

**Flags**

/Q	Command is not printed in the command line or history area.
/Z	No error reporting.

**See Also**

**Operation Queue** on page IV-278 for more details on using Execute/P with the operation queue.

**ExecuteScriptText**

**ExecuteScriptText** [*flags*] *textStr*

The ExecuteScriptText operation passes your text to Apple's scripting subsystem for compilation and execution or to the Windows command line.

Text produced by the script, including error messages, is returned in the string output variable S\_value. On Windows, S\_value is set only if you include the /B (execute in background) flag.

**Parameters**

*textStr* must contain a valid AppleScript program or Windows command line.

**Flags**

/B	Execute the script in the background task, i.e., leaving Igor as the active application. If you omit /B on Windows, the program invoked by the script becomes the active program. If you include /B, Igor remains the active program. /B is ignored on Macintosh and the script is always executed in the background.
/W= <i>waitTime</i>	<i>waitTime</i> is the maximum time in seconds to wait before ExecuteScriptText returns. /W is accepted on any platform but is supported only on Windows.
/UNQ	Removes any leading and trailing double-quote characters from S_value. /UNQ is useful on Macintosh only and has no effect on Windows. The /UNQ flag was added in Igor Pro 7.00.
/Z	Include /Z if you want to handle errors in your code. Omit it if you want ExecuteScriptText to report errors to Igor. See <i>ExecuteScriptText Error Handling</i> below for details.

**ExecuteScriptText Error Handling**

If you omit the /Z flag, if the script returns an error, ExecuteScriptText returns the error to Igor and procedure execution stops.

If you include the /Z flag then ExecuteScriptText sets the V\_flag output variable to a nonzero value if the script returned an error or to zero if no error occurred. ExecuteScriptText does not report the error to Igor so procedure execution continues.

**ExecuteScriptText on Macintosh**

Text generated by the script, including error messages, is returned in the S\_value output string variable.

Prior to Igor Pro 8.00, S\_value containing a text value was typically quoted. This does not appear to be the case starting with Igor Pro 8 which uses more up-to-date system calls. We recommend that you use the /UNQ flag with Igor 7 and later so that you can reliably expect S\_value to be unquoted.

The /B flag (execute in background) is ignored.

The /W flag is ignored and ExecuteScriptText does not return until the script is finished.

Prior to Igor Pro 8.00, the current directory was always / and the PATH environment variable was the system default, not the user default (usually PATH=/usr/bin:/bin:/usr/sbin:/sbin.). Now neither of these is guaranteed to be the case. You should not make any assumptions about the current directory or the value of any environment variable. Use full paths to files and programs whenever you need to provide a path.

### Macintosh Examples

```
// Macintosh: Convert file.PICT to file.GIF
String script = "tell application \"clip2gif\" "
script += "to save file \"HD:file.PICT\" \"r"
ExecuteScriptText/Z script

// Macintosh: Execute a Unix shell command
Function/S DemoUnixShellCommand()
    // Paths must be POSIX paths (using /).
    // Paths containing spaces or other nonstandard characters
    // must be single-quoted. See Apple Technical Note TN2065 for
    // more on shell scripting via AppleScript.
    String unixCmd
    unixCmd = "ls '/Applications/Igor Pro 8 Folder'"

    String igorCmd

    sprintf igorCmd, "do shell script \"%s\"", unixCmd
    Print igorCmd // For debugging only

    ExecuteScriptText/UNQ igorCmd
    Print S_value // For debugging only
    return S_value
End
```

### ExecuteScriptText on Windows

textStr contains the full path to a batch file or the name of an executable file with optional Windows-style path and optional arguments:

```
[path]executableName [.exe] [arg1 ]...
```

If you omit path and executableName ends with ".exe" or with no extension, Igor locates the executable by searching first the registry and then along the PATH environment variable. If not found, then the Igor Pro Folder directory is assumed. Here is an example of a command consisting of just the name of an executable:

```
ExecuteScriptText "calc" // calc.exe, calculator
```

When calling a batch file or other non-\*.exe file, supply the full path. If the path (or file name) contains spaces, you must quote the path:

```
ExecuteScriptText "\"C:\\Program Files\\my.bat\""
```

The outer double-quotes are consumed by ExecuteScriptText. The inner double quotes, represented by the \" escape sequence, remain in the command passed to the operating system.

Use the /B flag to run the command in the background, keeping Igor as the active application. Omit /B to allow the program launched by the command to become the active program. When executing a batch file, /B prevents the DOS window from appearing.

If you include /B, output from the command is returned via the S\_value output string variable. If you omit /B, S\_value is set to "".

The /W=waitTime flag provides a way to time out if a command takes too long. It is useful when calling non-GUI programs only. For GUI programs, ExecuteScriptText returns as soon as the GUI program starts processing events, regardless of the value specified for waitTime.

For non-GUI programs, if you specify a positive value for waitTime, ExecuteScriptText waits up to that many seconds for the command to complete. If the command fails to complete within that period of time, ExecuteScriptText returns an error. If you omit the /W flag or if you pass zero for waitTime, ExecuteScriptText waits until the command completes no matter how long it takes. If the script being executed creates a command window, via cmd.exe, for example, the Windows process completes only when the command window closes, not after the script commands execute.

### Windows Examples

```
// Windows: Execute a DOS command in the background
ExecuteScriptText/B "hostname"; Print S_value

// Windows: Open MatLab in background
ExecuteScriptText/B "C:\\Matlab\\bin\\matlab.exe myFile.m"

// Windows: Pass a script to Windows Script Host
ExecuteScriptText/W=5 "WScript.exe \"C:\\Test Script.vbs\""
```