Subject: Re: IDL excels in debugging??? Do you know something I dont?
Posted by zawodny on Mon, 22 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

In article <3pg7pi$b1u@newsstand.cit.cornell.edu> patterso@astrosun.tn.cornell.edu (Tim Patterson) writes:
> Mark Rivers (rivers@cars3.uchicago.edu) wrote:
>
> : >In fact, I would love to read a discussion of what other people are using
> : >for debugging techniques / tools.
>
> : Why not just use the STOP statement in your routine? It stops IDL, leaving you
> : at the command line, allowing you to examine all variables' values, etc.
> : without generating the error. Once you are done examining variable, etc. you
> : can continue on by just typing .CON.
>
> I just use control C and .con (or xmanager in a Windows situation)
> to debug my code. But it's not pretty :)
> I'd love it if the step command would actaully just oprint the relevant
> line of code to the screen so I knew where I was in the code.
>

Control C is a bit difficult to use in practice since you really
cannot control where you stop. Try stopping in a routine that spends
most of its time in other subroutines and you'll see what I mean.

Editing a proceedure to add a STOP gets tiresome after awhile. My
biggest gripe when it comes to debugging IDL is that the reported line
number is usually wrong. This may be due in part to my programming
habits, but it seems that whenever I use an @file to include common
blocks and/or 'standard' code or if I have mulltiple statements on the
same line (using the & operator) IDL cannot figure out where the
offending line is when it bombs. I end up adding a series of lines
like

```
 print,'I am at A'
```

and recompiling and rerunning the code to narrow down where the error
is. I guess it would be nice to have true interactive breakpoint
setting. A routine like

```
 SET_BREAK,module_name,line_number {,/cancel_break}
```

which could be called interactively either before or during (after a
STOP or programming error halts execution) would be really useful and
speed code development (for me at least). I have no idea what this
would do for execution speed (IDL and other interpreted languages are
slow enough as it is [reading things like "A well written IDL program

can actually run faster than a C program" has kept me from learning
C, but I digress further]).
--

Joseph M. Zawodny (KO4LW) NASA Langley Research Center
Internet: j.m.zawodny@larc.nasa.gov MS-475, Hampton VA, 23681-0001
TCP/IP: ko4lw@ko4lw.ampr.org Packet: ko4lw@n4hog.va.usa.na