Subject: Re: IDLWAVE 4.6
Posted by John-David T. Smith on Mon, 04 Dec 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:

> I really appreciate what you're doing here. I'm actually kicking
> myself because I downloaded 4.5 over the weekend. Doh.
>
> I have to admit though that I'm a bit baffled by all the options in
> these newer versions. I can figure out the Routine Info things, and
> eventually I will get my preferences plugged in there (finally I know
> how to indent my main levels to 2 characters)!
>
> However I can't get a handle on this debugging stuff, and that's the
> thing I want the most! I am the first to admit that debugging under
> IDL is really not very satisfactory (even with my own DXDEBUG, which I
> use sporadically). In IDLWAVE I tried to set breakpoints, but they
> don't seem to take effect. Is it because I type ".RUN myscript.pro"
> by hand in the shell?
>
> I'm not big on three-key control sequences, so it doesn't come
> naturally to me to do C-c C-d C-b. I will remember if I need to
> though. Under Microsoft debuggers it used to be easy to "mouse" a
> breakpoint, and the program would run to that point immediately. I
> think GDB has something similar. Can I do that with IDLWAVE?
>
> So my question is, to JD or Carsten: If there were *two* or *three*
> top things to remember about IDLWAVE's shell interaction, including
> debugging, what would they be? And are there caveats to remember?
>
> Thanks, and sorry for being an idiot,
> Craig

OK, I'll bite, but Carsten can of course speak with the final
authority. Here I'll assume you're using GNU emacs (not xemacs, which
is even a bit fancier), with a unix system.

0. Take a deep breath and realize that IDLWAVE shell interactions don't
do anything you couldn't do in IDL itself. It just does it much faster,
and with more features (e.g., highlighting first line with a syntax
error, breakpoints, etc.). You can let IDLWAVE do much of the grunt
work for you, or you can do it yourself, if you enjoy that kind of
thing. An example is setting a breakpoint. You can look up the line
number, carefully take note of the file name, and enter by hand "IDL>
breakpoint,'/path/to/foo.pro',42", or you can let IDLWAVE do all that
for you with a simple keystroke. There is no trouble mixing these.

1. Read
http://www.strw.LeidenUniv.nl/~dominik/Tools/idlwave/idlwave .html#SEC2,
also known as "IDLWAVE in a nutshell". Don't be afraid of the .emacs settings you see there. If the sight of lisp frightens you, do not feel that you are some sort of inferior being. It frightens most people (me included), although not Carsten, who occassionally writes me notes like:

((mailto (smith . jd) 'please find attached
(quote (, file))))

2. Build a user catalog. Carsten made this really easy: it examines your IDL_PATH and uses those directories as the starting point. You can get it from the IDLWAVE menu->Routine info. IDLWAVE then scans all of your routines or libraries (e.g., the Nasa Library), and can give you all kinds of information about calling options, etc. This isn't too relevant for debugging, but important nonetheless.

3. If you don't like C-c C-d C-b, and your alt key is free, you can switch all C-c C-d C-x to A-x with idlwave-shell-activate-alt-keybindings, or set it to any other prefix key with idlwave-shell-prefix-key. You can also of course redefine any keys you like, in your .emacs. For example, suppose you'd like F5-F8 to be debugging commands. You could simply add:

```
(local-set-key [f5] 'idlwave-shell-break-here)
(local-set-key [f6] 'idlwave-shell-clear-current-bp)
(local-set-key [f7] 'idlwave-shell-cont)
(local-set-key [f8] 'idlwave-shell-clear-all-bp)
```

to your idlwave-mode-hook and idlwave-shell-mode-hook. See the manual for examples on using these hooks (it's pretty much cut and paste). Now I can use F5 to set a break (the line is highlighted in a color of my choosing, or if you have xemacs or some future version of emacs, a little glyph appears next to the line), F6 to clear it, F7 to continue past if I've hit a break. F8 to clear them all. You should never live with key bindings you aren't comfortable with... it's so easy to change them.

4. The mouse bindings shift middle-click and control-shift middle click for printing or help on variables you click on are very important (for me at least) Here's an example fast debugging cycle:

a) Set a breakpoint with C-c C-d C-b (or F5, or whatever).
b) .run with program with C-c C-d C-c (or again, the key combo of your choosing). You can also do this by hand... no problem.
c) If there are any compile errors, the first offending line will be highlighted. Fix your dumb mistake, repeat b).

d) Run your command (up arrow in the shell - or setup C-c C-d C-y to send and execute "myfunnyroutinewhichisbuggy, a, b, c, .1").
c) When stopped at the breakpoint (still highlighted), shift mouse-2 on a few things to find out what their values are. C-c C-d C-up (or whatever -- I use C-keypad+ or -) to skip up and down the calling stack above your breakpoint. You are transported to that call in the code (which is highlighted). Use the mouse print and help freely... variables are teleported from other levels (this is where idlwave puts the idlde debugger to real shame).
d) Convince yourself you've fixed the bug. C-c C-d C-d to delete the breakpoint, repeat b).

Other hints (sorry this is so long). Sometimes you want to set your own breakpoints, by hand (but remember that you can set "after breakpoints" trivially - e.g. C-u 3 C-c C-d C-b to get "breakpoint,'/home/jdsmith/foo.pro',10,after=3"). This is no problem. Simply use M-x idlwave-shell-bp-query to update IDLWAVE's notion of breakpoints, if you want the highlighting to be accurate (but as soon as you run into the break IDLWAVE will figure that out). Guess what? I bound that to a key too.

The take home message is that IDLWAVE debugging (and in general) can be whatever you want it to be, thanks to the ease with which emacs can be modified, and the tremendous underlying functionality IDLWAVE provides. If you have trouble getting your breakpoints to take, let Carsten know, but I suspect it was a simple misunderstanding.

JD
