RELOAD CHANGED PROCS, added in Igor Pro 9.00, causes Igor to reload procedure windows from their disk files if they have been modified outside of Igor. This is useful if you run a system that modifies procedure files using external software which is part of a system running from Igor. The sequence of events is:

1. Most likely, you use ExecuteScriptText to trigger the external modifications.

2. You use Execute/P "RELOAD CHANGED PROCS " to reload the modified procedures.

3. You use Execute/P "COMPILEPROCEDURES " to compile the modified procedures.

Operation Queue Example

```
Function DemoQueue()
   Execute/P "INSERTINCLUDE <Multipeak Fitting>"
   Execute/P "COMPILEPROCEDURES "
   Execute/P "fStartMultipeakFit2()"
   Execute/P "Sleep 00:00:04"
   Execute/P "NEWEXPERIMENT "
   Execute/P "LOADFILE :Examples:Feature Demos:Live mode.pxp"
   Execute/P "DoWindow/F Graph0"
   Execute/P "StartButton(\"StartButton\")"
End
```

Operation Queue For Loading Packages

One important use of the operation queue is providing easy access to useful procedure packages. The "Igor Pro Folder/Igor Procedures" folder contains a procedure file named “DemoLoader.ipf” that creates the Packages submenus found in various menus. To try it out, choose one of the items from the Analysis→Packages menu.

DemoLoader.ipf is an independent module. To examine it, execute:

```
SetIgorOption IndependentModuleDev=1
```

and then use the Windows→Procedure Windows submenu.

User-Defined Hook Functions

Igor calls specific user-defined functions, called “hook” functions, if they exist, when it performs certain actions. This allows savvy programmers to customize Igor’s behavior.

In some cases the hook function may inform Igor that the action has been completely handled, and that Igor shouldn’t perform the action. For example, you could write a hook function to load data from a certain kind of text file that Igor can not handle directly.

This section discusses general hook functions that do not apply to a particular window. For information on window-specific events, see Window Hook Functions.

There are two ways to get Igor to call your general hook function. The first is by using a predefined function name. For example, if you create a function named AfterFileOpenHook, Igor will automatically call it after opening a file. The second way is to explicitly tell Igor that you want it to call your hook using the **SetIgorHook** operation.

If you use a predefined hook function name, you should make the function static (private to the file containing it) so that other procedure files can use the same predefined name.

Here are the predefined hook functions.

Action	Hook Function Called
Procedures were successfully compiled	**AfterCompiledHook**
A file or experiment was opened	**AfterFileOpenHook**