Subject: Re: global variables in IDL
Posted by John-David T. Smith on Fri, 28 Dec 2001 16:22:23 GMT
View Forum Message <> Reply to Message

Pavel Romashkin wrote:
>
> I agree with Kristine. Working around Commons writing a global variable
> code, if you use Xmanager, is kind of silly becasue Xmanager uses Common
> blocks in the first place.

This objection, though oft-repeated, is somewhat unfounded, I think. The chief problem with common blocks for global state data is that only a single such block exists. For example, if you store the widget id of a button in:

```
common mywidget, mybutton
mybutton=widget_button(base,value='Foo')
```

Then `mybutton' will be the *only* button that can exist for your program in a single IDL session. You won't be able to invoke your program more than once at a time. Actually, you will, and that single button variable slot will get overwritten, and very bad things will happen.

XManager does use common blocks, but, by design, it uses them intelligently. Instead of overwriting common block variables, it maintains global lists of all the widgets it needs to manage, extending and pruning the lists as necessary. It's an example of common block usage which doesn't prevent multiple copies of a single widget program from running. (Here I gloss over the subtlety that widget programs run with /NO_BLOCK are actually handled by the command-line loop code, and not XManager at all.)

The fact that common blocks, in their basic usage, can cause this subtle problem doesn't imply that this problem is endemic to their use: XManager is an example of a more intelligent setup which capitalizes on, rather than suffers from, the single global instance status of common block variables.

JD