Subject: Re: Global variables and IDL
Posted by davidf on Sat, 24 Apr 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Folks,

In returning to this newsgroup after a short absence I have
been shocked and appalled to see evidence of the ever
quickening pace of moral decay in the world today. I
refer, of course, to the large number of IDL programmers
confessing to the use of COMMON blocks in their programs.

I, too, used a COMMON block--once. I think I stored the
seed of a random number in it until I realized I could just
as well include the seed in my info structure along with all
the other stuff I needed in my program. And there is
evidence that even more programmers might soon come forward
with the same confession. It is hard to imagine how
it could be otherwise, what with RSI supplying example
code liberally sprinkled with COMMON blocks and most
of us cutting our teeth on FORTRAN programs. (Is it
my imagination, or are most of the people confessing to
COMMON block usage at least middle aged?)

But, alright. The evidence is overwhelming--even
to me. There are some very good reasons why a COMMON
block may be used for some good purpose. Just because
I've never stumbled onto one in the course of my own
programming doesn't make it any less true. It's
probably because I don't have enough programming
imagination to see how clever using a COMMON block
would be.

But I would like to correct a few misconceptions about
COMMON blocks, if I may.

Martin Shultz writes:

> But here are two examples where I used common blocks --
> and I would be happy
> to learn how I could have avoided them:
>
> * in my EXPLORE tool (which can handle several "instances" at least if
> opened from within), I use
> a common block to keep track of the drawing windows that have been
> opened and used.

There is absolutely no need for a COMMON block in this instance.

Widget programs can easily keep track of drawing windows through
the used of draw widget IDs, even draw widgets that are detached
from the main window. See, for example, the SLICE program on my
web page, which can spawn any number of copies of itself, or--for
a simpler example, the ZIMAGE program which can tell if the zoom
window is open and on the display or not. If it is not on the
display (it had, perhaps, been killed by the user) the program
simply creates another.

   http://www.dfanning.com/programs/zimage.pro

Martin goes on to write this, which scares me very much indeed:

> This is
> needed to kill a window when the associated widget is closed as well as
> to open a new window
> for a new widget instance. At first it would seem that I could simply
> use the /free keyword to WINDOW and store the window number locally with
> the widget, but I have to close *all* windows when I exit the program.
> Should I use the event notification method?

Using normal IDL graphics windows from within a widget program
is a *terrible* idea! (No offense to Martin, who I know is a VERY
good IDL programmer.) But if you display graphics in normal IDL
windows you have absolutely no control over them from within
your widget program. You would, indeed, probably need a COMMON
block to account for them. If you need to display graphics inside
a widget program I can say categorically, without exception, that
you need a draw widget in which to display them! (I'm not saying
it can't be done. I'm saying you don't want to do it.)

Cheers,

David

P.S. After thinking about it for a long time I just
put a COMMON block back into one of the programs on
my web page. (I won't say which one, because I'm not
certain that I am going to leave it there.) Whew!
It wasn't as bad as I thought it was going to be. ;-)

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155
