Subject: Re: Global variables and IDL
Posted by steinhh on Wed, 14 Apr 1999 07:00:00 GMT
View Forum Message <> Reply to Message

rmlongfield writes:

> HI All, I've been thinking about this question myself lately.
> As far as I understand, the use of COMMON blocks is not
> recommended when working with widgets, but is ok otherwise.
> I have heard that there may be a problem if there are too
> many variables, but 20 seems ok to me.

Well, I'd say that common blocks should only be used when there is *no* conceivable way you'd *ever* want to have more than one version (instance) of your data. You *may* want to generate a slightly altered data set and view that along side the original data set at some point in time.

As such, only a few situations *really* qualify -- like the common blocks used (internally) by XMANAGER. There's no way you'd ever want to have *two* lists of managed widgets.

Another extremely good example is the list of singleton objects used by J.D. Smith's singleton abstract class.

A singleton (sub)class is defined by the fact that only one instance (object) of that class should exist at any one time. The singleton INIT method needs to know whether or not an object of the same class already exists. In order to do this, one needs to have a list of existing singleton objects. There is no way you'd ever want to have *two* such lists (defeats the purpose...), so you can safely put it into a common block.

In general, common blocks are good for storing information used in a (globally available) system that keeps track of things... The "things" themselves should not be put into common blocks :-)

Having said this, I must confess that yes, I have used common blocks for other purposes, but that's only for very small, very experimental programs.

> I have an image processing tool which uses a base data set
> but many different widget modules, each with its own TOP
> LEVEL WIDGET. I would like to have access to this data from
> whatever or whichever widget I am working in. The data sets
> I read are created in the middle of my processing, so it
> would do no good to read it at the start and have a general

> pointer that can be included in each Top Level Widget.

You should take advantage of the fact that you can create a pointer without pointing it at anything:

```
storage = ptr_new()
xstartprog,storage=storage ;; Data will be available later...
xutility,storage=storage   ;; Ditto.
xreaddata,storage=storage  ;; This one will read the data...
```

Regards,

Stein Vidar