1. Choose "Global Waves", and then choose a wave from the popup wave browser.
2. Choose "Local WAVEs, SVARs, and Strings", and then choose a wave from among the objects listed.
3. Double-click any WAVE reference in the Variables list.

Inspecting Waves in a Table

You can edit the values in a wave using the table, just like a regular table. With the contextual menu you can alter the column format, among other things.

Inspecting Waves in a Graph

You can view waves in a Graph. With the contextual menu, you can choose to show or hide the axes and change display modes.

Two-dimensional waves are displayed as an image plot.

Inspecting Strings

Select a String or char array to be inspected by two methods:

1. Choose "Local WAVEs, SVARs, and Strings", and then choose a String, SVAR or char array from among the objects listed.
2. Double-click any String, SVAR or char array in the Variables list.

The Procedure Pane

The procedure pane contains a copy of the procedure window of the routine selected in the Stack List. You can set and clear breakpoints in this pane just as you do in a procedure window, using the breakpoint margin and the Control-click (*Macintosh*) or right-click (*Windows*) menu.

A very useful feature of the debugger is the automatic text expression evaluator that shows the value of a variable or expression under the cursor. The value is displayed as a tool tip. This is often faster than scrolling through the Variables List or entering an expression in the Expressions List to determine the value of a variable, wave, or structure member reference.

The value of a variable can be displayed whether or not the variable name is selected. To evaluate an expression such as "wave[ii]+3", the expression must be selected and the cursor must be over the selection.

The debugger won't evaluate expressions that include calls to user-defined functions; this prevents unintended side effects (a function could overwrite a wave's contents, for example). You can remove this limitation by creating the global variable root:V_debugDangerously and setting it to 1.

After You Find a Bug

Editing in the debugger window is disabled because the code is currently executing. Tracking down the routine after you've exited the debugger is easy if you follow these steps:

1. Scroll the debugger text pane back to the name of the routine you want to modify, and select it.
2. Control-click (Macintosh) or Right-click (Windows) the name, and choose "Go to <routineName>" from the pop-up menu.
3. Exit the debugger by clicking the "Go" button or by pressing Escape.

Now the selected routine will be visible in the top procedure window, where you can edit it.

Debugging ThreadSafe Code

The Igor debugger can not be used with threadsafe functions.

The debugger does not break on breakpoints in threadsafe functions.

The debugger does not allow you to step into a threadsafe function.