**Running Scripts**

To run an existing script file, press "Run Script" button on the MAXScript utility rollout, *File > Run Script* in the Listener menu bar, or *MAXScript > Run Script* in the *3ds Max* menu bar. This opens a common File Open dialog for choosing the script. MAXScript then reads and executes the selected script. Any output is printed to the Listener output pane.

You can also run a script from Listener or from within other scripts using the *fileIn()* method:

*fileIn* <filename\_string> [ *quiet:*<boolean> ]

where <filename\_string> is a string literal or an expression that evaluates to a string, and specifies the name of the script file whose content is executed. The script file content is executed one expression at a time, and halts processing if an error is encountered at any point. By default, the file is not listed as it is loaded; use *quiet:false* to get a running listing to the Listener.

|  |
| --- |
| *EXAMPLEUSAGES ARE:* |
| fileIn "my\_script.ms" |
| or |
| scriptfile="my\_script.ms"fileIn scriptfile |

The script file content is compiled in a global scope context, as opposed to the scope in effect when the *filein()* method is executed.

**Including Scripts Within Scripts**

MAXScript provides a compile-time source-file include mechanism, allowing you to break up large scripts into smaller files that can be included at nearly any point in a script. You can use the *include* <file> construct at any point in your code to effectively insert the contents of a file at that point in your code.

The form is:

*include* "filename\_string"

This is a compile-time construct, therefore the file name specification must be a string literal, and not a variable or an expression.

|  |
| --- |
| *EXAMPLE* |
| utility foo "Baz"  (  local a, b, c  include "foo-ui.ms"  rollout bar "Bar"  (  include "bar-rollout.ms"  )  include "foo-handlers.ms"  ) |

The *include* <file> construct is effectively replaced in the source code at that point with the contents of the named file.

You can nest included files as deeply as needed; included files can include other files, and so on.

Because *include* is a compile-time construct, the current MAXScript scope is maintained within the included file. This is opposed to the *fileIn()* method described in [Running Scripts](mk:@MSITStore:E:\maxscript-2015-help.chm::/files/GUID-86D82FCE-B88F-4487-9B34-B6222EDA1C71.htm), whose script file content is compiled in a global scope context.

The *include* <file> can appear at any point a new token is expected, such as a name, literal, or punctuation. This means that you could complete a partial expression with an included file.

|  |
| --- |
| *FOR EXAMPLE* |
| include "op1.ms"+ include "op2.ms"  if include "test2.ms" then print a else print b |

You cannot place an *include* <file> within a token. For example, the following is not valid:

|  |
| --- |
| *NOT VALID:* |
| timeval = 2:include "framenum.ms".0 |

**The SDK and MAXScript**

*3ds Max* gives you two ways to write plug-in applications: the SDK, which uses C++, and MAXScript, the program's native scripting language.

Which language you choose depends partly on how you want to work, and partly on what you want your plug-in to accomplish. Both languages have their strengths and limitations, but you can develop complex applications with either of them.

**MAXScript**

In general, MAXScript plug-ins run more slowly than comparable plug-ins written in C++, so if performance is an issue, using the SDK is probably preferable.

On the other hand, MAXScript provides some methods that are higher level than those to be found in the C++ SDK, and supports a few *3ds Max* features and capabilities that are not exposed to the SDK. If your feature needs functionality supported by MAXScript but not the SDK, then MAXScript is your only choice. In particular, exposing *3ds Max* features via OLE/ActiveX/DotNet controls is easier to code in MAXScript than it is with the SDK.

MAXScript can be useful for prototyping plug-ins, developing comparatively small features, and writing test suites.