Subject: Re: TEMPORARY (was Re: Fragmented memory with IDL)
Posted by webb on Sat, 22 May 1993 07:22:52 GMT
View Forum Message <> Reply to Message

jdlb@kukui.ifa.hawaii.edu (J-F Pitot de La Beaujardiere) writes:

>
> (deleted description of operation of TEMPORARY)
>
> TEMPORARY, whose effects are PERMANENT, might perhaps be more aptly named
> COPY_AND_DELETE. :^)
>

Actually, TEMPORARY is more like DONOTCOPY_AND_DELETE, since it specifically does not make a copy of the right hand side.

One somewhat odd but sometimes useful way to use TEMPORARY is to rename variables. Running

```
   A = TEMPORARY(B)
```

will effectively rename B to A, without incurring any memory overhead. Why might you do this? It's one way of making widget applications easier to write. Suppose you have you favourite widget application on the screen, and when you click the button you want something to happen to a variable VAR whose name is unknown to you when you started the widget. Inside the widget event handler, always operate on, say ACTIONVAR. Make ACTIONVAR a member of a common block declared within the event handler, then run

```
   ACTIONVAR = TEMPORARY(VAR)
```

from somewhare else that also declares this common block, and now you can work on your new variable. In the application I am working on, I keep a bunch of data sets in named variables, then use a selection widget to swap them in and out of my ACTIONVAR variable without incurring memory penalties. This keeps the event handling code simple yet flexible.

Of course, the proper way to do this would be to allow some kind of pointer type, then all this fooling around would be unecessary. Anybody else miss pointers? Maybe I assume to much, but it seems that the IDL interpreter must have pointers to variables readily (or not so readily) available?

Peter