Subject: Re: Destroying objects
Posted by chris_torrence@NOSPAM on Fri, 18 Jan 2013 17:12:48 GMT
View Forum Message <> Reply to Message

On Friday, January 18, 2013 9:33:43 AM UTC-7, Helder wrote:
> Hi,
>
> as usual I'm do things wrong believing I'm not until I wake-up.
>
> So here's the thing. I'm using a structure to describe some objects. In my case these are regions of interest. Within this structure I have real objects. So that my structure looks something like this:
>
>
>
> MyStruct = {ROI_Structure, a:ID_Nr, b:SomeOtherParameter, c:Obj_new()}
>
> I then fill in MyStruct.c by doing something like this:
>
> XROI, MyImage, REGIONS_OUT = XROI_ObjRef
>
> MyStruct.c = XROI_ObjRef
>
>
>
> The thing is that I then replicate this structure when I add new ROIs.
>
> So I end up with say 10 {ROI_Structure} where for each element I did something like this:
>
> for i=0,9 do MyStruct[i].c = XROI_ObjRef[i]
>
>
>
> If and when somewhere later in the program I want to delete object number 5... do I have to take care of using
>
> Obj_destroy, MyStruct[5].c
>
>
>
> Is there no alternative?
>
> Thing is, in this structure I have many pointers and objects...
>
>
>
> I'm I wrong in saying that I have to do the good-boy bookkeeping and free/destroy for each deleted structure element all pointers and objects?

>
>
>
> Thanks,
>
> Helder

Hi Helder,

If you are using IDL 8.0 or later, then as long as you don't have any "circular" references, the automatic garbage collection should free all of the pointers & objects once the variables go out of scope.

However, if you have circular references, where one object references another object, and that other object somehow references the first one, then the garbage collector won't be able to decrement the reference counts to zero.

Another possibility is to use HEAP_GC, which is a big hammer...

Cheers,
Chris
ExelisVIS
