Subject: Re: Fortran unformatted data: Big or little endian
Posted by David Fanning on Wed, 30 Apr 2003 12:55:44 GMT
View Forum Message <> Reply to Message

Michael Schroeter (schroeter@muk.uni-hannover.de) writes:

> I have a problem reading Fortran unformatted data in IDL. I have a file
> containing data of well known size (e.g., an array of 100 x 100 single
> precision floating point values). Unfortunately, I don't know something
> about the machine (big or little endian machine) on which the file has
> been produced. Is there a way to get this information on the fly by IDL
> in order to open the file automatically in the correctly (using the
> /SWAP_ENDIAN option or not). The way I used so far was trial and error.
> But since I have many of those files I'm searching for a solution
> without recompiling my source code.
>
> Do you have any ideas?

Here is an article with some ideas:

http://www.dfanning.com/tips/endian_machines.html

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Fortran unformatted data: Big or little endian
Posted by Michael Schroeter on Wed, 30 Apr 2003 13:30:13 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Michael Schroeter (schroeter@muk.uni-hannover.de) writes:
>
>
>> I have a problem reading Fortran unformatted data in IDL. I have a file
>> containing data of well known size (e.g., an array of 100 x 100 single
>> precision floating point values). Unfortunately, I don't know something
>> about the machine (big or little endian machine) on which the file has
>> been produced. Is there a way to get this information on the fly by IDL
>> in order to open the file automatically in the correctly (using the

>> /SWAP_ENDIAN option or not). The way I used so far was trial and error.
>> But since I have many of those files I'm searching for a solution
>> without recompiling my source code.
>>
>> Do you have any ideas?
>
>
> Here is an article with some ideas:
>
>   http://www.dfanning.com/tips/endian_machines.html
>
> Cheers,
>
> David
>

Hi David,

the problem is that I don't know the "endian nature" of the machine that
created the data.

In the meanwhile I tried using error catching, like shown in this
code-snippet:

```
    array = FLTARR(100,100)
    OPENR, lun, data_file, /F77_UNFORMATTED, /GET_LUN

    CATCH, error_status
    IF error_status NE 0 THEN BEGIN
       PRINT, 'Error index: ', error_status
       PRINT, 'Error message: ', !ERROR_STATE.MSG
       FREE_LUN, lun
       OPENR, lun, data_file, /SWAP_ENDIAN, /F77_UNFORMATTED, /GET_LUN
       READU, lun, array
       CATCH, /CANCEL
    ENDIF ELSE BEIN
 READU, array
    ENDELSE
```

and it works (up to now ;-)).

Anyway, thanks and best regards

Michael

Subject: Re: Fortran unformatted data: Big or little endian

Posted by David Fanning on Wed, 30 Apr 2003 13:39:14 GMT
View Forum Message <> Reply to Message

Michael Schroeter (schroeter@muk.uni-hannover.de) writes:

> and it works (up to now ;-)).

I suspect it works *most* of the time. I don't see anything
in there to *cause* the error. :-)

You can certainly read data any way you like. It's not
until you try to make *sense* of the data that you are
likely to run into trouble. In your case, you might try
reading a known data value from the file. It that value
is total nonsense (your were expecting 45 and you get
-20435), then you could "throw" an error, rewind the file,
and read the data again with the SWAP_ENDIAN keyword set.

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Fortran unformatted data: Big or little endian
Posted by Paul Van Delst[1] on Wed, 30 Apr 2003 14:48:43 GMT
View Forum Message <> Reply to Message

Michael Schroeter wrote:
>
> Hi David,
>
> the problem is that I don't know the "endian nature" of the machine that
> created the data.
>
> In the meanwhile I tried using error catching, like shown in this
> code-snippet:
>
> array = FLTARR(100,100)
> OPENR, lun, data_file, /F77_UNFORMATTED, /GET_LUN
>
> CATCH, error_status

```
>       IF error_status NE 0 THEN BEGIN
>          PRINT, 'Error index: ', error_status
>          PRINT, 'Error message: ', !ERROR_STATE.MSG
>          FREE_LUN, lun
>          OPENR, lun, data_file, /SWAP_ENDIAN, /F77_UNFORMATTED, /GET_LUN
>          READU, lun, array
>          CATCH, /CANCEL
>       ENDIF ELSE BEIN
>          READU, array
>       ENDELSE
>
>  and it works (up to now ;-)).
```

One possible solution:

Open the file as a "regular" binary file (i.e. no /F77_UNFORMATTED keyword)

```
  OPENR, lun, data_file, /GET_LUN
```

Since the file is Fortran unformatted sequential output and you know the record size is always 100x100 single precision (4 byte) floats then the first 4-bytes should be an integer with the value 10000x4. Read this value and check it. If it's not 40000 then it indicates the endian-ness is opposite of the default.

```
  ; -- Check the record size
  RecordSize = 10000L * 4L
  RecordSize_Test = 0L
  READU, lun, RecordSize_Test
  IF ( RecordSize_Test NE RecordSize ) THEN $
    Swap = 1 $
  ELSE $
    Swap = 0

  ; -- Close the file
  FREE_LUN, lun

  ; -- Open the file with the correct keywords
  OPENR, lun, data_file, SWAP_ENDIAN=Swap, /F77_UNFORMATTED, /GET_LUN

  READU, lun, array

 ....proceed with stuff.....
```

And the nice thing is this should work on big- and little-endian machine for either type of file. (Aside: I really don't understand the usefulness of the swap_if_little(big)_endian type of keyword.)

If you have access to the writing routines, one other way is to first write a "magic"

number to the data file - a PARAMETER (i.e. doesn't change) that you can check for. (You may also want to write the dimensions too). Liam Gumley has some handly little utilities for this in IDL (check his website). I've adopted those techniques for the flat binary files I use in my Fortran95 code to identify the endian-ness of those files - nice and simple and works every time.

The best solution (IMO) is to use platform independent data files (e.g. netCDF - my favourite coz it relatively simple). Not always possible of course.

paulv

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC
Ph: (301)763-8000 x7748
Fax:(301)763-8545

Subject: Re: Fortran unformatted data: Big or little endian
Posted by thompson on Wed, 30 Apr 2003 15:05:58 GMT
View Forum Message <> Reply to Message

David Fanning <david@dfanning.com> writes:

> Michael Schroeter (schroeter@muk.uni-hannover.de) writes:

>> and it works (up to now ;-)).

> I suspect it works *most* of the time. I don't see anything
> in there to *cause* the error. :-)

> You can certainly read data any way you like. It's not
> until you try to make *sense* of the data that you are
> likely to run into trouble. In your case, you might try
> reading a known data value from the file. It that value
> is total nonsense (your were expecting 45 and you get
> -20435), then you could "throw" an error, rewind the file,
> and read the data again with the SWAP_ENDIAN keyword set.

My experience has been that if one reads in floating point data with the wrong byte-order, not only is the result garbage, but that one ends up with some extremely large numbers, both positive and negative. For example, I reversed the byte-order of an ordinary array, and ended up with an array with values of 2.70437e+38 and -1.90746e+38. One can use this to augment the procedure you already have, by checking for values beyond a (generously defined) reasonable range. In a sense, this is the same idea as David's, extended to where you may not know what the exact value is that you expect, but you can recognize a ridiculous value when you see one.

Cheers,

Bill Thompson

Subject: Re: Fortran unformatted data: Big or little endian
Posted by Mark Hadfield on Wed, 30 Apr 2003 20:13:12 GMT
View Forum Message <> Reply to Message

"Michael Schroeter" <schroeter@muk.uni-hannover.de> wrote in message
news:b8og1i$6bt$1@newsserver.rrzn.uni-hannover.de...

> I have a problem reading Fortran unformatted data in IDL. I have a
> file containing data of well known size (e.g., an array of 100 x 100
> single precision floating point values). Unfortunately, I don't know
> something about the machine (big or little endian machine) on which
> the file has been produced. Is there a way to get this information
> on the fly by IDL in order to open the file automatically in the
> correctly (using the /SWAP_ENDIAN option or not). The way I used so
> far was trial and error. But since I have many of those files I'm
> searching for a solution without recompiling my source code.

If you have a reasonably recent version of IDL (>= 5.4??) and you know
the byte-order of the machine on which you're *reading* the data then
you can use OPEN's SWAP_IF_BIG_ENDIAN or SWAP_IF_LITTLE_ENDIAN
keywords. Here's the documentation entry for the former:

  SWAP_IF_BIG_ENDIAN

  Setting this keyword is equivalent to setting SWAP_ENDIAN; it only
  takes effect if the current system has big endian byte
  ordering. This keyword does not refer to the byte ordering of the
  input data, but to the computer hardware.

--

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

Subject: Re: Fortran unformatted data: Big or little endian
Posted by David Fanning on Wed, 30 Apr 2003 20:31:05 GMT
View Forum Message <> Reply to Message

Mark Hadfield (m.hadfield@niwa.co.nz) writes:

> If you have a reasonably recent version of IDL (>= 5.4??) and you know
> the byte-order of the machine on which you're *reading* the data then
> you can use OPEN's SWAP_IF_BIG_ENDIAN or SWAP_IF_LITTLE_ENDIAN
> keywords. Here's the documentation entry for the former:
>
> SWAP_IF_BIG_ENDIAN
>
> Setting this keyword is equivalent to setting SWAP_ENDIAN; it only
> takes effect if the current system has big endian byte
> ordering. This keyword does not refer to the byte ordering of the
> input data, but to the computer hardware.

The problem seems to be that the guy doesn't know
the ordering of the INPUT data. Like always, if you
don't know what it is you are reading, you are in
for a bit of bother (to paraphrase Whinnie the Pooh).

Maybe he will get lucky. It happens. :-)

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Fortran unformatted data: Big or little endian
Posted by Mark Hadfield on Wed, 30 Apr 2003 20:52:21 GMT
View Forum Message <> Reply to Message

"David Fanning" <david@dfanning.com> wrote in message
news:MPG.1919e072d7e2c9d9989b79@news.frii.com...
> Mark Hadfield (m.hadfield@niwa.co.nz) writes:
>
>> [deleted]
>
> The problem seems to be that the guy doesn't know
> the ordering of the INPUT data. Like always, if you
> don't know what it is you are reading, you are in
> for a bit of bother (to paraphrase Whinnie the Pooh).

I just spent 5 minutes writing a response explaining why my suggestion was

correct and then suddenly realised why it wasn't.

That's "Winnie", short for Winston, not "Whinnie", for the noise a horse
makes. (It's nice to be right occasionally about *something*.)

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

Subject: Re: Fortran unformatted data: Big or little endian
Posted by David Fanning on Wed, 30 Apr 2003 21:42:47 GMT
View Forum Message <> Reply to Message

Mark Hadfield (m.hadfield@niwa.co.nz) writes:

> That's "Winnie", short for Winston, not "Whinnie", for the noise a horse
> makes. (It's nice to be right occasionally about *something*.)

Oh, maybe that's what my spell checker was complaining
about. Oh, well, join the club (said in my best Eyore
voice).

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Fortran unformatted data: Big or little endian
Posted by Dick Jackson on Fri, 02 May 2003 16:24:38 GMT
View Forum Message <> Reply to Message

"David Fanning" <david@dfanning.com> wrote in message
news:MPG.1919f143d8713cba989b7a@news.frii.com...
> Mark Hadfield (m.hadfield@niwa.co.nz) writes:
>
>> That's "Winnie", short for Winston, not "Whinnie", for the noise a
horse
>> makes. (It's nice to be right occasionally about *something*.)

Spelling's right, just a bit off on the explanation. I'd be remiss if I
didn't kindly point out that the name Winnie comes indirectly from the
name of my home town, Winnipeg. Photos and more at
http://www.just-pooh.com/history.html

> Oh, maybe that's what my spell checker was complaining
> about. Oh, well, join the club (said in my best Eyore
> voice).

I don't suppose I should mention "Eeyore"... ;-)

Back to our regularly scheduled digression.

Cheers
--
-Dick

Dick Jackson / dick@d-jackson.com
D-Jackson Software Consulting / http://www.d-jackson.com
Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392

Subject: Re: Fortran unformatted data: Big or little endian
Posted by btt on Fri, 02 May 2003 16:54:14 GMT
View Forum Message <> Reply to Message

Dick Jackson wrote:
> "David Fanning" <david@dfanning.com> wrote in message
> news:MPG.1919f143d8713cba989b7a@news.frii.com...
>
>> Mark Hadfield (m.hadfield@niwa.co.nz) writes:
>>
>>
>>> That's "Winnie", short for Winston, not "Whinnie", for the noise a
>>
> horse
>
>>> makes. (It's nice to be right occasionally about *something*.)
>>
>
> Spelling's right, just a bit off on the explanation. I'd be remiss if I
> didn't kindly point out that the name Winnie comes indirectly from the
> name of my home town, Winnipeg. Photos and more at
> http://www.just-pooh.com/history.html
>
>
>> Oh, maybe that's what my spell checker was complaining
>> about. Oh, well, join the club (said in my best Eyore

>> voice).
>
>
> I don't suppose I should mention "Eeyore"... ;-)
>
> Back to our regularly scheduled digression.

Oh, silly old bear!

Winnie, as it turns out, comes from 'Edward Bear' not Winston. But that secret is only known to those of use who have read the first chapter ten million times in the last six months at kiddie bed time.

But you know, Pooh and his house could be a good example for how an object might work. Let's see...

Pooh leaves his house one morning to find Piglet. Not finding him, Pooh returns home to discover Piglet on his doorstep. Piglet has come to see Pooh, but no one answers his knock at the door. So Pooh steps up to the door and helps Piglet knock. After a while, Pooh says, "There isn't anyone home."

So Pooh has three methods for interacting with the house object:

```
IDL> Pooh->Knock
IDL> response = Pooh->Listen()
IDL> isHome = Pooh->Decide(response)
```

Pooh isn't allowed to see the contents of other objects' properties (like the house object's property of the number of occupants.) Pooh has to use his own three methods to determine the status of the other and he isn't allowed to cheat (as if he would!) with his insider knowledge that he *knew* nobody was home at his own house.

Geez, maybe it's a wobbly example (like Pooh's spelling) after all. Come on David, chime in anytime now - you've read Pooh a lot I'm sure.

Cheers,
Ben

Subject: Re: Fortran unformatted data: Big or little endian
Posted by David Fanning on Fri, 02 May 2003 23:26:41 GMT
View Forum Message <> Reply to Message

Ben Tupper (btupper@bigelow.org) writes:

> But you know, Pooh and his house could be a good example for how an
> object might work. Let's see...

I always wondered what happened to that flask I
used to keep stashed behind the Complete Works
of Shakespeare on the kid's bookshelf. (Strictly
to get myself fortified for the evening's
story entertainment.) How in the world do you
suppose it made it all the way to Maine from
the soccer club garage sale!?

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Fortran unformatted data: Big or little endian
Posted by Mark Hadfield on Sun, 04 May 2003 22:07:28 GMT
View Forum Message <> Reply to Message

"Dick Jackson" <dick@d-jackson.com> wrote in message
news:aZwsa.101820$ja.3889392@news2.calgary.shaw.ca...
> Spelling's right, just a bit off on the explanation. I'd be remiss if I
> didn't kindly point out that the name Winnie comes indirectly from the
> name of my home town, Winnipeg. Photos and more at
> http://www.just-pooh.com/history.html

Thank you! That's the most interesting thing I've ever learned from this
newsgroup.

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

Subject: Re: Fortran unformatted data: Big or little endian
Posted by David Fanning on Sun, 04 May 2003 23:17:19 GMT
View Forum Message <> Reply to Message

Mark Hadfield (m.hadfield@niwa.co.nz) writes:

> Thank you! That's the most interesting thing I've ever learned from this
> newsgroup.

Now, now. Where's your smiley face? :-)

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Fortran unformatted data: Big or little endian
Posted by Michael Schroeter on Mon, 05 May 2003 09:35:42 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
>> and it works (up to now ;-)).
>
>
> I suspect it works *most* of the time. I don't see anything
> in there to *cause* the error. :-)
>

On my machines (IDL5.6, Red Hat Linux and hpux10.20) "readu" returns an error ("Corrupted f77 unformatted file detected") if I swap to the wrong endian-ness at the opening procedure.

>
> You can certainly read data any way you like. It's not
> until you try to make *sense* of the data that you are
> likely to run into trouble. In your case, you might try
> reading a known data value from the file. It that value
> is total nonsense (your were expecting 45 and you get
> -20435), then you could "throw" an error, rewind the file,
> and read the data again with the SWAP_ENDIAN keyword set.
>

Possibly, checking the data against sense/nonsense or reading a "magic" number as suggested by Paul van Delst will be a better (maybe more

machine-independent?) solution.

Thanks, Michael

--
---------- ----------
Institute of Meteorology und Climatology
University of Hannover/Germany
Herrenhaeuser Str. 2, 30419 Hannover http://www.muk.uni-hannover.de

---

Subject: Re: Fortran unformatted data: Big or little endian
Posted by btt on Mon, 05 May 2003 14:07:17 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Ben Tupper (btupper@bigelow.org) writes:
>
>
>> But you know, Pooh and his house could be a good example for how an
>> object might work. Let's see...
>
>
> I always wondered what happened to that flask I
> used to keep stashed behind the Complete Works
> of Shakespeare on the kid's bookshelf. (Strictly
> to get myself fortified for the evening's
> story entertainment.) How in the world do you
> suppose it made it all the way to Maine from
> the soccer club garage sale!?
>

Is that what the Coors stuff is?

Actually, I think we are all ready for a 'softer' object analogy than
the standard bank account one. The bank account analogy is quite good -
but makes for some pretty dry stuff that no amount of arm-waving will help.

Cheers,
Ben
