Subject: Re: Passing info and destroying widgets...
Posted by Liam Gumley on Mon, 21 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

David Fanning <davidf@dfanning.com> wrote in message
news:MPG.11d85d6ea2cc37d9897de@news.frii.com...
> Liam Gumley (Liam.Gumley@ssec.wisc.edu) gives us an
> example of a program that can record the last instance
> of a button push in a non-blocking, non-modal widget
> when he writes:
>> Here's an example which works in non-blocking mode:
>
> No question it works. But I would argue that it works
> for all the wrong reasons and is a *terrible* programming
> practice in almost every instance. I mean, you can write
> an object method that returns a data pointer too, but
> by doing so you violate every tenet of good object programming
> practice, in which the data should be encapsulated and
> unseen by the outside world. Sucking the pointer out of
> a widget program, except perhaps in the hands of just the
> best programmers, is a practice that is guaranteed, it
> seems to me, to get most of the rest of us in a hell of
> a lot of trouble.

Well I guess I'll have to say that my example only demonstrates that it
*can* be done, not that it necessarily *should* be done.

> If you are going to recommend this, at the very least
> teach people how to use HEAP_GC at the same time because
> I'll bet a ton of money there will be leaking memory
> right and left!

As the saying goes, there was good news and bad news in the release of
IDL5.0:

The good news was, it included pointers.

The bad news was, it included pointers.

My personal programming philosophy is to only use pointers where absolutely
necessary (e.g. for retaining information when a widget dies, or for storing
structure elements which are of unknown size and type until runtime). I've
never used HEAP_GC in any of my programs; I rely on simple wrappers like
those I posted at http://cimss.ssec.wisc.edu/~gumley/pointers.html to
prevent me from getting into memory leakage problems.

> As for me, I'm sticking to widget programs that clean
> themselves up and don't leave the user holding the bag,

> er, pointer. :-)

I agree that widget *users* (e.g. of David's fine XCOLORS routine) don't want to know (and shouldn't be allowed to touch) the internals of "shrink-wrapped" widget routines. But I think any IDL application developer will eventually run across cases where items like information structures need to be shared between widgets which are performing closely related application-specific tasks.

Cheers,
Liam.

---