
Subject: Re: Seeing Source w/ Debugger
Posted by [peter](#) on Fri, 24 Jun 1994 00:16:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

Alan Tai (tai@rockola.larc.nasa.gov) wrote:

: I'm having great difficulty setting breakpoints and figuring out
: EXACTLY what line I'm on while stepping through my IDL 3.5 programs.
: IDL won't let me set breakpoints until after I've run the program.
: Even then, the breakpoint line numbers don't correspond with the
: line numbers I see in vi. This is very frustrating.
: Ideally, I'd like a full-screen debugger, but just displaying
: the source for the current line would be fine for me (like in dbx
: and gdb). I am considering writing a program to preprocess another
: program for debugging. The preprocessed program would load in its
: full source into an array, and at the beginning of every line that
: isn't a continuation, have "print, source(i) & " where i is the line
: number. Has anyone done this? Any suggestions? Thanks.

: Alan

Yes, the debugger has a lot of problems.....

For your particular complaints, however, things are maybe not so bad. Firstly, the routine needs to be compiled with a .RUN statement before you can set a breakpoint. You do not need to execute the program. Secondly, it seems that execution is stopped after the line on which the breakpoint is set, so you need to set it back one line.

My biggest complaint is that the debugger can only find routines that are in the directory in which IDL (or PV-WAVE for that matter, they both have the same problem) is started. Using CD does not help. If the routine lies in another directory, then the breakpoint routine reports not being able to find the relevant source statements. Since I like to keep IDL routines in a different directory than my data (and for large projects, even split the source across multiple directories), and it is most convenient to start IDL in the data directory, I have to kill it and restart it elsewhere when debugging. Similarly, it makes having a library of routines stored in a different directory somewhat inconvenient -- you cannot set breakpoints into any routine in the library.

I have told RSI and PVI of my dissatisfaction with breakpoint, to no avail so far (they didn't seem to be aware that it was a problem). If anybody knows of a workaround to the above, please let us all know.

Peter

Subject: Re: Seeing Source w/ Debugger
Posted by [robijn](#) on Fri, 24 Jun 1994 13:02:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <CrvKsA.Lro@hpl.hp.com>, Peter Webb <peter@hpl.hp.com> wrote:

>
> My biggest complaint is that the debugger can only find routines that
> are in the directory in which IDL (or PV-WAVE for that matter, they both
> have the same problem) is started.

That is not true, although I must admit that the filename argument interpretation of BREAKPOINT could be improved. It all depends how you compiled the routine. If you type:

```
.run [path/]func[.pro]
```

([] indicate optional parts) and the file [path/]func.pro exists, you should use that name in BREAKPOINT. Otherwise you should prepend the directory where the file resides exactly as that directory appears in the !PATH variable.

It would be a good idea to let BREAKPOINT search the path in the same way as .RUN does. Better still, make the path-search code available as separate routine as well. I wrote an IDL routine called 'which' that does the trick for DOS and Unix systems.

If you place 'which.pro' somewhere in the IDL path, you can set your breakpoints as:

```
breakpoint, which ([path/]func.pro), ...
```

Frank

--

```
_____/_____  
/ / / Frank Robijn Internet: Robijn@Strw.LeidenUniv.NL  
/___/___/ Sterrewacht Leiden Bitnet: Robijn@HLERUL51  
/ / \ Phone (31) 71 275841 Local: Robijn@HL628  
/ / \ Fax : (31) 71 275819 Snail: P.O.Box 9513, 2300 RA Leiden,  
The Netherlands
```

Subject: Re: Seeing Source w/ Debugger
Posted by [landers](#) on Fri, 24 Jun 1994 13:08:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

PV-WAVE is working on a debugger. It's still (I think) in beta test. It's basically just a tcl wrapper around a wave session, but it does a pretty good

job of setting breakpoints (by clicking source code lines) and stuff like that. It does have some limitations, but most are no different than you get with the command line version.... I understand that there are more improvements in the works (like linking things more directly into the WAVE kernel).

You can pick up a copy on their ftp site, <ftp://ftp.pvi.com/VNI/> (or something very much like that).

;Dave
