Subject: Re: Returning A Variable Length struct to IDL from C
Posted by Craig Markwardt on Tue, 06 Nov 2001 02:16:31 GMT
View Forum Message <> Reply to Message

Greetings--

K Banerjee <kbanerje2@home.com> writes:

> 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);

Here is your first problem. IDL_ImportArray only works when you import static data, not dynamically allocated. Basically ImportArray only works once per piece of memory per IDL session.

> IDL_STRUCT_TAG_DEF vbHeaderTags[] =
> {
> {"VERS", 0, (void *) IDL_TYP_STRING},
> {"USERHEADER", dims_user_header, (void *) IDL_TYP_STRING},
> {"BYTEOFFSET", 0, (void *) IDL_TYP_LONG},
> {0}
> };

Here (in my opinion) is your second problem: trying to do something too complicated within a DLM function. I personally think that while it's not impossible, trying to manipulate complex data structures within a DLM are very *close* to impossible and are really unnecessary.

In my opinion, the proper approach is to have your DLM function do the dirty work, and construct only the simplest of data, and then have a wrapper function, written in IDL itself, compose it into something more complex. What that means in this case, is to use positional or keyword arguments to return three variables (VERS, USERHEADER [an array of IDL strings], and BYTEOFFSET). Then in your wrapper function you can easily compose these variables into a structure.

Even returning simple data values is not particularly simple, at least not in my experience. Be sure you test your program for memory leaks afterwards.

Craig

--

---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------