
Subject: Re: passing object references
Posted by [davidf](#) on Wed, 11 Nov 1998 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Philip Aldis (teal@dera.gov.uk) writes:

> Well what a surprise, the mistake was in the place that was far too obvious to
> even consider, that indeed the object wasn't valid, and that I had deleted it
> myself.

"The trick, Dear Watson, is to eliminate everything that cannot possibly happen. Then, whatever is left must be the truth, however improbable."

--Sherlock Holmes

Subject: Re: passing object references
Posted by [philip aldis](#) on Wed, 11 Nov 1998 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

> Then I would like to see how you are storing it in your info structure.
> Or, another possibility, did you destroy the first widget program to
> run the Animator program? If so, your clean-up routine on the first
> program destroyed the object. If this is happening, you may have to
> make an actual copy of the object itself and pass that. I just added
> a tip on my web page about how to do this:
>
> http://www.dfanning.com/tips/copy_objects.html
>

Well what a surprise, the mistake was in the place that was far too obvious to even consider, that indeed the object wasn't valid, and that I had deleted it myself. I had called the animator program, and then in the next line of code destroyed the loader widget, which destroyed the object, but did so only after the first lot of animator code had executed. Thus the object was onlt invalid when the vent handlers were being used.

The only problem with copying the object is that it has another object, and many pointers, and so will be a little fiddly. So, what about having a flag in my info structure which is only set when I have called the animation prog. and cleanup only destroys the object if flag is set to 0, i.e. if the user quitted out before calling animation program - bit messy, but it should work.

cheers,
Phil Aldis

Philip Aldis
Room B803/4
DERA Malvern tel:0044 1684 897241
St.Andrew's Road fax:0044 1684 894185
WR14 3PS
England

The information contained in this e-mail and any subsequent correspondence is private and is intended solely for the intended recipient(s). For those other than the intended recipient(s) any disclosure, copying, distribution, or any action taken or omitted to be taken in reliance on such information is prohibited and may be unlawful.

DERA disclaimer:

The views expressed above are entirely those of the writer and do not represent the views, policy or understanding of any other person or official body.

Subject: Re: passing object references
Posted by [davidf](#) on Wed, 11 Nov 1998 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Philip Aldis (teal@dera.gov.uk) writes:

> I having a bit of trouble with my object references and passing them
> round. I have a feeling it's to do with passing by value and by
> reference, but I haven't had time to test this out yet, so before I
> start ripping my code apart - I'll see if what I suspect is true.

Hang on, hang on. Don't start ripping code just yet! :-)

> I have two widget programs which deal with loading some data in and then
> the second one does an animation. The loader program creates the object
> and loads the data, using the methods and then passes the object with
> its data loaded in to the animator widget program which calls the object
> methods for animating. I pass the object in as a keyword.
>
> Previously the object had been part of the info structure of my loading
> widget program, and then I simply call my animator program
> ANIMATOR, object = info.object

Now, the only things that are passed by reference in IDL are variables. Everything else, and this includes structure de-references like we have here, are passed by value. So it is a **copy** of the object reference that is going into the Animator program, clearly. But I would argue that that is exactly what you want to happen and, in fact, is the **point** of heap variables. The reference always points to the actual object, located on the heap.

>fine so far. Now I've tested the object I send, the object I
> receive and they are all valid. Within the main animation program - i.e.
> before the call to xmanager, I've got the user value and tested the
> validity of the object, and yep you guessed it, it's valid. BUT when I
> get the user value in my event handlers, the object is suddenly no
> longer valid.

Then I would like to see how you are storing it in your info structure. Or, another possibility, did you destroy the first widget program to run the Animator program? If so, your clean-up routine on the first program destroyed the object. If this is happening, you may have to make an actual copy of the object itself and pass that. I just added a tip on my web page about how to do this:

http://www.dfanning.com/tips/copy_objects.html

> Is it something to do with the fact that I have sent it in a keyword?

I doubt it. If you are trying to get information **back** from keywords, you can have problems. But that doesn't sound like what you are doing here.

Cheers,

David

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

Note: A copy of this article was e-mailed to the original poster.
