As of Igor Pro 9, you can provide the optional beGraceful flag when calling ThreadGroupRelease. If you specify beGraceful=1, ThreadGroupRelease sets a clearable abort flag. "Clearable" means that, if a thread worker function has a catch block, the abort flag is cleared when the catch block is called. This allows code in the catch block to execute without interference. The catch block must include a return statement so that the thread worker function returns.

The interaction between Igor and threads when an abort occurs is complex. You don't need to understand more that is explained in the preceding paragraphs of this section. Advanced programmers who want to understand the details can find them in the Threads and Aborts example experiment.

More Multitasking Examples

More multitasking examples can be found in the following example experiments:

The Multithreaded LoadWave demo experiment in "Igor Pro Folder/Examples/Programming".

The Multithreaded Mandelbrot demo experiment in "Igor Pro Folder/Examples/Programming".

The Multiple Fits in Threads demo experiment in "Igor Pro Folder/Examples/Curve Fitting".

The Slow Data Acq demo experiment in "Igor Pro Folder/Examples/Programming".

The Thread-at-a-Time demo experiment in "Igor Pro Folder/Examples/Programming".

Cursors — Moving Cursor Calls Function

You can write a hook function which Igor calls whenever a cursor is moved.

Graph-Specific Cursor Moved Hook

The preferred way to do this is to use SetWindow to designate a window hook function for a specific graph window (see **Window Hook Functions** on page IV-293). In your window hook function, look for the cursormoved event. Your hook function receives a **WMWinHookStruct** structure containing fields that describe the cursor and its properties.

For a demo of this technique, choose File→Example Experiments→Techniques→Cursor Moved Hook Demo.

Global Cursor Moved Hook

This section describes an old technique in which you create a hook function that is called any time a cursor is moved in any graph. This technique is more difficult to implement and kludgy, so it is no longer recommended.

You can write a hook function named CursorMovedHook. Igor automatically calls it whenever any cursor is moved in any graph, unless Option (*Macintosh*) or Alt (*Windows*) is pressed.

The CursorMovedHook function takes one string argument containing information about the graph, trace or image, and cursor in the following format:

```
GRAPH:graphName;CURSOR:<A - J>;TNAME:traceName; MODIFIERS:modifierNum;
ISFREE:freeNum;POINT:xPointNumber; [YPOINT:yPointNumber;]
```

The *traceName* value is the name of the graph trace or image to which the cursor is attached.

The *modifierNum* value represents the state of some of the keyboard keys summed together:

1 If Command (*Macintosh*) or Ctrl (*Windows*) is pressed.

2 If Control (*Macintosh only*) is pressed.

4 If Shift is pressed.

8 If Caps Lock is pressed.