Subject: deallocating ptrs
Posted by R.G. Stockwell on Sat, 11 Jun 2005 15:53:12 GMT
View Forum Message <> Reply to Message

Hi,
I am reading an HDF, and have a template created with the HDF_browser.
The template is a structure that has over 200 pointers deeply nested in
several labrynthical "subnested structure arrays of structure ptr array
structures".
:)

So, I tried to deallocate this monster with a heap_free call, and it doesn't
get
deallocated. (IDL 6.1)
In fact I have for the moment been reduced to a heap_gc call, which seems
inelegant.

So why doesn't heap_free free the heap? It looks like is isn't even trying.
It goes from 238 to 164 ptrs.

Anyone have a nice generic deallocating routine to crawl down the structure.
I actually have an old routine that prints out a structure nicely, that
recursively
steps through each element. It could be modified to test each element and
deallocate it if it is a ptr, but there has to be a routine that already
does that.

Cheers,
bob

Subject: Re: deallocating ptrs
Posted by Michael Wallace on Mon, 13 Jun 2005 20:38:22 GMT
View Forum Message <> Reply to Message

> So, I tried to deallocate this monster with a heap_free call, and it doesn't
> get
> deallocated. (IDL 6.1)
> In fact I have for the moment been reduced to a heap_gc call, which seems
> inelegant.
>
> So why doesn't heap_free free the heap? It looks like is isn't even trying.
> It goes from 238 to 164 ptrs.

I don't know what could be going on here, but the heap_free
documentation includes this little nugget:

HEAP_FREE releases the referenced heap variables in an unspecified order

which depends on the current state of the internal data structure used
by IDL to hold them. This can be confusing for object destructor methods
that expect all of their contained data to be present. If your
application requires a specific order for the release of its heap
variables, you must explicitly free them in the correct order. HEAP_FREE
cannot be used in such cases.

Something like this is the only reason I can think of where IDL would be unable to free everything. You could also try adding the /VERBOSE flag to heap_free to see if anything strange is happening.

-Mike