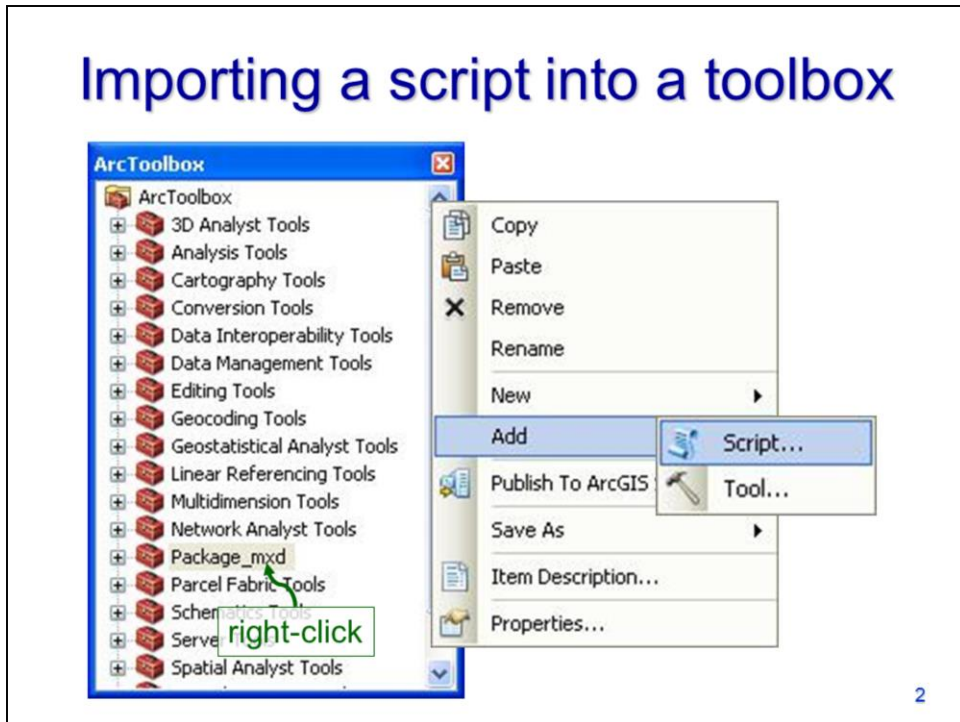# Importing the script into ArcToolbox

Creating ArcScripts

This video will discuss how to import a script into ArcToolbox.

Importing a script into a toolbox

In ArcToolbox, a script can be added to a new or existing toolbox. Note that scripts cannot be added to the built-in toolboxes because they are locked for editing.

To add a script to a toolbox which can be edited, right-click on the toolbox and click **Add**, and then **Script**.

Import wizard – page 1
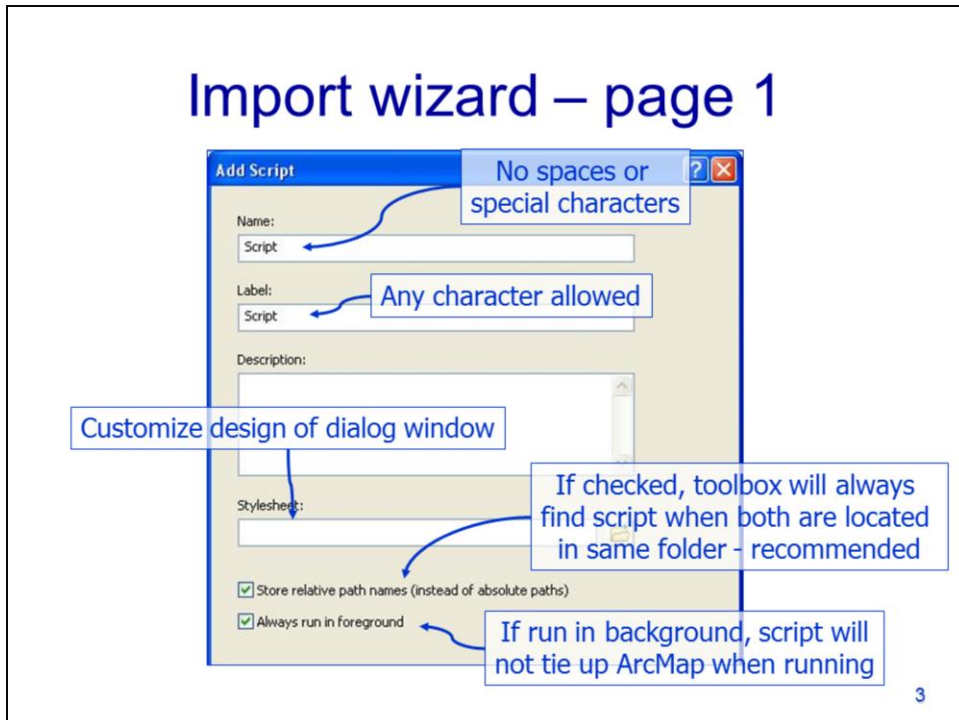
An import wizard will go through the process of importing the script. In the first window…
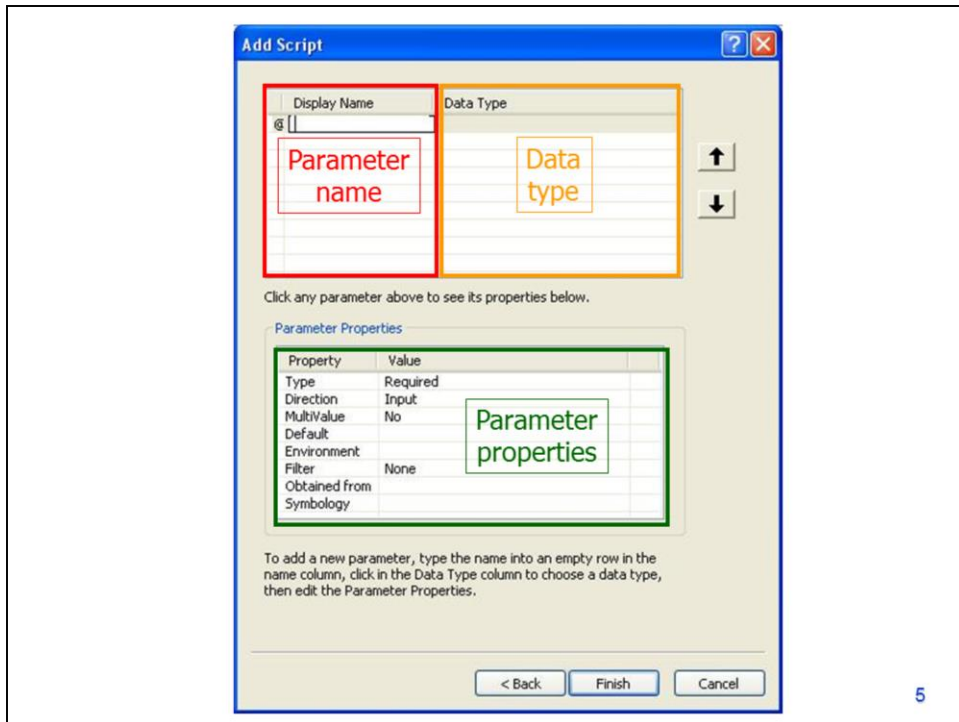
- Type in the name of the script as it will be referred to by other scripts. The name can only contain letters and numbers – no spaces or other characters can be used.
- The label is the tool name as it will appear in ArcToolbox. Spaces and special characters are allowed.
- The description is optional and provides an overview of the tool's purpose.
- The **stylesheet** allows you to customize the dialog window. Stylesheets are programmed in html – refer to ArcGIS's documentation for more information.
- The **Store relative path names** option will allow the toolbox to always find the script when they are located in the same folder – this is option is recommended if you expect to share or move the tool.
- Checking the option to **Always run in the foreground** will make script use all of ArcGIS's resources when running. When a script is run in the background, you are able to perform other tasks in ArcGIS while the tool is running. However, running any ArcTool or ArcScript as a background operation may slightly reduce its reliability.

Specify file path name of script
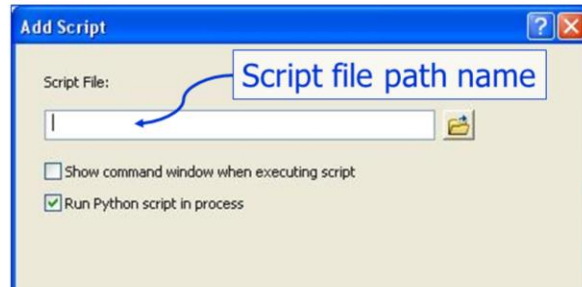
The 2nd page of the wizard will require you to specify the file name of the script. Note that the script file always remains separate from the toolbox file, even after importation is complete. To share a tool, you need to provide both the toolbox and the associated script.
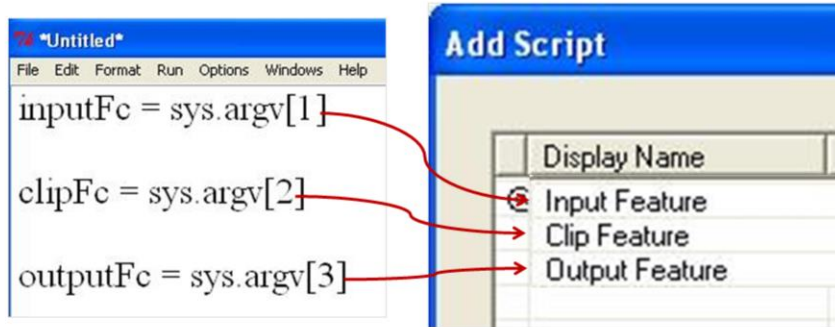
The 3rd page of the wizard sets up the user-defined parameters that will be required for the script.

The names of the parameters are specified in the top-left panel.

The parameter's data type is specified in the top-right panel.

The bottom panel allows the properties to be specified for each parameter.

The order that parameters are listed in the **Display name** panel must match the order in which they are requested by the script.

The order is determined by the parameter numbers specified in the **sys.argv** statements.

The **Display name** of the parameters will appear in the dialog window when the tool is run.

For each parameter, the **data type** must be specified.

The appropriate data type can be selected from the dropdown menu.

Parameter properties: type

Display Name	Data Type
@ Input Feature Class	Feature Class

Property	Value
Type	Required ▼
Direction	Required
MultiValue	Optional
Default	Derived
Environment	

- Derived type used when a script output needs to be exposed for use in a model.

The properties for each parameter must be specified. The type indicates whether the parameter is required, optional, or derived.

The derived parameter type is used when a parameter needs to be exposed for use in a model.

Parameter properties: direction

Display Name	Data Type
@ Input Feature Class	Feature Class

Property	Value
Type	Required
Direction	Input ▼
MultiValue	Input
Default	Output
Environment	

- Inputs provides data for script to run.
- Outputs are data created by the script.

The direction of the parameter indicates whether it is an **input** or **output**.

Inputs are used by the script whereas outputs are created by the script. Input dataset parameters require that an existing file be specified. Output dataset parameters require that a new file name be specified.

Parameter properties: multi-value

Display Name	Data Type
@ Input Feature Class	Feature Class

Property	Value
Type	Required
Direction	Input
MultiValue	No ▼
Default	No
Environment	Yes

Creates list in dialog window

buffer distance

+

×
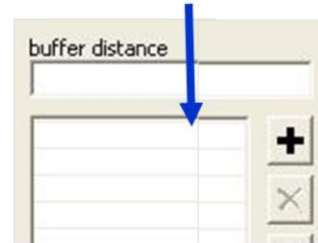
- Multi-value parameters sent to script as a strings with values separated by ";"

split string by ";"

"'500 FEET';'1000 FEET';'2000 FEET'"

10

The **multi-value** property determines if multiple values can be specified for a given parameter.

In the dialog window, a multi-value parameter will have a list that allows multiple values to be input.

When multiple values are specified for a parameter, ArcGIS will pass the parameter values to the script as a string. The values will be separated by a semi-colon.

The script must split the string to extract the values.

Parameter properties: default

Display Name	Data Type
@ Input Feature Class	Feature Class

Property	Value
Type	Required
Direction	Input
MultiValue	No
Default	C:\NRME_387\Data\Wetlands.shp
Environment	

- Default value automatically entered for parameter in dialog window unless overridden.

11

A default value may be specified for a parameter.

The default value will be automatically entered for the parameter in the dialog window when the script is run. The default is overridden if the user specifies a different value for the parameter.

Parameter properties: environment

	Display Name	Data Type
@	Input Feature Class	Feature Class

Property	Value
Type	Required
Direction	Input
MultiValue	No
Default	
Environment	

Cartographic Coordinate System [cartog
Cell Size [cellSize]
Coincident Points [coincidentPoints]
Compression [compression]
Current Workspace [workspace]
Default Output Z Value [outputZValue]
Extent [extent]
Geographic Transformations [geograph
Level Of Comparison Between Projecti

- Sets parameter equal to value of an environment setting…
  - i.e. set parameter value equal to the cell size specified in ArcMap environment settings.

12

The parameter can be set to have the value of a current environment setting in ArcMap.
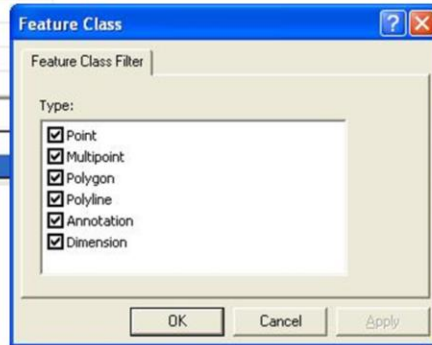
The filter property can restrict the acceptable values for a parameter.

The filter options depend on the data type of the parameter. A parameter that has a feature class data type, for example, will have filter options that can restrict the type of feature classes that can be specified.

Parameter properties: obtained from

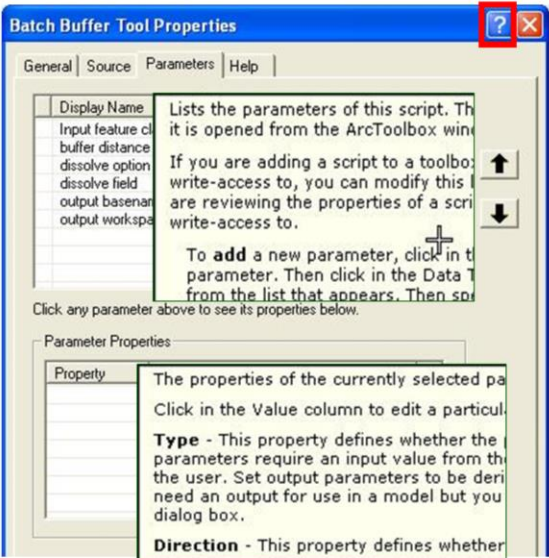| Display Name	Data Type
input FC	Feature Class
Field name	Field

Property	Value	
Type	Required	
Direction	Input	
MultiValue	No	
Default		
Environment		
Filter	None	
Obtained from	input_FC	▼
Symbology	input_FC	

- If Field name is obtained from inputFC, then only fields that exist in inputFC can be selected for Field name in dialog window.

14

The **obtained from** property can create a dependency of one parameter on another parameter. For example, if a Field type parameter is "obtained from" a feature class type parameter, then the only values that can be specified for the Field type parameter are the fields in the feature class.

To get help with settings on the parameter page, click the question mark and then click on one of the panels.

ArcScripts

- After a script is imported into a toolbox…
  - it can be run from the dialog window,
  - it can be documented like a model…
    - dialog window
    - help page
- ArcScripts can be used in models.
- Script file (.py) remains separate from the toolbox file (.tbx).
  - both files need to be present for ArcTool to work

16

After completing the parameter settings page of the wizard, the script should be added to the toolbox in ArcGIS as an ArcScript. The ArcScript is run through ArcToolbox and uses the dialog window interface. Documentation on the tool can be provided for both the dialog window and the tool's ArcGIS help page.

ArcScripts can be used in ModelBuilder, just as any other ArcTool.

The ArcGIS toolbox only references the script, it does not actually import it into the toolbox file. In order to run the script from the toolbox, the toolbox must have the access to the python file. To share an ArcScript, both the toolbox file and the python script file must be provided.