

Action	Hook Function Called
The Windows-only "MDI frame" (main application window) was resized	AfterMDIFrameSizedHook
A target window was created	AfterWindowCreatedHook
The debugger window is about to open	BeforeDebuggerOpensHook
An experiment is about to be saved	BeforeExperimentSaveHook
A file or XOP is about to be opened	BeforeFileOpenHook
A modification to a procedure window is about to cause procedures to be uncompiled	BeforeUncompiledHook (Igor Pro 8.03 or later)
HDF5 dataset for wave is about to be written	HDF5SaveDataHook (Igor Pro 9.00 or later)
Igor is about to open a new experiment	IgorBeforeNewHook
Igor is about to quit	IgorBeforeQuitHook
Igor is building and enabling menus or about to handle a menu selection	IgorMenuHook
Igor is about to quit	IgorQuitHook
Igor launching or creating a new experiment	IgorStartOrNewHook

To create hook functions, you must write functions with the specified names and store them in any procedure file. If you store the procedure file in "Igor Pro User Files/Igor Procedures" (see **Igor Pro User Files** on page II-31 for details), Igor will automatically open the file and compile the functions when it starts up and will execute the IgorStartOrNewHook function if it exists.

To allow for multiple procedure files to define the same predefined hook function, you should declare your hook function static. For example:

```
static Function IgorStartOrNewHook(igorApplicationNameStr)
    String igorApplicationNameStr
```

The use of the static keyword makes the function private to the procedure file containing it and allows other procedure files to have their own static function with the same name.

Igor calls static hook functions after the **SetIgorHook** functions are called. The static hook functions themselves are called in the order in which their procedure file was opened. You should not rely on any execution order among the static hook functions. However, any hook function which returns a nonzero result prevents remaining hook functions from being called and prevents Igor from performing its usual processing of the hook event. In most cases hook functions should exercise caution in returning any value other than 0. For hook functions only, returning a NaN or failing to return a value (which returns a NaN) is considered the same as returning 0.

The following sections describe the individual hook functions in detail.

AfterCompiledHook

AfterCompiledHook ()

AfterCompiledHook is a user-defined function that Igor calls after the procedure windows have all been compiled successfully.

You can use AfterCompiledHook to initialize global variables or data folders, among other things.

The function result from AfterCompiledHook must be 0. All other values are reserved for future use.

See Also

SetIgorHook, **BeforeUncompiledHook**, **User-Defined Hook Functions** on page IV-280.