Subject: error reading a large number of binary files
Posted by Mark Branson on Fri, 20 Apr 2007 22:21:24 GMT
View Forum Message <> Reply to Message

Is there any particular reason why IDL would struggle reading a really large number of binary files? I'm trying to compute a number of statistics from the ECWMF ERA40 reanalysis 4*daily data, which is a HUGE amount of data. The data was originally in Grib format, which I converted to ieee binary with the wgrib program. Each file contains a month's worth of data, which has 23 pressure levels. In trying to track down the cause of the error I'm getting, I've pared the program down to what you can see below (just compute the long-term average zonal wind). FWIW, I'm running this on a Mac PPC Quad G5 with 8GB of RAM, and it's IDL 6.3.

What happens is that I get an error message like this:

```
% READU: Corrupted f77 unformatted file detected. Unit: 100, File:
daily_4/u196710.bin
% Execution halted at: READUDAILY       19 /Users/mark/Datasets/
era40/readudaily.pro
%                $MAIN$
```

And on repeated trials, the error always occurs in a different file.
And I know that none of the files are actually corrupted.
Interestingly, I wrote a similar program in fortran and using the NAGWare compiler it has similar problems, but it worked successfully using the xlf fortran compiler. However, I'd prefer to have it work in IDL, but I cannot for the life of me figure out what I'm doing wrong.

Any ideas?

Thanks in advance,
Mark Branson
CSU Dept. of Atmospheric Science

```
pro readudaily

; read 4*daily ECMWF ERA-40 reanalysis wind data

x = fltarr(144,73)
ubar = fltarr(144,73,23)

infiles = FILE_SEARCH('daily_4/u*.bin')

icnt = 0L & totcnt = 0L
```

```
mm = 9   ; first file is sept 1957
for i = 0,n_elements(infiles)-1 do begin
  openr, lun, infiles[i], /get_lun, /binary, /f77_unformatted
  print, infiles[i]

  iday = 1 & ihr = 0
  while not EOF(lun) do begin
    for ilev = 0,22 do begin
      readu, lun, x
      ubar[*,*,ilev] = ubar[*,*,ilev] + x[*,*]
    endfor
    icnt = icnt+1
  endwhile

  free_lun, lun

  case 1 of
    (mm eq 2) : begin
          if icnt NE 112 then $
            print, '>>> file = ',infiles[i],' icnt = ',icnt
        end
    (mm eq 1 or mm eq 3 or mm eq 5 or mm eq 7 or mm eq 8 or mm eq 10
or mm eq 12) : begin
          if icnt NE 124 then $
            print, '>>> file = ',infiles[i],' icnt = ',icnt
        end
    (mm eq 2 or mm eq 4 or mm eq 6 or mm eq 9 or mm eq 11) : begin
          if icnt NE 120 then $
            print, '>>> file = ',infiles[i],' icnt = ',icnt
        end
  endcase

  if mm EQ 12 then mm = 1 else mm = mm+1
  totcnt = totcnt + icnt
  icnt = 0L

endfor
print, '>>> totcnt = ',totcnt

; compute mean
for ilev = 0,22 do $
  ubar[*,*,ilev] = ubar[*,*,ilev]/float(totcnt)

openw, lun, 'ubar-annual-idl.data', /get_lun
writeu, lun, ubar
free_lun, lun

end
```

Subject: Re: error reading a large number of binary files
Posted by Paul Van Delst[1] on Mon, 23 Apr 2007 14:24:04 GMT
View Forum Message <> Reply to Message

mark@atmos.colostate.edu wrote:
> Is there any particular reason why IDL would struggle reading a really
> large number of binary files? I'm trying to compute a number of
> statistics from the ECWMF ERA40 reanalysis 4*daily data, which is a
> HUGE amount of data. The data was originally in Grib format, which I
> converted to ieee binary with the wgrib program. Each file contains a
> month's worth of data, which has 23 pressure levels. In trying to
> track down the cause of the error I'm getting, I've pared the program
> down to what you can see below (just compute the long-term average
> zonal wind). FWIW, I'm running this on a Mac PPC Quad G5 with 8GB of
> RAM, and it's IDL 6.3.
>
> What happens is that I get an error message like this:
>
> % READU: Corrupted f77 unformatted file detected. Unit: 100, File:
> daily_4/u196710.bin
> % Execution halted at: READUDAILY 19 /Users/mark/Datasets/
> era40/readudaily.pro
> % $MAIN$
>
> And on repeated trials, the error always occurs in a different file.
> And I know that none of the files are actually corrupted.
> Interestingly, I wrote a similar program in fortran and using the
> NAGWare compiler it has similar problems, but it worked successfully
> using the xlf fortran compiler. However, I'd prefer to have it work
> in IDL, but I cannot for the life of me figure out what I'm doing
> wrong.
>
> Any ideas?

Are the original files little or big endian?

On what machine did you create the "new" files? A little or big endian machine? (Your mac is big endian, right?)

On what machines did you test the fortran program? All your NAG and xlf compilers on you mac?

What happens when you use the /SWAP_ENDIAN keyword in your OPENR?

What does the binary keyword do?
> openr, lun, infiles[i], /get_lun, /binary, /f77_unformatted
I can't find a reference to it in the docs.

cheers,

paulv

--
Paul van Delst Ride lots.
CIMSS @ NOAA/NCEP/EMC Eddy Merckx

Subject: Re: error reading a large number of binary files
Posted by Mark Branson on Mon, 23 Apr 2007 14:59:52 GMT
View Forum Message <> Reply to Message

On Apr 23, 8:24 am, Paul van Delst <Paul.vanDe...@noaa.gov> wrote:
> Are the original files little or big endian?
>
> On what machine did you create the "new" files? A little or big endian machine? (Your mac
> is big endian, right?)
>
> On what machines did you test the fortran program? All your NAG and xlf compilers on you
mac?
>
> What happens when you use the /SWAP_ENDIAN keyword in your OPENR?
>
> What does the binary keyword do?> openr, lun, infiles[i], /get_lun, /binary, /f77_unformatted
>
> I can't find a reference to it in the docs.
>
> cheers,
>
> paulv
>
> --
> Paul van Delst Ride lots.
> CIMSS @ NOAA/NCEP/EMC Eddy Merckx

Hi Paul. I'm 99% sure this is not a big-endian/little-endian or
f77_unformatted issue. I deal with grib files and their conversion
quite often in my work, and my experience is that if you don't have
the switches for them set up correctly, then IDL (or fortran) will
either fail reading the (first) file or you'll get garbage for input.
In this case, I know I can read any of the input files okay because I
can print out a sample from within the program and they are correct
(and I ran the wgrib program on my G5, which for the record is big-
endian). Why it will suddenly fail x number of files into the read
process is where I'm stuck, and it seems to occur at a random spot
with repeated trials.

Before I submitted this post, I searched this newsgroup for related
discussion on this subject, and someone mentioned that the use of the /

binary keyword might be helpful, but like you I don't see it documented in current IDL help for OPENR. Since I've been clutching at straws I decided to insert it, and while it did not produce an error message, it also didn't fix the problem.

Oh, and yes -- I'm running IDL as well as the fortran read program on my Mac G5. I have the NAGWare, Absoft and XLF compilers available. I believe the Absoft compiler also produces the same error, but I should double-check that.

Cheers,
Mark

Subject: Re: error reading a large number of binary files
Posted by Paul Van Delst[1] on Mon, 23 Apr 2007 16:03:15 GMT
View Forum Message <> Reply to Message

mark@atmos.colostate.edu wrote:
> On Apr 23, 8:24 am, Paul van Delst <Paul.vanDe...@noaa.gov> wrote:
>> Are the original files little or big endian?
>>
>> On what machine did you create the "new" files? A little or big endian machine? (Your mac
>> is big endian, right?)
>>
>> On what machines did you test the fortran program? All your NAG and xlf compilers on you
mac?
>>
>> What happens when you use the /SWAP_ENDIAN keyword in your OPENR?
>>
>> What does the binary keyword do?> openr, lun, infiles[i], /get_lun, /binary, /f77_unformatted
>>
>> I can't find a reference to it in the docs.
>>
>> cheers,
>>
>> paulv
>>
>> --
>> Paul van Delst Ride lots.
>> CIMSS @ NOAA/NCEP/EMC Eddy Merckx
>
> Hi Paul. I'm 99% sure this is not a big-endian/little-endian or
> f77_unformatted issue. I deal with grib files and their conversion
> quite often in my work, and my experience is that if you don't have
> the switches for them set up correctly, then IDL (or fortran) will
> either fail reading the (first) file or you'll get garbage for input.
> In this case, I know I can read any of the input files okay because I

> can print out a sample from within the program and they are correct
> (and I ran the wgrib program on my G5, which for the record is big-
> endian). Why it will suddenly fail x number of files into the read
> process is where I'm stuck, and it seems to occur at a random spot
> with repeated trials.

Hi,

Hmm. Is there *no* pattern as to which file (and where in the file) the failure occurs?

Is there a way to rewrite your IDL code that completely separates the grib read from what follows? E.g.

```
  begin file loop...

    call read function....

    do other stuff....

  end file loop
```

Maybe with the file reader completely separated from the body of your code you can test it differently.

What happens if you *only* read the file data, but don't do anything with it? E.g.:

```
pro readudaily
  ; Just because.....
  CATCH, Error_Status
  IF ( Error_Status NE 0 ) THEN BEGIN
    CATCH, /CANCEL
    MESSAGE, !ERROR_STATE.MSG, /CONTINUE
    RETURN
  ENDIF

; read 4*daily ECMWF ERA-40 reanalysis wind data
x = fltarr(144,73)
infiles = FILE_SEARCH('daily_4/u*.bin')
icnt = 0L & totcnt = 0L

for i = 0,n_elements(infiles)-1 do begin

  ; Read the file
  openr, lun, infiles[i], /get_lun, /f77_unformatted
  print, infiles[i]
  while not EOF(lun) do begin
    for ilev = 0,22 do begin
      readu, lun, x
```

```
    endfor
    icnt = icnt+1
  endwhile
  free_lun, lun

  ; Increment counters
  totcnt = totcnt + icnt
  icnt = 0L

endfor
print, '>>> totcnt = ',totcnt

end
```

??

Do you still get failures?

cheers,

paulv

> Before I submitted this post, I searched this newsgroup for related
> discussion on this subject, and someone mentioned that the use of the /
> binary keyword might be helpful, but like you I don't see it
> documented in current IDL help for OPENR. Since I've been clutching
> at straws I decided to insert it, and while it did not produce an
> error message, it also didn't fix the problem.
>
> Oh, and yes -- I'm running IDL as well as the fortran read program on
> my Mac G5. I have the NAGWare, Absoft and XLF compilers available. I
> believe the Absoft compiler also produces the same error, but I should
> double-check that.
>
> Cheers,
> Mark
>

--
Paul van Delst Ride lots.
CIMSS @ NOAA/NCEP/EMC Eddy Merckx

Subject: Re: error reading a large number of binary files
Posted by Mark Branson on Mon, 23 Apr 2007 17:16:23 GMT
View Forum Message <> Reply to Message

On Apr 23, 10:03 am, Paul van Delst <Paul.vanDe...@noaa.gov> wrote:
> Hi,
>
> Hmm. Is there *no* pattern as to which file (and where in the file) the failure occurs?
>
> Is there a way to rewrite your IDL code that completely separates the grib read from what
> follows? E.g.
>
> begin file loop...
>
> call read function....
>
> do other stuff....
>
> end file loop
>
> Maybe with the file reader completely separated from the body of your code you can test it
> differently.
>
> What happens if you *only* read the file data, but don't do anything with it? E.g.:
>
> pro readudaily
> ; Just because.....
> CATCH, Error_Status
> IF (Error_Status NE 0) THEN BEGIN
> CATCH, /CANCEL
> MESSAGE, !ERROR_STATE.MSG, /CONTINUE
> RETURN
> ENDIF
>
> ; read 4*daily ECMWF ERA-40 reanalysis wind data
> x = fltarr(144,73)
> infiles = FILE_SEARCH('daily_4/u*.bin')
> icnt = 0L & totcnt = 0L
>
> for i = 0,n_elements(infiles)-1 do begin
>
> ; Read the file
> openr, lun, infiles[i], /get_lun, /f77_unformatted
> print, infiles[i]
> while not EOF(lun) do begin
> for ilev = 0,22 do begin
> readu, lun, x
> endfor
> icnt = icnt+1

```
>     endwhile
>     free_lun, lun
>
>     ; Increment counters
>     totcnt = totcnt + icnt
>     icnt = 0L
>
>  endfor
>  print, '>>> totcnt = ',totcnt
>
>  end
>
>  ??
>
>  Do you still get failures?
>
>  cheers,
>
>  paulv
>
```

Hi Paul. Thanks for taking an interest in this. On Friday I actually tried something similar to your idea of just reading the files and not doing any computations, and that also failed. And just now I took the program you created above, saved it as readudaily2.pro and tried it three times. Here are the results:

attempt #1:
daily_4/u196112.bin
daily_4/u196201.bin
daily_4/u196202.bin
% READUDAILY2: READU: Corrupted f77 unformatted file detected. Unit: 100, File: daily_4/u196202.bin

attempt #2:
daily_4/u198409.bin
daily_4/u198410.bin
daily_4/u198411.bin
% READUDAILY2: READU: Corrupted f77 unformatted file detected. Unit: 100, File: daily_4/u198411.bin

attempt #3:
daily_4/u199610.bin
daily_4/u199611.bin
% READUDAILY2: READU: Corrupted f77 unformatted file detected. Unit: 100, File: daily_4/u199611.bin

As you can see, it fails on a different file each time. Although

since it seemed to get further along on each successive trial, maybe I should try one or two more times and perhaps I'll get lucky??? :)
Actually, I think this is just random and not a trend.

I wonder if the same error would occur if I generated a large number of binary files of a smaller size (say a 5x5 array of random numbers), because maybe it's not how large the files are but how many of them there are. ?????

Thanks,
Mark

Subject: Re: error reading a large number of binary files
Posted by Paul Van Delst[1] on Mon, 23 Apr 2007 17:47:24 GMT
View Forum Message <> Reply to Message

mark@atmos.colostate.edu wrote:
>
> Hi Paul. Thanks for taking an interest in this. On Friday I actually
> tried something similar to your idea of just reading the files and not
> doing any computations, and that also failed. And just now I took the
> program you created above, saved it as readudaily2.pro and tried it
> three times. Here are the results:

Well, to be honest I didn't think it would work, but it eliminates one of the facile answers.

What a bizarre problem!

The only thing that occurs to me right now (and insert a bunch of hand waving here), is that maybe the IDL procedures are not as sophisticated as the RTLs used by the Fortran codes in reading the files. By that I mean if the record markers in the files do not *exactly* correspond to the length of the data you want to read in, maybe IDL chucks a wobbly and stops. For Fortran code, as long as the input data list is less than or equal to what is in the record, it doesn't matter if you read *less* than what the record contains - the next read will skip the remainder of the last record. If IDL, however, just stops reading the current record and simply starts up again for the next read, then it may treat the remaining data of the previous record as the record marker for the next one and, thus, things go pear shaped. (I told you I was waving my hands about :o)

It may be worthwhile writing a test/check IDL script that opens the files as direct access (i.e. without the /f77_unformatted) and explicitly reads the record markers that bookend each record to check their size and compare that to what you expect. If they're different, then maybe that's the problem. However, it's a not insignificant amount of work to do it.

cheers,

paulv

>
> attempt #1:
> daily_4/u196112.bin
> daily_4/u196201.bin
> daily_4/u196202.bin
> % READUDAILY2: READU: Corrupted f77 unformatted file detected. Unit:
> 100, File: daily_4/u196202.bin
>
> attempt #2:
> daily_4/u198409.bin
> daily_4/u198410.bin
> daily_4/u198411.bin
> % READUDAILY2: READU: Corrupted f77 unformatted file detected. Unit:
> 100, File: daily_4/u198411.bin
>
> attempt #3:
> daily_4/u199610.bin
> daily_4/u199611.bin
> % READUDAILY2: READU: Corrupted f77 unformatted file detected. Unit:
> 100, File: daily_4/u199611.bin
>
> As you can see, it fails on a different file each time. Although
> since it seemed to get further along on each successive trial, maybe I
> should try one or two more times and perhaps I'll get lucky??? :)
> Actually, I think this is just random and not a trend.
>
> I wonder if the same error would occur if I generated a large number
> of binary files of a smaller size (say a 5x5 array of random numbers),
> because maybe it's not how large the files are but how many of them
> there are. ?????
>
> Thanks,
> Mark
>

--
Paul van Delst Ride lots.
CIMSS @ NOAA/NCEP/EMC Eddy Merckx

Subject: Re: error reading a large number of binary files
Posted by Nigel Wade on Tue, 24 Apr 2007 08:38:27 GMT
View Forum Message <> Reply to Message

mark@atmos.colostate.edu wrote:

>
> Hi Paul. Thanks for taking an interest in this. On Friday I actually
> tried something similar to your idea of just reading the files and not
> doing any computations, and that also failed. And just now I took the
> program you created above, saved it as readudaily2.pro and tried it
> three times. Here are the results:
>
> attempt #1:
> daily_4/u196112.bin
> daily_4/u196201.bin
> daily_4/u196202.bin
> % READUDAILY2: READU: Corrupted f77 unformatted file detected. Unit:
> 100, File: daily_4/u196202.bin
>
> attempt #2:
> daily_4/u198409.bin
> daily_4/u198410.bin
> daily_4/u198411.bin
> % READUDAILY2: READU: Corrupted f77 unformatted file detected. Unit:
> 100, File: daily_4/u198411.bin
>
> attempt #3:
> daily_4/u199610.bin
> daily_4/u199611.bin
> % READUDAILY2: READU: Corrupted f77 unformatted file detected. Unit:
> 100, File: daily_4/u199611.bin
>
> As you can see, it fails on a different file each time. Although
> since it seemed to get further along on each successive trial, maybe I
> should try one or two more times and perhaps I'll get lucky??? :)
> Actually, I think this is just random and not a trend.
>
> I wonder if the same error would occur if I generated a large number
> of binary files of a smaller size (say a 5x5 array of random numbers),
> because maybe it's not how large the files are but how many of them
> there are. ?????
>
> Thanks,
> Mark

Just one more suggestion plucked out of the air.

Have you checked your system logs to see if you are getting read errors on the disk? Or perhaps you have a faulty memory module (if it were a PC I'd suggest memtest86, is there an equivalent for Macs?), or perhaps processor/memory overheating, a bad IDE/SCSI cable or some other hardware problem. It would be quite unusual for these to result in random corruption of the data returned by

a read, but not beyond the bounds of wild speculation...

--
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

Subject: Re: error reading a large number of binary files
Posted by Paul Van Delst[1] on Wed, 25 Apr 2007 14:21:24 GMT
View Forum Message <> Reply to Message

mark@atmos.colostate.edu wrote:
> ** UPDATE on the status of my problem **
>
> I decided to reprocess the original grib files with wgrib again but
> this time with the -nh switch turned on, which eliminates the f77-
> style header record. Then in the IDL read routine I just had to turn
> off the /f77_unformatted switch. Well, this has worked flawlessly in
> the three tests I have tried!!!
>
> That made me wonder if my problem was with IDL reading ANY fortran-
> style binary files, or just the ones created by wgrib. So I wrote a
> fortran-90 program to generate roughly the same size and number of
> binary files but just writing random numbers to the files, and then a
> simple readbin.pro IDL routine. Sure enough, it crashes in random
> places just like my previous set of ECWMF binary files with f77
> headers did. Here's some sample output from a few trials -- these
> random number files are just named x001.bin, x002.bin, ..., x520.bin.
>
> test #1
>>>> x062.bin
>>>> x063.bin
>>>> x064.bin
> % READU: Corrupted f77 unformatted file detected. Unit: 100, File:
> x064.bin
> % Execution halted at: READBIN2 10 /Users/mark/Datasets/
> era40/testbin/readbin2.pro
> % $MAIN$
>
> test #2
>>>> x300.bin
>>>> x301.bin
>>>> x302.bin
>>>> x303.bin
> % READU: Corrupted f77 unformatted file detected. Unit: 100, File:
> x303.bin

> % Execution halted at: READBIN2 10 /Users/mark/Datasets/
> era40/testbin/readbin2.pro
> % $MAIN$
>
> AND the problem seems to only occur once I make the files very large.
> i.e., when i just wrote out 520 files of arrays of size 5, it flies
> through all of those with no problems.

Interesting. Exactly how large do the individual files need to be for this problem to surface?

cheers,

paulv

> I'd really love to get to the bottom of this problem, and I should
> mention that one of the ITTVIS consultants has indeed been emailing
> with me and has been extremely helpful. But in the meantime I will
> obviously proceed with the "no-header" files for this project.
>
> Thanks to all of you who offered suggestions -- it's very helpful and
> encouraging!
>
> Cheers,
> Mark
>

--
Paul van Delst Ride lots.
CIMSS @ NOAA/NCEP/EMC Eddy Merckx

Subject: Re: error reading a large number of binary files
Posted by Mark Branson on Wed, 25 Apr 2007 20:32:16 GMT
View Forum Message <> Reply to Message

** UPDATE ON MY PROBLEM STATUS **

I decided to reprocess the original grib files with wgrib again but this time with the -nh switch turned on, which eliminates the f77-style header record. Then in the IDL read routine I just had to turn off the /f77_unformatted switch. Well, this has worked flawlessly in the numerous tests I have tried -- not a single error!!!

That made me wonder if my problem was with IDL reading ANY fortran-style binary files, or just the ones created by wgrib. So I wrote a fortran-90 program to generate roughly the same size and number of

binary files but just writing random numbers to the files, and then a simple readbin.pro IDL routine. Sure enough, it crashes in random places just like my previous set of ECWMF binary files with f77 headers did. Here's some sample output from a few trials -- these random number file are just named x001.bin, x002.bin, ..., x520.bin.

```
>>> x062.bin
>>> x063.bin
>>> x064.bin
% READU: Corrupted f77 unformatted file detected. Unit: 100, File:
x064.bin
% Execution halted at: READBIN2          10 /Users/mark/Datasets/
era40/testbin/readbin2.pro
%                       $MAIN$

>>> x300.bin
>>> x301.bin
>>> x302.bin
>>> x303.bin
% READU: Corrupted f77 unformatted file detected. Unit: 100, File:
x303.bin
% Execution halted at: READBIN2          10 /Users/mark/Datasets/
era40/testbin/readbin2.pro
%                       $MAIN$
```

AND the problem seems to only occur once I make the files very large. i.e., when i just wrote out 520 files of arrays of size 5, it flies through all of those with no problems. I have no idea why that would make a difference.

Thanks for the suggestions and ideas. You guys always think of things to try that never occur to me.

Cheers,
Mark

Subject: Re: error reading a large number of binary files
Posted by David Fanning on Wed, 25 Apr 2007 21:54:51 GMT
View Forum Message <> Reply to Message

mark@atmos.colostate.edu writes:

> Thanks for the suggestions and ideas. You guys always think of things
> to try that never occur to me.

One learns the art of avoiding real work only though years of hard-won experience. :-)

Cheers,

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

Subject: Re: error reading a large number of binary files
Posted by Mark Branson on Wed, 25 Apr 2007 22:40:03 GMT
View Forum Message <> Reply to Message

** UPDATE ON MY PROBLEM STATUS ** (repeated attempts to post this have failed for some strange reason????)

I decided to reprocess the original grib files with wgrib again but this time with the -nh switch turned on, which eliminates the f77-style header record. Then in the IDL read routine I just had to turn off the /f77_unformatted switch. Well, this has worked flawlessly in the numerous tests I have tried -- not a single error!!!

That made me wonder if my problem was with IDL reading ANY fortran-style binary files, or just the ones created by wgrib. So I wrote a fortran-90 program to generate roughly the same size and number of binary files but just writing random numbers to the files, and then a simple readbin.pro IDL routine. Sure enough, it crashes in random places just like my previous set of ECWMF binary files with f77 headers did. Here's some sample output from a few trials -- these random number file are just named x001.bin, x002.bin, ..., x520.bin.

```
>>> x062.bin
>>> x063.bin
>>> x064.bin
% READU: Corrupted f77 unformatted file detected. Unit: 100, File:
x064.bin
% Execution halted at: READBIN2          10 /Users/mark/Datasets/
era40/testbin/readbin2.pro
%                       $MAIN$

>>> x300.bin
>>> x301.bin
>>> x302.bin
>>> x303.bin
% READU: Corrupted f77 unformatted file detected. Unit: 100, File:
x303.bin
```

% Execution halted at: READBIN2 10 /Users/mark/Datasets/era40/testbin/readbin2.pro
% $MAIN$

AND the problem seems to only occur once I make the files very large. i.e., when i just wrote out 520 files of arrays of size 5, it flies through all of those with no problems. I have no idea why that would make a difference.

Thanks for the suggestions and ideas. You guys always think of things to try that never occur to me.

Cheers,
Mark

Subject: Re: error reading a large number of binary files
Posted by Mark Branson on Wed, 25 Apr 2007 22:47:18 GMT
View Forum Message <> Reply to Message

** UPDATE ON MY PROBLEM STATUS **

I decided to reprocess the original grib files with wgrib again but this time with the -nh switch turned on, which eliminates the f77-style header record. Then in the IDL read routine I just had to turn off the /f77_unformatted switch. Well, this has worked flawlessly in the numerous tests I have tried -- not a single error!!!

That made me wonder if my problem was with IDL reading ANY fortran-style binary files, or just the ones created by wgrib. So I wrote a fortran-90 program to generate roughly the same size and number of binary files but just writing random numbers to the files, and then a simple readbin.pro IDL routine. Sure enough, it crashes in random places just like my previous set of ECWMF binary files with f77 headers did. Here's some sample output from a few trials -- these random number file are just named x001.bin, x002.bin, ..., x520.bin.

>>> x062.bin
>>> x063.bin
>>> x064.bin
% READU: Corrupted f77 unformatted file detected. Unit: 100, File: x064.bin
% Execution halted at: READBIN2 10 /Users/mark/Datasets/era40/testbin/readbin2.pro
% $MAIN$

>>> x300.bin
>>> x301.bin

>>> x302.bin
>>> x303.bin
% READU: Corrupted f77 unformatted file detected. Unit: 100, File:
x303.bin
% Execution halted at: READBIN2 10 /Users/mark/Datasets/
era40/testbin/readbin2.pro
% $MAIN$

AND the problem seems to only occur once I make the files very large.
i.e., when i just wrote out 520 files of arrays of size 5, it flies
through all of those with no problems. I have no idea why that would
make a difference.

Thanks for the suggestions and ideas. You guys always think of things
to try that never occur to me.

Cheers,
Mark

Subject: Re: error reading a large number of binary files
Posted by Paul Van Delst[1] on Fri, 27 Apr 2007 19:35:33 GMT
View Forum Message <> Reply to Message

mark@atmos.colostate.edu wrote:
> ** UPDATE ON MY PROBLEM STATUS ** (repeated attempts to post this have
> failed for some strange reason????)

No, google news has apparently been having some issues lately.

Although you can't tell, your post *has* made it through. Multiple times.

:o)

cheers,

paulv

--
Paul van Delst Ride lots.
CIMSS @ NOAA/NCEP/EMC Eddy Merckx
