
Subject: Error in reading large Fortran unformatted files

Posted by [OM](#) on Wed, 16 Feb 2011 12:00:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello everyone,

I'm new to this group, but I hope I won't look too silly...

I've recently started doing calculations in Fortran that result in files containing an n^3 real (single precision) array. As long as n was up to 512, everything worked fine, and I could read the result file with IDL just fine. As long as I switched to $n=1024$, though... (array sizes must be powers of 2, for the FFT that is yet to come). I can open the file, and I can assign an array of the proper size, but as soon as I try to read the file into the array, I get the error:
% READU: Corrupted f77 unformatted file detected. Unit: 2

I checked, and according to here (http://www.physics.nyu.edu/grierlab/idl_html_help/files10.html) the size of the file should be within limits (it's too big for 32 bit systems, but I made sure that I'm running a 64 bit version of IDL on a 64 bit machine). It's not a question of endianness, since I'm running the same Fortran code on the same dataset, and the only thing that changes is the size of the grid. Just to be sure of that point, I also made sure I can read the result file correctly with Fortran and tried opening the file with the /SWAP_ENDIAN and /SWAP_IF_LITTLE_ENDIAN keywords (not at the same time, of course), and I still get the same error. I'm out of ideas by this point... I'll really appreciate any kind of help.

Thanks,
Ofer.

Subject: Re: Error in reading large Fortran unformatted files

Posted by [OM](#) on Thu, 17 Feb 2011 08:24:46 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Feb 16, 5:44 pm, "Kenneth P. Bowman" <k-bow...@null.edu> wrote:

> In article

> <69299e37-2172-46f3-ade9-3a04fa211...@s11g2000yqc.googlegroup s.com >,
>

> OM <metu...@gmail.com> wrote:

>> There's really not much to it...

>> I do it all from the command line:

>> f='name_of_file'

>> d=FLTARR(1024,1024,1024)

>> OPENR, 1, f, /F77_UNFORMATTED

>> READU, 1, d

>
>> At that point I get the error. Again, for 512 this worked fine.
>
> I have always had better luck not setting the /F77_UNFORMATTED
> flag and reading the length words myself.
>
> That is,
>
> nb1 = 0
> nb2 = 0
> OPENR, 1, f
> READU, 1, nb1, f, nb2

Well, I get:

nb1=9

nb2=21461

I have no idea what any of this means...

> In any case, I suspect that you have a 64-bit filesystem, but that
> Fortran only writes 32-bit record lengths. In this case, your variable
> is 4 GB, which will not fit into 32 bits.

From what I understand, that's not the case:

```
IDL> PRINT, !VERSION.FILE_OFFSET_BITS  
      64
```

> The file may be OK, and you can just ignore nb1 and nb2, since
> you know how big the record is in the file.
> Have a look at nb1 and nb2 and see what you find.
>
> Ken Bowman

Thanks for taking an interest!
Ofer.

Subject: Re: Error in reading large Fortran unformatted files
Posted by [Foldy Lajos](#) on Thu, 17 Feb 2011 09:20:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thu, 17 Feb 2011, OM wrote:

```
>> nb1 = 0  
>> nb2 = 0  
>> OPENR, 1, f  
>> READU, 1, nb1, f, nb2  
>
```

> Well, I get:
> nb1=9
> nb2=21461
> I have no idea what any of this means...
>

I think 32 bit integers should be used here, probably unsigned. Try with

```
nb1=0ul  
nb2=0ul  
OPENR, 1, f  
READU, 1, nb1, f, nb2
```

regards,
Lajos

Subject: Re: Error in reading large Fortran unformatted files
Posted by [penteado](#) on Thu, 17 Feb 2011 14:05:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Feb 17, 6:24 am, OM <metu...@gmail.com> wrote:
>> In any case, I suspect that you have a 64-bit filesystem, but that
>> Fortran only writes 32-bit record lengths. In this case, your variable
>> is 4 GB, which will not fit into 32 bits.
>
> From what I understand, that's not the case:
>
> IDL> PRINT, !VERSION.FILE_OFFSET_BITS
> 64

That is just the size of IDL's file offsets. It has no relation to the contents of any particular file.

Subject: Re: Error in reading large Fortran unformatted files
Posted by [OM](#) on Thu, 17 Feb 2011 14:15:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Feb 17, 4:05 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
> On Feb 17, 6:24 am, OM <metu...@gmail.com> wrote:
>
>>> In any case, I suspect that you have a 64-bit filesystem, but that
>>> Fortran only writes 32-bit record lengths. In this case, your variable
>>> is 4 GB, which will not fit into 32 bits.
>
>> From what I understand, that's not the case:

>
>> IDL> PRINT, !VERSION.FILE_OFFSET_BITS
>> 64
>
> That is just the size of IDL's file offsets. It has no relation to the
> contents of any particular file.

Oh, I'm sorry, I misread your message. How can I check this?

Ofer.

Subject: Re: Error in reading large Fortran unformatted files
Posted by [OM](#) on Thu, 17 Feb 2011 14:16:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Feb 17, 11:20 am, FÖLDY Lajos <fo...@rmki.kfki.hu> wrote:

> On Thu, 17 Feb 2011, OM wrote:
>>> nb1 = 0
>>> nb2 = 0
>>> OPENR, 1, f
>>> READU, 1, nb1, f, nb2
>
>> Well, I get:
>> nb1=9
>> nb2=21461
>> I have no idea what any of this means...
>
> I think 32 bit integers should be used here, probably unsigned. Try with
>
> nb1=0ul
> nb2=0ul
> OPENR, 1, f
> READU, 1, nb1, f, nb2
>
> regards,
> Lajos

The output is now:
nb1=2147483657
nb2=995288272

I still have no idea what this means.

Ofer.

Subject: Re: Error in reading large Fortran unformatted files
Posted by [Kenneth P. Bowman](#) on Thu, 17 Feb 2011 15:02:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article

<45a7d29c-1223-4e0e-8390-5a549f91cd02@s11g2000yqh.googlegroups.com>,

OM <metukio@gmail.com> wrote:

> The output is now:
> nb1=2147483657
> nb2=995288272
>
> I still have no idea what this means.

nb1 is the largest possible positive 32-bit signed integer

```
IDL> print, 2L^31 - 1  
2147483647
```

The fact that nb2 does not match nb1 indicates a problem.

The problem is probably that the version of Fortran that wrote your file only allows 2 GB records. That is, the length word at the beginning and end of each record is only 4 bytes.

A 64-bit fortran may or may not use 8-byte length words. You may be able to set this as a Fortran option.

You should be able to find out in your Fortran documentation, or you can find it experimentally by running a Fortran program that writes a 2 GB record. I suggest that you write 4-byte integer zeroes. Then try reading the file in IDL.

You can try this, which will read the first 4 bytes as the length word

```
nb1=0ul  
nb2=0ul  
OPENR, 1, f  
READU, 1, nb1, d, nb2
```

or this, which will read the first 8 bytes as the length word

```
nb1 = 0ULL  
nb2 = 0ULL  
OPENR, 1, f  
READU, 1, nb1, d, nb2
```

Then repeat the experiment with a 4GB record.

Ken Bowman

Subject: Re: Error in reading large Fortran unformatted files

Posted by [Nigel Wade](#) on Thu, 17 Feb 2011 16:16:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

On 17/02/11 15:02, Kenneth P. Bowman wrote:

> In article

> <45a7d29c-1223-4e0e-8390-5a549f91cd02@s11g2000yqh.googlegroups.com>,

> OM <metukio@gmail.com> wrote:

>

>> The output is now:

>> nb1=2147483657

>> nb2=995288272

>>

>> I still have no idea what this means.

>

> nb1 is the largest possible positive 32-bit signed integer

>

> IDL> print, 2L^31 - 1

> 2147483647

>

The value quoted is 2147483657, which is 10 more than that. Assuming OM cut'n'pasted the output, so it's not just a typo, it's a number which has no immediate significance that I can think of.

I do, however, agree that the problem is almost certainly due to trying to write 4GB of data as a single FORTRAN unformatted record. I doubt that when the FORTRAN unformatted format was devised it was never envisioned that someone would try to output that much data in a single write statement. The record length is a 32bit quantity. I don't see that that can be altered based on platform, the format must be the same for 32bit and 64bit platforms, and applications. I think the max. you can possibly write in a single record is 2GB-1. To write 4GB will require at least 3 records.

--

Nigel Wade

Subject: Re: Error in reading large Fortran unformatted files

Posted by [OM](#) on Thu, 17 Feb 2011 18:15:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Feb 17, 6:16 pm, Nigel Wade <nmw-n...@ion.le.ac.uk> wrote:

> On 17/02/11 15:02, Kenneth P. Bowman wrote:
>
>> In article
>> <45a7d29c-1223-4e0e-8390-5a549f91c...@s11g2000yqh.googlegroup s.com >,
>> OM <metu...@gmail.com> wrote:
>
>>> The output is now:
>>> nb1=2147483657
>>> nb2=995288272
>
>>> I still have no idea what this means.
>
>> nb1 is the largest possible positive 32-bit signed integer
>
>> IDL> print, 2L^31 - 1
>> 2147483647
>
> The value quoted is 2147483657, which is 10 more than that. Assuming OM
> cut'n'pasted the output, so it's not just a typo, it's a number which
> has no immediate significance that I can think of.
>
> I do, however, agree that the problem is almost certainly due to trying
> to write 4GB of data as a single FORTRAN unformatted record. I doubt
> that when the FORTRAN unformatted format was devised it was never
> envisioned that someone would try to output that much data in a single
> write statement. The record length is a 32bit quantity. I don't see that
> that can be altered based on platform, the format must be the same for
> 32bit and 64bit platforms, and applications. I think the max. you can
> possibly write in a single record is 2GB-1. To write 4GB will require at
> least 3 records.
>
> --
> Nigel Wade

Well, here's the pickle - I'm getting no errors in writing the file,
and with slight modifications I can read the data in Fortran and it
seems to be valid.

Ofer.

Subject: Re: Error in reading large Fortran unformatted files
Posted by [Kenneth P. Bowman](#) on Thu, 17 Feb 2011 18:22:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article
<41f34b18-af51-49e4-867b-69eaba5f8454@f18g2000yqd.googlegroups.com>,
OM <metukio@gmail.com> wrote:

> Well, here's the pickle - I'm getting no errors in writing the file,
> and with slight modifications I can read the data in Fortran and it
> seems to be valid.

Are there endian issues?

In any case, the simplest solution is probably to break your
Fortran write into four 1 GB records (to be on the safe side).

Then read it in the same way in IDL.

Ken Bowman

Subject: Re: Error in reading large Fortran unformatted files

Posted by [Nigel Wade](#) on Fri, 18 Feb 2011 09:45:07 GMT

[View Forum Message](#) <> [Reply to Message](#)

On 17/02/11 18:15, OM wrote:

> On Feb 17, 6:16 pm, Nigel Wade <nmw-n...@ion.le.ac.uk> wrote:

>> On 17/02/11 15:02, Kenneth P. Bowman wrote:

>>

>>> In article

>>> <45a7d29c-1223-4e0e-8390-5a549f91c...@s11g2000yqh.googlegroup s.com >,

>>> OM <metu...@gmail.com> wrote:

>>

>>>> The output is now:

>>>> nb1=2147483657

>>>> nb2=995288272

>>

>>>> I still have no idea what this means.

>>

>>> nb1 is the largest possible positive 32-bit signed integer

>>

>>> IDL> print, 2L^31 - 1

>>> 2147483647

>>

>> The value quoted is 2147483657, which is 10 more than that. Assuming OM

>> cut'n'pasted the output, so it's not just a typo, it's a number which

>> has no immediate significance that I can think of.

>>

>> I do, however, agree that the problem is almost certainly due to trying

>> to write 4GB of data as a single FORTRAN unformatted record. I doubt

>> that when the FORTRAN unformatted format was devised it was never

>> envisioned that someone would try to output that much data in a single

>> write statement. The record length is a 32bit quantity. I don't see that

>> that can be altered based on platform, the format must be the same for

>> 32bit and 64bit platforms, and applications. I think the max. you can
>> possibly write in a single record is 2GB-1. To write 4GB will require at
>> least 3 records.
>>
>> --
>> Nigel Wade
>
> Well, here's the pickle - I'm getting no errors in writing the file,
> and with slight modifications I can read the data in Fortran and it
> seems to be valid.
>
> Ofer.

Well, maybe the underlying point is that the actual contents of FORTRAN unformatted records are actually undefined, at least they never were defined up to F77 which is last version of FORTRAN I used. They are an implementation issue, each compiler on each platform was free to define the format to be what it chose. Unformatted data was never meant to be portable, it was merely an efficient means of saving data from one FORTRAN program which could be read back by another FORTRAN program compiled by the same compiler on the same platform.

An ad hoc "standard" developed, which was that the first 4 and last 4 bytes contained the record length. This allowed some consistency check and limited portability (endian issues and other things). Maybe the FORTRAN compiler you are using has a different way of writing unformatted data records which extend beyond the limit of the previous 2GB "standard". Obviously it can read back data which it wrote, but IDL cannot.

--
Nigel Wade

Subject: Re: Error in reading large Fortran unformatted files
Posted by [OM](#) on Fri, 18 Feb 2011 15:13:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Feb 17, 8:22 pm, "Kenneth P. Bowman" <k-bow...@null.edu> wrote:
> In article
> <41f34b18-af51-49e4-867b-69eaba5f8...@f18g2000yqd.googlegroup s.com >,
>
> OM <metu...@gmail.com> wrote:
>> Well, here's the pickle - I'm getting no errors in writing the file,
>> and with slight modifications I can read the data in Fortran and it
>> seems to be valid.
>
> Are there endian issues?

>
> In any case, the simplest solution is probably to break your
> Fortran write into four 1 GB records (to be on the safe side).
>
> Then read it in the same way in IDL.
>
> Ken Bowman

Well, the data I'm saving is one 3D array. The only way I can think of breaking the file is actually breaking the array, and I'd really rather do that only as a last resort. Do you have any other idea regarding this?

Ofer.

Subject: Re: Error in reading large Fortran unformatted files

Posted by [OM](#) on Fri, 18 Feb 2011 15:14:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Feb 18, 11:45 am, Nigel Wade <nmw-n...@ion.le.ac.uk> wrote:

> On 17/02/11 18:15, OM wrote:

>

>

>

>> On Feb 17, 6:16 pm, Nigel Wade <nmw-n...@ion.le.ac.uk> wrote:

>>> On 17/02/11 15:02, Kenneth P. Bowman wrote:

>

>>>> In article

>>>> <45a7d29c-1223-4e0e-8390-5a549f91c...@s11g2000yqh.googlegroup s.com > ,

>>>> OM <metu...@gmail.com> wrote:

>

>>>> > The output is now:

>>>> > nb1=2147483657

>>>> > nb2=995288272

>

>>>> > I still have no idea what this means.

>

>>>> nb1 is the largest possible positive 32-bit signed integer

>

>>>> IDL> print, 2L^31 - 1

>>>> 2147483647

>

>>> The value quoted is 2147483657, which is 10 more than that. Assuming OM

>>> cut'n'pasted the output, so it's not just a typo, it's a number which

>>> has no immediate significance that I can think of.

>

>>> I do, however, agree that the problem is almost certainly due to trying

>>> to write 4GB of data as a single FORTRAN unformatted record. I doubt
>>> that when the FORTRAN unformatted format was devised it was never
>>> envisioned that someone would try to output that much data in a single
>>> write statement. The record length is a 32bit quantity. I don't see that
>>> that can be altered based on platform, the format must be the same for
>>> 32bit and 64bit platforms, and applications. I think the max. you can
>>> possibly write in a single record is 2GB-1. To write 4GB will require at
>>> least 3 records.
>
>>> --
>>> Nigel Wade
>
>> Well, here's the pickle - I'm getting no errors in writing the file,
>> and with slight modifications I can read the data in Fortran and it
>> seems to be valid.
>
>> Ofer.
>
> Well, maybe the underlying point is that the actual contents of FORTRAN
> unformatted records are actually undefined, at least they never were
> defined up to F77 which is last version of FORTRAN I used. They are an
> implementation issue, each compiler on each platform was free to define
> the format to be what it chose. Unformatted data was never meant to be
> portable, it was merely an efficient means of saving data from one
> FORTRAN program which could be read back by another FORTRAN program
> compiled by the same compiler on the same platform.
>
> An ad hoc "standard" developed, which was that the first 4 and last 4
> bytes contained the record length. This allowed some consistency check
> and limited portability (endian issues and other things). Maybe the
> FORTRAN compiler you are using has a different way of writing
> unformatted data records which extend beyond the limit of the previous
> 2GB "standard". Obviously it can read back data which it wrote, but IDL
> cannot.
>
> --
> Nigel Wad

So I take it the only viable solution you can think of is as suggested
by Ken - to break down the file into manageable bits?

Ofer.

Subject: Re: Error in reading large Fortran unformatted files
Posted by [penteado](#) on Fri, 18 Feb 2011 15:35:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Feb 18, 1:14 pm, OM <metu...@gmail.com> wrote:
> So I take it the only viable solution you can think of is as suggested
> by Ken - to break down the file into manageable bits?

That is what it seems to me, unless the compiler has some options that can be changed regarding how it writes unformatted data. It seems the file size is not a problem, only the record length is, so just separate writes to the same file would do it.

Subject: Re: Error in reading large Fortran unformatted files
Posted by [penteado](#) on Fri, 18 Feb 2011 15:37:01 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Feb 18, 1:35 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
> On Feb 18, 1:14 pm, OM <metu...@gmail.com> wrote:
>
>> So I take it the only viable solution you can think of is as suggested
>> by Ken - to break down the file into manageable bits?
>
> That is what it seems to me, unless the compiler has some options that
> can be changed regarding how it writes unformatted data. It seems the
> file size is not a problem, only the record length is, so just
> separate writes to the same file would do it.

I should say "regarding how the program will write the data", as it is not the compiler that does the writing.

Subject: Re: Error in reading large Fortran unformatted files
Posted by [Nigel Wade](#) on Fri, 18 Feb 2011 17:15:35 GMT

[View Forum Message](#) <> [Reply to Message](#)

On 18/02/11 15:14, OM wrote:

>
> So I take it the only viable solution you can think of is as suggested
> by Ken - to break down the file into manageable bits?
>

I don't know of any other. I think it would be easier to write, and to read back, as 1024 records. Programming the loop parameters would be simpler.

I've had a play with a little FORTRAN program, compiled using gfortran, which writes files >2GB. It seems the first 32bit word is always 2147483657, regardless of the length of the record actually output.

Maybe this is some special flag to the FORTRAN I/O library to indicate a large record, I don't know. I'm not about to load a file that size into a binary editor to look at it.

Also, the file is actually 8bytes longer than it should be for a single record. So it may be that FORTRAN is actually splitting the record into two, and it knows this because of that special record length indicator. Presumably IDL doesn't understand this new "feature" of the GNU FORTRAN compiler, and fails to read the file.

--

Nigel Wade

Subject: Re: Error in reading large Fortran unformatted files
Posted by [OM](#) on Sun, 20 Feb 2011 15:10:27 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Feb 18, 7:15 pm, Nigel Wade <nmw-n...@ion.le.ac.uk> wrote:

> On 18/02/11 15:14, OM wrote:

>

>

>

>> So I take it the only viable solution you can think of is as suggested

>> by Ken - to break down the file into manageable bits?

>

> I don't know of any other. I think it would be easier to write, and to
> read back, as 1024 records. Programming the loop parameters would be
> simpler.

>

> I've had a play with a little FORTRAN program, compiled using gfortran,
> which writes files >2GB. It seems the first 32bit word is always
> 2147483657, regardless of the length of the record actually output.

> Maybe this is some special flag to the FORTRAN I/O library to indicate a
> large record, I don't know. I'm not about to load a file that size into
> a binary editor to look at it.

>

> Also, the file is actually 8bytes longer than it should be for a single
> record. So it may be that FORTRAN is actually splitting the record into
> two, and it knows this because of that special record length indicator.
> Presumably IDL doesn't understand this new "feature" of the GNU FORTRAN
> compiler, and fails to read the file.

>

> --

> Nigel Wade

Well, thanks everybody. I was hoping there's some way around this, but I guess I was wrong... :/

Ofer.

Subject: Re: Error in reading large Fortran unformatted files
Posted by [Kenneth P. Bowman](#) on Mon, 21 Feb 2011 17:15:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article
<0e586501-f675-4677-bb14-4100b379d730@k30g2000yqb.googlegroups.com>,
OM <metukio@gmail.com> wrote:

> Well, thanks everybody. I was hoping there's some way around this, but
> I guess I was wrong... :/

I think I remember (but it has been a very long time) that when
we had Dec Alphas running 64-bit Unix, it would write 8-byte
length words at the beginning and end of Fortran records. This
allowed records greater than 2 GB.

So, if it is possible, it will have to be something you do on
the Fortran side.

In fact, the IDL documentation mentions this possibility

http://idlastro.gsfc.nasa.gov/idl_html_help/Reading_and_Writing_FORTTRAN_Data.html

Ken Bowman
