A quick tour of using iterators

ArcMap 10.7
|
Other versions

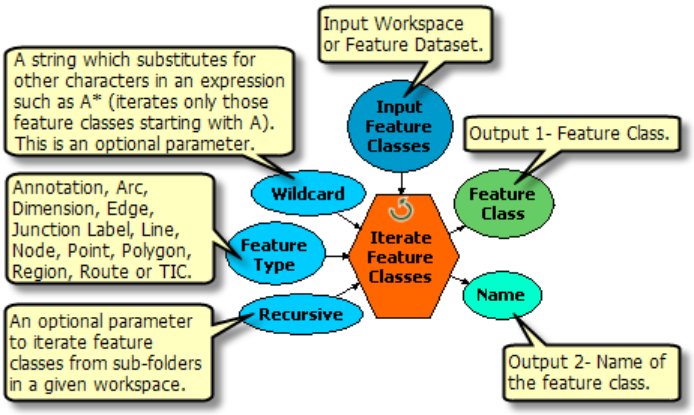
- 10.8
- 10.7
- 10.6
- 10.5
- 10.4
- 10.3

Iteration, often referred to as looping, means to repeat a process over and over with some degree of automation. Iteration is very important because automating repetitive tasks reduces the time and effort required to perform the tasks. With iteration in ModelBuilder, a process can be executed over and over using different settings or data in each iteration. ModelBuilder also provides flexibility in iteration, as an entire model or simply a single tool or process can be executed repeatedly.

Iterator	Description
For	Iterates over a starting and ending value by a given value. It works exactly like For in any scripting/programming language, executing through a set number of items.
While	Works exactly like 'while' in any scripting/programming language, executing "while" a condition is true or false for the input or set of inputs.
Iterate Feature Selection	Iterates over features in a feature class.
Iterate Row Selection	Iterates over rows in a table.
Iterate Field Values	Iterates over each value in a field.
Iterate Multivalue	Iterates over a list of values.
Iterate Datasets	Iterates over datasets in a Workspace or Feature Dataset.
Iterate Feature Classes	Iterates over feature classes in a Workspace or Feature Dataset.
Iterate Files	Iterates over files in a folder.
Iterate Rasters	Iterates over rasters in a Workspace or a Raster Catalog.
Iterate Tables	Iterates over tables in a workspace.
Iterate Workspaces	Iterates over workspaces in a folder.

Understanding an iterator

Each iterator has a set of parameters that may differ from the other iterators, but the overall structure of all iterator tools is very similar. A commonly used iterator, Iterate Feature Classes, is explained below.



Iterate Feature Classes requires an Input Workspace where all the feature classes to iterate through are stored. Two additional parameters, Wildcard and Feature Type, are used to restrict what feature classes in the workspace are iterated:

- Wildcard limits the feature classes by their names.
- Feature Type limits the feature classes by their feature types: annotation, arc, dimension, edge, junction, label, line, node, point, polygon, region, route, or TIC.

The Recursive parameter is used to control the iteration over feature classes within subfolders in the workspace.

Iterate Feature Classes has two output variables: the output feature class and the name of the feature class. The output feature class can be connected to the next tool for processing, and the Name variable can be used for inline variable substitution. For example, if the Buffer tool was added to the model, and the Feature Class variable was connected to the tool, every feature class in the workspace would be buffered.

Iterator input and output

Below is a list of iterators and what their input and outputs are. A number of iterators have a Value or Name as second output, which can be used for inline variable substitution.

Iterator	Input	Output 1	Output 2
For	Values	Value	-
While	Values	Boolean—True or False Value	-
Iterate Feature Selection	Features	Feature	Value
Iterate Row Selection	Table	Record	Value
Iterate Field Values	Table	Field Value	-
Iterate Multivalue	Values	Value	-
Iterate Datasets	Workspace or Feature Dataset	Dataset	Name
Iterate Feature Classes	Workspace or Feature Dataset	Feature Class	Name
Iterate Files	Folder	File	Name
Iterate Rasters	Workspace or Raster Catalog	Raster Dataset	Name
Iterate Tables	Workspace	Table	Name
Iterate Workspaces	Folder	Workspace	Name

Note:

- Only one iterator can be used per model. The options to add another iterator will be disabled if one iterator exists in the model.
- If an iterator is added to a model, all tools in the model iterate for each value in the iterator. If you do not want to run each tool in the model for each iterated value, create a submodel/model within a model/nested model that contains only the iterator and add it as a model tool into the main model.
- If a model containing an iterator is exported to a Python script, the script will not include the iteration logic. Python listing logic can be added to the script to achieve a similar effect.
- Using an iterator will set a default value of -1 in Iteration options in Model Properties, which simply means that the model will run for an unlimited number of times, or based on the number of inputs in an iterator, and not on a set number.
- The output of any tool connected to the iterator can have (if required) a unique name for each iteration to avoid being overwritten by
  - Using the system variable %n%, for example, C:\Scratch\scratch.gdb\output_%n%.
  - Using the Name or Value output of the iterator used in the model as an inline variable, for example, C:\Scratch\scratch.gdb\output_%Name%, C:\Scratch\scratch.gdb\output_%Value%.
  - Using any other variable in the model as an inline variable; for example, if there is a variable XYZ, the name of the output can be C:\Scratch\scratch.gdb\output_%XYZ%. This variable should not contain a constant value, but a value that changes with each iteration; otherwise, the file will be overwritten with each iteration.
- Tool outputs with the Add To Display option checked are added to the display in ArcMap using the variable name. If you want to add the output of each iteration to display in ArcMap with the actual unique output name instead of the name of the variable:
  1. Connect the output to be displayed to the Collect Values tool.
  2. Right-click the output of Collect Values and check the Add To Display.
  3. If the model will be run from the model tool dialog box, then make the output of Collect Values a model parameter, as only the output model parameters are added to display.

Legacy:

Iterators replace the series option in Model Properties available prior to ArcGIS 10.

Learn more

- Which iterator to use for what task
- An example of using an iterator
- Accessing iterators in ModelBuilder

Related topics

- Accessing data using cursors
- Create lists of data