Subject: dlm returning ptr array and string array
Posted by R.Bauer on Mon, 03 Mar 2003 10:26:29 GMT
View Forum Message <> Reply to Message

Hi,

I started a bit in dlm programming. Because of the latest bug in implementing the netCDF library I have to think in this case about getting more independent from rsi.

If they are able to fix it sooner as I am able to program my own netCDF dlm routines then I will save this idea for further usasge.

As you know many of our data access and analysing routines are based on a structure (icg data structure)
This is a structure in many struct levels which are not named. The reason is only tags which are really necessary should be every time included. If someone likes to give more information to his data this should be included into the structure too (e.g. stdev or quality flags).

For example if someones data has a unit, a long_name the structure looks for example:

```
struct.time.units      =  'seconds since 2000-01-01 00:00:00 UTC'
struct.time.long_name  =  'time'
struct.time.param      =  [ .....]
struct.time.short_name =  'time'

struct.o3.units        =  'ppm'
struct.o3.long_name    =  'mixing ratio'
struct.o3.param        =  [ .....]
struct.o3.short_name   =  'O3'
```

Some other tags describes something about the dataset itselfs this is stored in a main level structure for example pi or experiment information,

```
struct.!global.pi.name           = 'M.Mustermann'
struct.!global.dataset.experiment = 'myex'
struct.!global.platform,type      = 'AIRCRAFT'
```

and possible others.

(There are more than 100 other definitions of tags sometimes useful)

I am quite sure it is not easy to get a structure like this returned from a dlm. But I think an other way should be possible.

As some of you know we have some routines which are able to add or change
tags and parameters on each level of a given structure.
This is possible because we got the idea to build from a structure a vector
of strings and a vector of pointers to the data.
For deleting a tag this means for example to remove an index from both
vectors and to free this pointer. Then both vectors are used as input to a
function returning the structure.

Now my questions:
Is it possible to return a pointer vector by a dlm?
Could they freed by idl?
Is this mostly stable?
Does it give memory leakage?
Are there memory limitiations generally using dlms?

regards

Reimar
--
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
http://www.fz-juelich.de/icg/icg-i/
 == ======
a IDL library at ForschungsZentrum Juelich
 http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html

---

Subject: Re: dlm returning ptr array and string array
Posted by Randall Skelton on Mon, 03 Mar 2003 20:00:00 GMT
View Forum Message <> Reply to Message

I'll bite.

> I started a bit in dlm programming. Because of the latest bug in
> implementing the netCDF library I have to think in this case about getting
> more independent from rsi.

Re-implementing the interface to a file format such as netCDF would
probably be a fair amount of work.

> 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).

You can, however, allocate some very complex hash, linked-list, or other
structure that IDL has no knowledge about in C and pass a simple
string/index array back to IDL. You could even go as far as simply
keeping a hash table of indicies, strings, pointers and/or file offsets
that point to your data. Then when you want a specific data block, you
would use the name/index/??? and a separate c routine to return the block
as a structure or array. This is what diploma/co-op students are for ;)

> Could they freed by idl?

Again, there is no way to create or act on an IDL pointer in C. You can
create an return your large structure in C and then index it in IDL but
that isn't particularly clean or easy.

> Is this mostly stable?

Passing ordinary variables, arrays and structures is defined by the stable
DLM API. So, yes.

> Does it give memory leakage?

Depends how good of a C programmer you are ;)

> Are there memory limitiations generally using dlms?

Not really. Because DLMs are raw C code that are linked to IDL as shared
libraries, you basically have access to as much memory as you have
available. Once you allocate memory on the C side, you use various
components of the API to pass pointers to this memory so IDL knows about
it.

Subject: Re: dlm returning ptr array and string array
Posted by R.Bauer on Mon, 03 Mar 2003 20:19:30 GMT

View Forum Message <> Reply to Message

Randall Skelton wrote:

>
> I'll bite.
>
>> I started a bit in dlm programming. Because of the latest bug in
>> implementing the netCDF library I have to think in this case about

>> getting more independent from rsi.
>
> Re-implementing the interface to a file format such as netCDF would
> probably be a fair amount of work.

I won't do this at the moment I am thinking about writing our central
reading and writing routines in c. This routine needs as input only the
filename and a list of short names (and optional start, count and stride).

>
>> 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

Reimar

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

---

Subject: Re: dlm returning ptr array and string array
Posted by Mark Hadfield on Mon, 03 Mar 2003 20:22:37 GMT
View Forum Message <> Reply to Message

"Reimar Bauer" <R.Bauer@fz-juelich.de> wrote in message
news:b408sr$4dlp$1@zam602.zam.kfa-juelich.de...
>
> [In response to a request for details of netCDF bugs in IDL]
>

> On unix systems if you do an inquire of an attribute which isn't
> defined you got in the past versions the type as UNKNOWN
> returned. Then you know it isn't there. Now you get a valid type
> mostly LONG back.
>
> ...
>
> The other problem we will get in the future if unidata did not set
> up a mask character for special signs for the definition of
> parameter names. Because now in idl is 2.4 included. This allows
> you to define ()@:. and some other signs in the parameter
> names. There was a change about this in 3.0 because of preventing
> ncgen to get trouble by some special signs. Russ Rew gaves as
> always a patch to get the old convention back. Actual version is now
> 3.5

Thanks for the details. Another "feature" of IDL's netCDF library is
that it silently adds trailing null characters to strings when they
are stored as netCDF attributes. According to the netCDF gurus this
was a netCDF convention early in netCDF history, but is now
deprecated. When I reported this to RSI their response, as of
September 2001, was that they "have an open feature request for IDL to
include support for the netCDF 3.5 library". I guess the level of
interest shown by customers is not yet sufficient for this feature
request to be acted on.

Sorry I can't help with the DLM, Reimar, but I'd love to use it when
you've written it!

--
Mark Hadfield "Ka puwaha te tai nei, Hoea tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: dlm returning ptr array and string array
Posted by Randall Skelton on Mon, 03 Mar 2003 22:34:49 GMT
View Forum Message <> Reply to Message

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.

Cheers,
Randall

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
 === =======

Subject: Re: dlm returning ptr array and string array
Posted by Nigel Wade on Tue, 04 Mar 2003 11:03:55 GMT
View Forum Message <> Reply to Message

Reimar Bauer wrote:

> 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.
>

Unfortunately, it's only useable within code which can utilize raw memory
addresses; it's a C pointer, which is in essence a memory address.

Can you use the same code you currently use to build the vector of IDL
pointers from the structures returned by a DLM?

--
Nigel Wade, System Administrator, Space Plasma Physics Group,
University of Leicester, Leicester, LE1 7RH, UK
E-mail : nmw@ion.le.ac.uk
Phone : +44 (0)116 2523548, Fax : +44 (0)116 2523555