Subject: Re: Finding Memory Leak ?
Posted by davidf on Thu, 21 Sep 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Richard Tyc (Richard_Tyc@sbrc.umanitoba.ca) writes:

> I am starting to lose hair over this one. I recently noticed that when I
> close my program, two pointer variables remain which I am reluctant to
> remove using PTR_FREE, PTR_VALID()
>
> I have checked every line of code where I create a new variable using
> PTR_NEW and I always properly used PTR_Free on it. I also use the DICOM
> method GetValue alot (which returns a pointer to the data) but always use
> /NO_COPY so the pointers point to actual data within the object which should
> get properly removed when the object is destroyed.
>
> Anyone care to give any advice how I can find the source of the bug ? I
> have been trying to print out all the valid pointers (ie. print,
> PTR_VALID()) between Function/Procedure calls to determine when the 2
> pointers come alive but this is very tedious (and so far unsuccessful) !!

I would suspect the Dicom object first. But this could
be (since I know you are writing a log of object code)
that ghost pointer Pavel was going on about last week.
He was telling me to today that RSI confirmed that as
a bug that should be corrected in IDL 5.4, due out
very soon now.

Good luck tracking this down.

Cheers,

David

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

Subject: Re: Finding Memory Leak ?
Posted by ronn on Fri, 22 Sep 2000 00:17:44 GMT
View Forum Message <> Reply to Message

in article 8qduob$eh6$1@canopus.cc.umanitoba.ca, Richard Tyc at
Richard_Tyc@sbrc.umanitoba.ca wrote on 9/21/00 2:32 PM:

> I have checked every line of code where I create a new variable using
> PTR_NEW and I always properly used PTR_Free on it. I also use the DICOM
> method GetValue alot (which returns a pointer to the data) but always use
> /NO_COPY so the pointers point to actual data within the object which should
> get properly removed when the object is destroyed.

Richard,

Two things cross my mind right away.
1. Are you creating a subclass of the DICOM object? If so, you have to call the cleanup method directly from the child object. If you don't then any pointers created by the parent will not be freed.
2. Are you creating the DICOM object more than once so that when you destroy it only the last one is destroyed?

-Ronn

--
Ronn Kling
Ronn Kling Consulting
email: ronn@rlkling.com
"Application Development with IDL"ï¿½ programming book updated for IDL5.3!
Shareware and Freeware at: http://www.rlkling.com/

Subject: Re: Finding Memory Leak ?
Posted by Richard Tyc on Fri, 22 Sep 2000 07:00:00 GMT
View Forum Message <> Reply to Message

At first I suspected the DICOM object but the two leftover pointers point to data that appears to be the 'SIZE' of one of my array pointers (And both pointers have the same value??!!). ie. my global sState or info structure (attached to the base widgets uvalue) has several pointers which point to 3D data I load in. The contents of the leftover pointer is a BYTARR(3) which has the size(data, /N_DIMENSIONS) value ? Looking further into it, I do in several places something like :

szData = SIZE(*sState.3Ddata, /N_Dimensions) and the value of szData is equivalent to the leftover pointer contents. But I never create a pointer to szData so the memory should be cleaned up when the Function/Procedure exits right ??

Still stumped

Rich

ronn kling <ronn@rlkling.com> wrote in message
news:B5F01D44.2A7%ronn@rlkling.com...

> in article 8qduob$eh6$1@canopus.cc.umanitoba.ca, Richard Tyc at
> Richard_Tyc@sbrc.umanitoba.ca wrote on 9/21/00 2:32 PM:
>
>> I have checked every line of code where I create a new variable using
>> PTR_NEW and I always properly used PTR_Free on it. I also use the DICOM
>> method GetValue alot (which returns a pointer to the data) but always
use
>> /NO_COPY so the pointers point to actual data within the object which
should
>> get properly removed when the object is destroyed.
> Richard,
>
> Two things cross my mind right away.
> 1. Are you creating a subclass of the DICOM object? If so, you have to
call
> the cleanup method directly from the child object. If you don't then any
> pointers created by the parent will not be freed.
> 2. Are you creating the DICOM object more than once so that when you
destroy
> it only the last one is destroyed?
>
> -Ronn
>
> --
> Ronn Kling
> Ronn Kling Consulting
> email: ronn@rlkling.com
> "Application Development with IDL" programming book updated for IDL5.3!
> Shareware and Freeware at: http://www.rlkling.com/
>
>