
Subject: Re: Common or not common
Posted by [davidf](#) on Fri, 14 Nov 1997 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Liam Gumley (liam.gumley@ssec.wisc.edu) answered
Mr. Morisset's original question better than I could have
myself. :-)

But let me make just a few other points.

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.

> 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 use parameters when I want to deal with procedure or
> function I want to apply to different variables. But when
> in every routines/functions of the program
> the spectrum, lambdas, corrections of
> different types and a lot of other things,
> have always the same name, it's easier and safer
> to pass them via a common, no?

No. I think it is a great idea to always name things the
same. I wouldn't do it any other way. User values are
completely safe because each event is processed individually.
If you take the info structure out with a No_Copy at the top
of your event handler, and put it back with a No_Copy at the
bottom (you *do* use No_Copy don't you), then you NEVER get
in trouble. Ain't possible.

> 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?

Liam writes:

> PS: David I'll take that case of beer whenever you're ready.

If he buys my book, I'll mail you the case of beer. :-)

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
