
Subject: IDLWAVE 5.6 -- idlwave.org
Posted by [JD Smith](#) on Tue, 10 May 2005 02:08:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

IDLWAVE 5.6 -- <http://idlwave.org>

This new IDLWAVE release features performance improvements and adds several exciting new features. A new right-clickable "active" breakpoint line is on by default, and allows you to perform common operations on breakpoints (like disabling, adding a condition, etc.) trivially with the mouse, and also pops up a tip describing the breakpoint when the mouse hovers over the line. See the new screenshot: <http://idlwave.org/screenshots/>.

Breakpoint operation stability and performance has been significantly enhanced, especially when you have a large number of compiled routines (e.g. thousands). Large structures are now scanned much more quickly for class and structure-level in text help, and global structure re-indentation. Completion and help on inherited structures (in addition to inherited classes) is now supported, even if the structure is not documented in its own `__define` file, but instead in the current class file. With the optional `complete-structtag` module loaded, structure member can now be completed directly in the shell; IDL is queried in the background to determine the field names for you (similar to object methods). Thanks to Carsten Dominik for submitting this useful addition.

A few new convenience bindings: `C-c C-t` prompts to visit a routine in this file (with completion), and `\pv` and `\pv` are new abbreviations for "if `ptr_valid()`" then and "`ptr_valid()`". Examining complicated expression at higher levels on the calling stack is now more robust, and should not cause IDLWAVE to pause. Many smaller bug fixes and improvements can be found as well; see the `CHANGES` file.

Get your fix at:

<http://idlwave.org>

JD

P.S. Those of you who set your own shell prompt to something other than "IDL>" should be sure to use the format "`^\r? ?IDL>` " to avoid problems with IDL's intermittent use of return characters in the prompt, which would result in an apparently hung IDL session (anecdotal evidence indicates the frequency of this problem increased with IDLv6.1).

P.P.S. The eagle-eyed among you may notice that IDLWAVE v5.5 went missing: this special version was made for inclusion with a new GNU

Emacs release (at some point).

=====

IDLWAVE Tip of the Month:

Electric Debug mode is IDLWAVE's quick and powerful debugging mode, and it makes stepping through code, moving through the calling stack, examining variables and expressions, setting and modifying breakpoints, and much more a matter of a few key presses. If you aren't using it, and you debug IDL code and use breakpoints, you should give it a try.

This is how it works. By default, when a breakpoint is hit, the buffer with that breakpoint is loaded, and the breakpoint line is highlighted in purple. The word "* Debugging *" should appear in the status bar at the bottom of your buffer, which indicates you are in "Electric Debug Mode". When this mode is enabled, the buffer is made read-only, and all sorts of great debugging commands are reduced to a single keystroke. Hit Control-? to list them. Space steps through code, descending into function calls, 'n' steps over function calls, 'h' runs the code up to the line the cursor is in, etc. A very powerful command is 'x', the all-purpose examine command. Use it on a variable or expression near the cursor, or mark a region and 'C-u x' to examine it. If you pause after hitting 'x', up will pop a list of examine commands available with one more keystroke. This list is even configurable... see the manual. E.g. x s invokes structure help, x d prints the dimensions, etc. If you like electric debug mode, consider turning it on for errors as well as breakpoints, with:

(setq idlwave-shell-automatic-electric-debug t)

=====