Subject: Re: dlm returning ptr array and string array
Posted by R.Bauer on Tue, 04 Mar 2003 08:20:07 GMT
View Forum Message <> Reply to Message

Randall Skelton wrote:
> On Mon, 3 Mar 2003, Reimar Bauer wrote:
>
>
>>>> Now my questions:
>>>> Is it possible to return a pointer vector by a dlm?
>>>
>>> I am not entirely sure I understand what you are asking here. You cannot
>>> create an IDL pointer in C, so if that is what you are asking you'll need
>>> to rethink things. You can, create and return IDL variables and arrays of
>>> any type (other than pointers). Likewise with structures but you cannot
>>> directly interface these as objects (unless, of course, Ronn has some new
>>> tricks to show us)
>>
>> I found an article from Nigel Wade by searching google so I think it is
>> possible or ?
>> .http://groups.google.de/groups?hl=de&lr=&ie=UTF-8&a
mp;threadm=aoehm9%247pf6%241%40rook.le.ac.uk&rnum=9&
prev=/groups%3Fq%3Ddlm%2B%252Bpointer%26hl%3Dde%26lr%3D%26ie
%3DUTF-8%26selm%3Daoehm9%25247pf6%25241%2540rook.le.ac.uk%26 rnum%3D9
>
>
> Yes, if all you want to do is return a C pointer, then what Nigel suggests
> is absolutely fine. In my dlms for postgresql rather than return the
> pointer directly cast as a long, I build a table of simple connection or
> result indicies that keeps track of the C pointers. In this way, I can
> return a simple index number rather that I can subsequently use to
> 'lookup' the pointers in C (much like the IDL/Fortran LUNs work). I did
> this after a few people using my code commented that the bizarre unsigned
> long integers that represented my C pointers must signify an error has
> occurred. What you cannot do is directly create an IDL pointer in C (i.e.
> IDL> a = ptr_new(...)).
>
> If you are going to write your own data format, you may want to use the
> XDF or CDF network formats as these are cross-platform for posix machines.
>

Dear Randall,

thanks again.

I am not thinking about writing one more own format. We all have enough
done already. Since 1996 we are using netCDF and I don't like changing this.
We are using for attributes which might be necessary and parameter names

a definition in our institutes. This means wrong typed attributes are ignored by reading and writing. But the sources must be only once written and doesn't need cases for different written attributes. 'pi_organisation','pio' of everyone.

So we have designed in 1998 a data definition structure for exchanging data between netCDF files and IDL. This structure is able to load every kind of data from every kind of data format. We have a lot of experience of this. In the last years this structure got's more and more important for us because several people wrote routines for example to make a time synchronisation, read/write different dataformats, different plot routines, statistical routines etc.

Here is a routine to create a template of the data structure I am speaking from

wget
 http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl _html/dbase/download/write_icgspro.sav

or

wget
 http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl
_html/dbase/download/write_icgspro.tar.gz

write_icgspro,'test.pro',/small,short=['time','O3']

So now back to dlm. I will try to find a way to get independent from the used netCDF lib by rsinc for our reading routine. Later netCDF versions for example did not need to copy first the file if you add some new data. And if neccesary I can use a patch.

I have again a question about the pointer I got back from the dlm. Can I use it in idl, print,*ptr or is it only useable from an other c call.

Reimar

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
 ---------- -------

a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html
========== =======