Subject: Re: Common or not common
Posted by morisset on Mon, 17 Nov 1997 08:00:00 GMT
View Forum Message <> Reply to Message

From meinel@aero.org who can't post:

Could one of you forward this to comp.lang.idl-pvwave if you think
it is worth it. For some reason it refused to post.

davidf@dfanning.com (David Fanning) writes:
>
> Mr. Morisset writes:
>
>> I can't imagine passing 25 or more parameters to the routines.
>
> Nor can I. Although event handlers often need at least this
> much information to function properly. You can pass the
> information by common blocks, which has many, many limitations
> or you can pass the information via "pointers". I prefer the
> latter.
>

Talk to C programmers. Pointers are a two-edged sword. You can do
pretty neat stuff with pointers, but they are also the source of
about half of the bugs.

>
>> And what happens when I decide to add one other parameter to
>> one routine?
>
> More to the point, what happens when *you* and your common
> blocks have to add another parameter. I don't have to exit
> IDL, that is for sure. :-) Nor do I have to update all of
> my program modules. I just add the parameter to my structure.
>

I currently use commons and named structures. There are times
when I don't know the size of some arrays until well after the
structure is created. However, I just read in the manual (TFM)
that anonymous structures _can_ be modified after creation.
I'll have to give that a try...

Which brings up an IDL gripe -- why do named structures have
different properties than anonymous structures? And why aren't
structure properties consistent between types? According to TFM:

"Once defined, a given named structure type cannot be changed."

Yet in the given example, NAME is defined as a zero-length string, but subsequent assignments can have any number of characters. Are strings the only variables that can be changed in a named structure? Why are strings handled differently? Should I make everything in my named structure a string and then convert to some other type at the appropriate time? Inquiring minds want to know.

>> Can somebody explain me what I miss?
>
> You miss the ability to have more than one version of your
> program running at any one time. How good is that great image
> processing routine if you can only process one image at a time?
>

That's odd. I have a "great image processing routine" that uses commons and I can process more than one image at a time. Maybe I'm just "smarter than your average bear" (for those of you who didn't watch Hanna-Barberra cartoons, that was a quote from Yogi Bear).

Ed
meinel@aero.org