Subject: Importing data from C/C++ to IDL when type is only known at runtime
Posted by kathryn.ksm on Tue, 25 Apr 2006 20:29:20 GMT
View Forum Message <> Reply to Message

Hi Folks,

I'm trying to write a DLM that uses existing C/C++ libaries to read data in from files that have a custom (and fairly complex) format. This is my first attempt at doing anything serious with IDL and I'm having a hard time figuring out what's possible and what isn't. So far most of what I've tried doesn't seem to be! So I could use some help.

My existing C code reads selected portions of the data files into a set of arrays, where the data type is specified at run-time by the user. So, on the C side of things, I have an array of void pointers that point to dynamically allocated arrays with various types and sizes depending on what the user asked for.

What I need to do is get this data into variables of appropriate types in an interactive session in IDL. Ideally, what I would like to do is to define and fill a set of nested structures in the DLM code at run-time, and return a single structure variable to the IDL session. This doesn't seem to be possible though. I can see how I could define such a structure at run-time using an array of IDL_STRUCT_TAG_DEFs and IDL_MakeStruct, but I don't see any way to fill that structure without creating an analogous structure in C in advance (which doesn't seem doable, but maybe I'm wrong about that).

A second approach I've considered is to return my array of arrays to IDL, and to use a wrapper .pro routine to sort that into a structure. Is there any way to return some kind of analogue to a void** from a DLM? Or, in other words, is there any way that I can return an array of arrays that have different types?

A third option that I've wondered about is if I could use a two-step process: read in all of the information that I need to call something like create_struct from the IDL side, and then pass the resulting structure to my C function to actually read the data. Then I would need to know how to get pointers to the structure's data elements from within the DLM, which I'm not sure about either.

Sorry for the long-winded question, but I figure there are experts out there who can easily see where I'm missing something. I'd be so grateful for any help figuring out how to do this!

Many thanks,

Kathryn M.