
Subject: read a C written binary file with IDL
Posted by [Thibault Garel](#) on Fri, 10 Feb 2012 03:44:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I am having a problem with reading a C written binary file with IDL.
It may come from differences of type definitions between C and IDL but
I could not really figure out from Google...

In C, it writes a structure containing the following variable types:

```
struct MyStruct
{
int a;
long long b;
int c;
float d;
};
```

Then, in IDL, I read this with:

```
MyStruct = {$
  a      : 0L, $
  b      : 0LL, $
  c      : 0L, $
  d      : 0.0 $
}
```

```
openr, 1, filename, /SWAP_IF_BIG_ENDIAN
readu, 1, MyStruct
close, 1
```

but this gives me wrong values.

Did I miss something about the type conversion??

If someone could please clarify this, it would really help!
Thanks !

Subject: Re: read a C written binary file with IDL
Posted by manodeep@gmail.com on Tue, 21 Feb 2012 03:43:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Feb 20, 7:49 pm, bing999 <thibaultga...@gmail.com> wrote:
> Hi,


```
>
>>>> This is because C pads the structure to produce alignments. Under
>>>> 'normal' operations, you would expect MyStruct to be 20 bytes,
>>>> however, if you do a sizeof(struct MyStruct), you will probably see
>>>> that the size is 24. (And you can enable the warning for gcc by using
>>>> the compile time option -Wpadded).
>
>>> The N_TAGS() function has LENGTH and DATA_LENGTH keywords:
>
>>> IDL> print, n_tags(mystruct, /length)
>>>      24
>>> IDL> print, n_tags(mystruct, /data_length)
>>>      20
>
>>> I mention it because one might not think to look at the N_TAGS
>>> function
>>> for this sort of information.
>
>>> --Wayne
```

As Paul suggested, you should check out the assoc function. I can read the data in just fine through IDL:

```
-----
openu, 1, 'data.bin', /SWAP_IF_BIG_ENDIAN
a = assoc(1, {a:0L, b:0LL, c:lonarr(5),d:fltarr(9),e:0L,f:fltarr(24)})
print,a[0]
-----
```

Cheers,
Manodeep

Subject: Re: read a C written binary file with IDL
Posted by [Thibault Garel](#) on Wed, 22 Feb 2012 05:45:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hey,

Actually I cannot use assoc i guess. To be fully complete this time :), the binary file contain a header (some int) + the structure mystruct which is replicated N times. From what I understand of the assoc function, it won't work in my case...
And the problem is that i cannot changer the compiling options of the C code...

cheers

Thibault

```
>
>
>> Hi,
>
>> sorry to reply so late, but I was away recently.
>
>> First, as the issue seems to be more complicated than i first thought,
>> i have to say that the structure i have to read is bigger than what I
>> wrote previously just to make an example.
>
>> It actually contains (in the right order):
>> int x 1
>> long long x 1
>> int x 5
>> float x 9
>> int x 1
>> float x 24
>
>> that is, 7 int, 33 float and 1 long long (written in C).
>
>> When I look at the IDL commands you suggest, i get:
>
>> print,n_tags(mystruct,/length) 176
>
>> print,n_tags(mystruct,/data_length) 172
>
>> So you were right.
>
>> Now, the question is "How to handle that?? " :)
>
>> I inserted an extra int (0L) in the structure of my IDL reading
>> routine in second position of the structure (just before the long
>> long), and it gives results coherent with what one gets with a correct
>> Python routine. However, I am not quite satisfied with this
>> solution...
>
>> Does anyone can think of a better way to proceed?
>
>> Thanks !!
>
>>> Bill Nel wrote:
>>>> On Feb 9, 11:56 pm, Manodeep Sinha <manod...@gmail.com> wrote:
>>>> > Hi,
>
>>>> > This is because C pads the structure to produce alignments. Under
```

>>>> > 'normal' operations, you would expect MyStruct to be 20 bytes,
>>>> > however, if you do a sizeof(struct MyStruct), you will probably see
>>>> > that the size is 24. (And you can enable the warning for gcc by using
>>>> > the compile time option -Wpadded).
>
>>>> The N_TAGS() function has LENGTH and DATA_LENGTH keywords:
>
>>>> IDL> print, n_tags(mystruct, /length)
>>>> 24
>>>> IDL> print, n_tags(mystruct, /data_length)
>>>> 20
>
>>>> I mention it because one might not think to look at the N_TAGS
>>>> function
>>>> for this sort of information.
>
>>>> --Wayne
>
> As Paul suggested, you should check out the assoc function. I can read
> the data in just fine through IDL:
>
> -----
> openu, 1, 'data.bin', /SWAP_IF_BIG_ENDIAN
> a = assoc(1, {a:0L, b:0LL, c:lonarr(5),d:fltarr(9),e:0L,f:fltarr(24)})
> print,a[0]
> -----
>
> Cheers,
> Manodeep

Subject: Re: read a C written binary file with IDL
Posted by [David Fanning](#) on Wed, 22 Feb 2012 13:24:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

bing999 writes:

> Actually I cannot use assoc i guess. To be fully complete this
> time :), the binary file contain a header (some int) + the structure
> mystruct which is replicated N times. From what I understand of the
> assoc function, it won't work in my case...

I think your understanding must be faulty, as this
is *exactly* the kind of file ASSOC works best with. :-)

The "offset" is used to account for the header.

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: read a C written binary file with IDL
Posted by [Thibault Garel](#) on Mon, 12 Mar 2012 02:49:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi all,
thanks for your help. Indeed, the assoc function works fine (without using the /packed keyword).
Cheers
Bing

> bing999writes:
>> Actually I cannot use assoc i guess. To be fully complete this
>> time :), the binary file contain a header (some int) + the structure
>> mystruct which is replicated N times. From what I understand of the
>> assoc function, it won't work in my case...
>
> I think your understanding must be faulty, as this
> is *exactly* the kind of file ASSOC works best with. :-)
>
> The "offset" is used to account for the header.
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:<http://www.idlcoyote.com/>
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")
