

## Named Window Hook Functions

A named window hook function takes one parameter - a `WMWinHookStruct` structure. This built-in structure provides your function with information about the status of various window events.

The named window hook function has this format:

```
Function MyWindowHook(s)
    STRUCT WMWinHookStruct &s

    Variable hookResult = 0

    switch(s.eventCode)
        case 0:           // Activate
            // Handle activate
            break

        case 1:           // Deactivate
            // Handle deactivate
            break

        // And so on . . .
    endswitch

    return hookResult      // 0 if nothing done, else 1
End
```

If you handle a particular event and you want Igor to ignore it, return 1 from the hook function. However, you cannot make Igor ignore a window kill event - once the kill event is received the window will be killed.

## Named Window Hook Events

Here are the events passed to a named window hook function:

eventCode	eventName	Notes
0	"Activate"	
1	"Deactivate"	
2	"Kill"	Returning 1 when you receive this event does not cause Igor to ignore the event. At this point, you cannot prevent the window from being killed. See the <b>killVote</b> event to prevent the window being killed.
3	"Mousedown"	
4	"Mousemoved"	
5	"Mouseup"	
6	"Resize"	
7	"Cursormoved"	See <b>Cursors — Moving Cursor Calls Function</b> on page IV-339.
8	"Modified"	A modification to the window has been made. See <b>Modified Events</b> on page IV-299.
9	"Enablemenu"	
10	"Menu"	
11	"Keyboard"	See <b>Keyboard Events</b> on page IV-300.
12	"moved"	
13	"renamed"	

## Chapter IV-10 — Advanced Topics

eventCode	eventName	Notes
14	"subwindowKill"	One of the window's subwindows is about to be killed.
15	"hide"	The window or one of its subwindows is about to be hidden. See <b>Window Hook Show and Hide Events</b> on page IV-304.
16	"show"	The window or one of its subwindows is about to be unhidden. See <b>Window Hook Show and Hide Events</b> on page IV-304.
17	"killVote"	Window is about to be killed. Return 2 to prevent the window from being killed, otherwise return 0.  <b>Note:</b> Don't delete data structures during this event, use killVote only to decide whether the window kill should actually happen. Delete data structures in the kill event. See <b>Window Hook Deactivate and Kill Events</b> on page IV-303.
18	"showTools"	
19	"hideTools"	
20	"showInfo"	
21	"hideInfo"	
22	"mouseWheel"	
23	"spinUpdate"	This event is sent only to windows marked via DoUpdate/E=1 as progress windows. It is sent when Igor spins the beachball cursor. See <b>Progress Windows</b> on page IV-156 for details.
24	"tableEntryAccepted"	This event is sent to tables only. It is sent when the user manually accepts text entered in the table entry line, for example by clicking the checkmark button or pressing the Enter or Return keys.  It is also sent by a <b>ModifyTable</b> entryMode=1 command.  This hook event was added in Igor Pro 8.03.
25	"tableEntryCancelled"	This event is sent to tables only. It is sent when the user cancels text entry, for example by clicking the X button or pressing the Escape key.  It is also sent by a <b>ModifyTable</b> entryMode=0 command.  This hook event was added in Igor Pro 8.03.
26	"earlyKeyboard"	See <b>EarlyKeyboard Events</b> on page IV-302.