Subject: Re: Returning A Variable Length struct to IDL from C
Posted by Richard Younger on Tue, 06 Nov 2001 15:58:25 GMT
View Forum Message <> Reply to Message

K Banerjee wrote:

```
>
> Folks,
>
> I wrote a DLM, following the examples in "Calling C From IDL"
> by Mr. Ronn Kling. Here's the simplified layout of what I
> am doing:
```

[...]

```
> theHeader->userHeader = new IDL_STRING[userHeaderArrayLength];
```

[...]

```
> The next two lines are used to create the return value to IDL:
>
> void *psDef = IDL_MakeStruct(NULL, vbHeaderTags);
> IDL_VPTR ivReturn = IDL_ImportArray(1, ilDims, IDL_TYP_STRUCT,
> (UCHAR *) theHeaderActual, releaseMemory, psDef);
>
> where releaseMemory is the function:
>
> extern "C" void releaseMemory(UCHAR *ptr)
> {
>   deleteMem(ptr);
> } // extern "C" void releaseMemory(UCHAR *ptr)
>
```

[...]

```
> The second IDL line causes a core dump due to a sementation
> fault occurring in IDL_MemFree():
>
> #0  0x4008cf58 in IDL_MemFree () at ../../gcc-2.95.2/gcc/cp/exception.cc:343
> 343     ../../gcc-2.95.2/gcc/cp/exception.cc: No such file or directory.
>
> What I think happens is that upon the first call to the function
> idlvbio_get_cube_header(), a certain amount of memory is allocated
> to the IDL variable h. Now the second call to idlvbio_get_cube_header()
> does not cause IDL to delete the memory already allocated to h
> and then reallocate memory to h. More memory is now needed by h
> since there are 15 user header lines in file2.
```

[...]

Hi, K.

I've had troubles in the past allocating and deallocating memory with IDL, so I wouldn't be too surprised if it's a memory issue. It looks to me like you're right, and the problems happen when you try to clean up. Does reassigning h (h=0) cause the same problem?

But I'm a bit puzzled by the 'No such file or directory' error. I haven't used gcc for a couple years, so I'm not confident about the nature of the error message you've included. One could interpret it as IDL_MemFree() is trying to call deleteMem() (or possibly releaseMemory()) and not finding it. I'm not familiar with deleteMem() as a library function. Where is it defined?

But maybe I'm reading the error message too literally, and the problem is the cleanup isn't being called properly or in the correct context. I never give IDL memory allocated by new or delete, as new and delete have been a source of troubles in the past. I never successfully got IDL_ImportArray() to work with memory from new, though I didn't try too hard.

In any case, a workaround, if deleteMem() is being called and working correctly, would be to use IDL_GetScratch() and go through the bother of traversing the IDL variable to get to your scratch memory. Since this particular snippet of code that creates the memory is IDL aware (i.e. export.h defined), it's not too hard.

Best,
Rich

--
Richard Younger