Subject: Re: Workaround for "Program unit has too many local variables"?
Posted by davidf on Wed, 23 Oct 1996 07:00:00 GMT
View Forum Message <> Reply to Message

Charlie Zender <zender@ncar.ucar.edu> writes:

> IDL 3.6 will not compile a procedure that has grown rather large.
> the error message is:
> "Program unit has too many local variables"
> is there a workaround to this which does not involve rewriting
> the procedure? e.g., is there a system variable i can set to
> increase the # of local variables allowed?

There is no system variable that can be set to increase the number of local variables allowed in IDL.

But it would be a strange program indeed that couldn't be made more modular (and probably improved) by splitting functionality off into utility procedures and functions.

Besides, doing this will get you in shape for writing good widget programs, where small modules with specific tasks lend themselves to programs that are easy to extend and maintain.

BTW, I'm told that many of these kind of limitations will go away in the next version of IDL, due out in 1997.

David

--
David Fanning, Ph.D.
Phone: 970-221-0438
Fax: 970-221-4728
E-Mail: davidf@fortnet.org

Subject: Re: Workaround for "Program unit has too many local variables"?
Posted by Liam Gumley on Wed, 23 Oct 1996 07:00:00 GMT
View Forum Message <> Reply to Message

Charlie Zender wrote:
> IDL 3.6 will not compile a procedure that has grown rather large.
> the error message is:
> "Program unit has too many local variables"
> is there a workaround to this which does not involve rewriting
> the procedure? e.g., is there a system variable i can set to
> increase the # of local variables allowed?

You probably need to increase the memory used for compiling programs.
This is done with the .SIZE command, e.g.

.SIZE 60000 20000

which resizes the code and data areas to 60000 and 20000 bytes
respectively (the default is 32768 and 8192 bytes respectively). There
is an upper limit for the code and data size for each O/S. This will
help you out up to a point. If you have lots of arrays defined in your
program (like you would do with DATA statements in FORTRAN), then you
might not be able to increase the code and data areas to a large enough
size (the program will still fail to compile). In this case, you will
have to store your variables in data files, and read them on start-up. I
know this can be inconvenient, but it's just the way IDL works. The
other option is to break up your program into a number of smaller
procedures/functions, since every procedure/function has it's own
code/data space.

Cheers,
Liam.
