
Subject: Re: IDLWAVE 4.6

Posted by [dominik](#) on Mon, 04 Dec 2000 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

>>>> > "CM" == Craig Markwardt <craigmnet@cow.physics.wisc.edu> writes:

CM> Hi Carsten (and JD?)--

Yes, definitely: and JD. He's my constant checkup and source of inspiration. He is also the one guilty for at least half of these options ;-)

CM> However I can't get a handle on this debugging stuff, and that's
CM> the thing I want the most! I am the first to admit that debugging
CM> under IDL is really not very satisfactory (even with my own
CM> DXDEBUG, which I use sporadically). In IDLWAVE I tried to set
CM> breakpoints, but they don't seem to take effect. Is it because I
CM> type ".RUN myscript.pro" by hand in the shell?

No this should not make a difference, I hope.

CM> I'm not big on three-key control sequences, so it doesn't come
CM> naturally to me to do C-c C-d C-b. I will remember if I need to
CM> though.

Well, firstly *all* the debugging related stuff starts with C-c C-d.
This is a hassle initially, but your fingers will learn to do these
two all by themselves.

After that, remembering gets a lot easier: If I omit the C-c C-d, it
is (highlighting the mnemonic letter)

C-b set a [b]reakpoint in the current line
C-d [d]elete the breakpoint in the current line
C-a clear [a]ll breakpoints
C-r [r]esume execution to next breakpoint
C-h continue to [h]ere
C-z [z]ap corrupted state and reset IDL (does RETALL etc)
C-p [p]rint the value of the variable at point
C-up (Thats the up arrow) Move up the calling stack
C-down (Thats the down arrow) Move down the calling stack

These last two are for you Craig, since you implemented this walking
the calling stack first and showed me how it could be done.

CM> Under Microsoft debuggers it used to be easy to "mouse" a
CM> breakpoint, and the program would run to that point immediately.

CM> I think GDB has something similar. Can I do that with IDLWAVE?

That would be `C-c C-d C-h'. I can show you how to bind this to a mouse event if you like, but not now. Emacs is best with keys, and in the end this is faster since you will not have to move your hands back and forth between keyboard and mouse.

That said, there is lots of mouse support in IDLWAVE which you cannot see. Under XEmacs, IDLWAVE has a toolbar just like every other developing environment. GNU Emacs users will have to wait for Emacs 21 to get the toolbar as well.

But I am getting too long again. Now here is what you and David are asking for (I hope). Mini Tutorial version 1.

1. Edit your source file (and put a syntax bug in, to be sure).
2. Launch the shell with C-c C-l. Emacs should pop up a new window or split the current to show the shell interaction buffer.
3. In the source buffer type C-c C-d C-c to compile it.
In the shell interaction buffer you see that magically

.run filename

is typed. If your program had a syntax error, that line will be highlighted and the cursor positioned so that you can fix the typo. Repeat C-c C-d C-c until this works.

4. Move the cursor to the line where you want to have a breakpoint.
Type C-c C-d C-b. The line should be highlighted in pink. XEmacs users get the familiar red dot.
5. Switch to the shell window with C-c C-s and execute the program by typing the name of the compiled procedure. IDL should run it and stop at the breakpoint. If not, I would like to know about it.
This may be difficult to see since there are now two different highlightings of the same line: red for the breakpoint and green for the stop position. Emacs displays only one of them. XEmacs (and Emacs 21) make this easier.
6. Hold down the SHIFT key while you click with the middle mouse button on variables in the source window you would like to check.
The shell prints their value.
7. Resume execution with C-c C-d C-r or step through the program with C-d C-d C-s

8. When you are done debugging, remove the breakpoints with C-c C-d C-a.

CM> So my question is, to JD or Carsten: If there were *two* or

CM> *three* top things to remember about IDLWAVE's shell interaction,

CM> including debugging, what would they be? And are there caveats to

CM> remember?

Maybe the above was *not* after all what you wanted. Two or three things to remember? JD?

CM> Thanks, and sorry for being an idiot, Craig

I'd say, I need to improve the docs if you cannot understand how it works.

- Carsten

--

Carsten Dominik <dominik@astro.uva.nl> _ /
Sterrenkundig Instituut "Anton Pannekoek" |X|
Kruislaan 403; NL-1098 SJ Amsterdam /| |\ _ _ _ / \
phone +31 (20) 525-7477; FAX +31 (20) 525-7484 _|o|_ / ~ ~ _ / ~ ~ ~
