Igor supports the creation and execution of simple AppleScripts in order to send commands to other programs.

To execute an AppleScript program, you first compose it in a string and then pass it to the ExecuteScriptText operation, which in turn passes the text to Apple's scripting module for compilation and execution. The result, which might be an error message, is placed in a string variable named S_value. Igor does not save the compiled script so every time you call ExecuteScriptText your script will have to be recompiled. See the **ExecuteScriptText** operation on page V-205 for additional details.

The documentation for the **ExecuteScriptText** operation (page V-205) includes an example that shows how to execute a Unix command.

Because there is no easy way to edit a script or to see where errors occur, you should first test your script using Apple's Script Editor application.

You can use "Silent 2" to prevent commands your script sends to Igor from being placed in the history area.

You can send commands to Igor without using the tell keyword.

You should check your quoting carefully. Your text must be quoted both for Igor and for Apple's scripting system. For example,

```
ExecuteScriptText "Do Script \"Print \\\"hello\\\"\""
```

You should compose scripts in string variables one line at a time to improve readability.

If an error occurs that you can't figure out, print the string, copy from the history and paste into a Script Editor for debugging.

If the script returns a text return value, it may be quoted within the S_value string. See the discussion of quoting in the **ExecuteScriptText** documentation for details.

Don't forget to include the carriage return escape code, \r, at the end of each line of a multiline script.

The first time you call this routine, it may take an extra long time while the Mac OS loads the scripting modules.

Executing Unix Commands on Mac OS X

On Mac OS X, you can use AppleScript to send a command to the Unix shell. Here is a function that illustrates this:

```
Function/S ExecuteUnixShellCommand(uCommand, printCommandInHistory,
                                   printResultInHistory [, asAdmin])
   String uCommand                 // Unix command to execute
   Variable printCommandInHistory
   Variable printResultInHistory
   Variable asAdmin                // Optional - defaults to 0

   if (ParamIsDefault(asAdmin))
      asAdmin = 0
   endif

   if (printCommandInHistory)
      Printf "Unix command: %s\r", uCommand
   endif

   String cmd
   sprintf cmd, "do shell script \"%s\"", uCommand
   if (asAdmin)
      cmd += " with administrator privileges"
   endif
   ExecuteScriptText/UNQ/Z cmd // /UNQ removes quotes surrounding reply

   if (printResultInHistory)
      Print S_value
   endif

   return S_value
End
```