Subject: Re: Recursive object destruction, Was: IDL Shapefile Object
Posted by David Fanning on Sat, 01 Dec 2001 15:58:21 GMT
View Forum Message <> Reply to Message

Richard Younger (younger@ll.mit.edu) writes:

> On the down side, there's some issues with the recursive passing of
> structure members. Since structure members are always passed by value,
> I'm a little worried that using this destroy routine on, say, structures
> with large arrays in them would use up more memory (and time spent
> copying data to be destroyed anyway) than is reasonable. Does anybody
> have suggestions on this? Am I right to be worried?

Don't know. I've been worrying all morning about
whether I was going to have to write an example
data set to test this. I've decided to live with
uncertainty. :-)

> So I don't know if some of the choices I made are the correct ones. Am I
> using _REF_EXTRA right?

Nearly. I would change this line:

  'OBJREF':   OBJ_DESTROY, thing, _REF_EXTRA=extr

To this:

  'OBJREF':   OBJ_DESTROY, thing, _EXTRA=extr

The final call in a chain will have to be done with
_Extra.

> Is there any point to checking whether I'm
> dealing with a (structure member) copy or the real McCoy?

If we are talking about pointers and objects in structures,
I don't see that it makes any difference at all if you
are dealing with a copy or the real thing. The copy points
to the real thing. That's what is important, it seems to me.

> What about
> undefining vs. setting the variable to zero and redundant destruction?

Undefining verses setting a variable to zero is more a
matter of style, I think, than anything of substance.
>
> Are there flaws that I haven't seen?

Oh, there are *always* flaws you haven't seen yet.
That's part of the mystery of programming.

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155
