

You can use the **SetFileInfo** operation (see page V-845) to set the visibility and read-only properties of a file:

```
SetFileInfo /INV=1 /RO=1 "<path to file>"
```

The file will be invisible in Igor the next time you open it, typically by opening an experiment or using a #include statement.

On Macintosh, the file disappears from the Finder when you execute the command.

On Windows, merely setting the hidden property is not sufficient to actually hide the file. It is actually hidden only if the Hide Files of These Types radio button in the View Options dialog is turned on. You can access this dialog by opening a folder in the Windows desktop and choosing View→Options from the folder's menu bar. Although the hidden property in Windows does not guarantee that the file will be hidden in the Windows desktop, it does guarantee that it will be hidden from within Igor.

After the files are set to be invisible and read-only, if you want to edit them in Igor, you must close them (typically by closing the open experiment), set the files to be visible and read/write again, and then open them again.

Igor's behavior is changed in the following ways for a procedure file set to invisible:

1. The window will not appear in the Windows→Procedure Windows menu.
2. Procedures in the window will not appear in the contextual pop-up menu in other procedure windows (Control-click on *Macintosh*, right-click on *Windows*).
3. If the user presses Option (*Macintosh*) or Alt (*Windows*) while choosing the name of a procedure from a menu, Igor will do nothing rather than its normal behavior of displaying the procedure window.
4. When cycling through procedure windows using Command-Shift-Option-M (*Macintosh*) or Ctrl+Shift+Alt+M (*Windows*), Igor will omit the procedure window.
5. The Button Control dialog, Pop-Up Menu Control dialog, and other control dialogs will not allow you to edit procedures in the invisible file.
6. The Edit Procedure and Debug buttons will not appear in the Macro Execute Error dialog.
7. If an error occurs in a function in the invisible file and the Debug On Error flag (Procedure menu) is on, the debugger will act as if Debug On Error were off.
8. The debugger won't allow you to step into procedures in the invisible file.
9. The **ProcedureText** and **ProcedureVersion** functions and **DisplayProcedure** will act as if procedures in the invisible file don't exist. The **MacroList** and **FunctionList** functions will, however work as usual.
10. The GetIgorProcedure and SetIgorProcedure XOPSupport routines in the XOP Toolkit will act as if procedures in the invisible file don't exist. The GetIgorProcedureList function will, however work as usual.

## Inserting Text Into a Procedure Window

On occasion, you may want to copy text from one procedure file to another. With the procedure window active, choose Edit→Insert File. This displays an Open File dialog in which you can choose a file and then inserts its contents into the procedure window.

## Adopting a Procedure File

Adoption is a way for you to copy a procedure file into the current experiment and break the connection to its original file. The reason for doing this is to make the experiment self-contained so that, if you transfer it to another computer or send it to a colleague, all of the files needed to recreate the experiment are stored in the experiment itself.