0: The background procedure executed normally.

1: The background procedure wants to stop the background task.

2: The background procedure encountered an error and wants to stop the background task.

Normally the background procedure should return 0 and the background task will continue to run. If you return a non-zero value, Igor stops the background task. You can tell Igor to terminate the background task by returning the value 1 from the background function.

If you forget to add a return statement to your background procedure, this acts like a non-zero return value and stops the background task.

Background Task Period

The CtrlNamedBackground operation's period keyword takes an integer parameter expressed in ticks. A tick is approximately 1/60th of a second. Thus the timing of Igor background tasks has a nominal resolution of 1/60th of a second.

You can override the specified period in the background task procedure by writing to the nextRunTicks field of the **WMBackgroundStruct** structure. This is needed only if you want your procedure to run at irregular intervals.

The actual time between calls to the background procedure is not guaranteed. Igor runs the background task from its outer loop, when Igor is doing nothing else. If you do something in Igor that takes a long time, for example performing a lengthy curve fit, running a user-defined function that takes a long time, or saving a large experiment, Igor's outer loop does not run so the background task will not run. If you do something that causes a compilation of Igor procedures to fail, the background task is not called. On Macintosh, the background task is not called while a menu is displayed or while the mouse button is pressed.

If you need your background task to continue running even if you edit other procedures in Igor, you need to make your project an independent module. See **Independent Modules** on page IV-238 for details.

If you need precise timing that can not be interrupted, things get much more complicated. You need to do your data acquisition in an Igor thread running in an independent module or in a thread created by an XOP that you write. See **ThreadSafe Functions and Multitasking** on page IV-329 for details.

The shortest supported period is one tick. The minimum actual period for the background task depends on your hardware and what your background task is doing. If you set the period too low for your background task, interacting with Igor becomes sluggish.

It is very easy to bog your computer down using background tasks. If the background task takes a long time to execute or if it triggers something that takes a long time (like a wave dependency formula or updating a complex graph) then it may appear that the system is hung. It is not, but it may take longer to respond to user actions than you are willing to wait.

Background Task Limitations

The principal limitation of Igor background tasks is that they are stopped while other operations are taking place. Thus, although you can type commands into the command line without disrupting the background task, when you press Return the task is stopped until execution of the command line is finished.

Background tasks do not run if procedures are in an uncompiled state. If you need your background task to continue running even if you edit other procedures in Igor, you need to make your project an independent module. See **Independent Modules** on page IV-238 for details.

On Macintosh, the background task does not run when the mouse button is pressed or when a menu is displayed.