

These restrictions apply even if you call a threadsafe function from a non-threadsafe function.

The main technique for debugging threadsafe code is the use of print statements. See Debugging With Print Statements.

You can use the debugger on functions marked as threadsafe by temporarily disabling multithreading by executing:

```
SetIgorOption DisableThreadSafe = 1      // Disable multithreading
```

This causes Igor to recompile all procedures and to ignore the Threadsafe and MultiThread keywords. You can then debug procedures using the debugger. When you are finished, re-enable multithreading by executing:

```
SetIgorOption DisableThreadSafe = 0      // Re-enable multithreading
```

Debugger Shortcuts

Action	Shortcut
To enable debugger	Choose Enable Debugger from the Procedure menu or choose Enable Debugger from the procedure window's pop-up menu after Control-clicking (<i>Macintosh</i>) or right-clicking (<i>Windows</i>).
To automatically enter the debugger when an error occurs	Choose Debug on Error from the Procedure menu or choose Enable Debugger from a procedure window's pop-up menu after Control-clicking (<i>Macintosh</i>) or right-clicking (<i>Windows</i>).
To set or clear a breakpoint	Click in the left margin of the procedure window or click anywhere on the procedure window line where you want to set or clear the breakpoint and choose Set Breakpoint or Clear Breakpoint from a procedure window's pop-up menu after Control-clicking (<i>Macintosh</i>) or right-clicking (<i>Windows</i>).
To enable or disable a breakpoint	Shift-click a breakpoint in the left margin of the procedure window. Click anywhere on the procedure window line where you want to enable or disable the breakpoint and choose Enable Breakpoint or Disable Breakpoint from a procedure window's pop-up menu after Control-clicking (<i>Macintosh</i>) or right-clicking (<i>Windows</i>).
To execute the next command	On <i>Macintosh</i> press Enter, keypad Enter, or Return. For <i>Windows</i> , if no button has the focus, press Enter or Return. Otherwise, click the yellow arrow button.
To step into a subroutine	Press the +, =, or keypad + keys, or click the blue descending arrow button.
To step out of a subroutine to the calling routine	Press the -, _ (underscore) or keypad - keys, or click the blue ascending arrow button.
To resume executing normally	Press Escape (Esc), or click the green arrow button.
To cancel execution	Click the red stop sign button.
To edit the value of a macro or function variable	Double-click the second column of the variables list, edit the value, and press Return or Enter.
To set the value of a function's string to null	Double-click the second column of the variables list, type "<null>" (without the quotes), and press Return or Enter.
To view the current value of a macro or function variable	Move the cursor to the procedure text of the variable name and wait. On <i>Macintosh</i> , the value appears to the right of the debugger buttons. On <i>Windows</i> , the value appears in a tooltip window.

Action	Shortcut
To view the current value of an expression	Select the expression text with the cursor, position the cursor over the selection, and wait. (Expressions involving user-defined functions will not be evaluated unless V_debugDangerously is set to 1.)
To view global values in the current data folder	Choose “local and global variables” from the debugger pop-up menu.
To view type information about variables	Choose “show variable types” from the debugger pop-up menu.
To resize the columns in the variables list	Drag a divider in the list to the left or right.
To show or hide the Waves, Structs, and Expressions pane	Drag the divider on the right side of the Variables list left or right.

--