
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, infile[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)
infile = FILE_SEARCH('daily_4/u*.bin')
icnt = 0L & totcnt = 0L

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

  ; Read the file
  openr, lun, infile[i], /get_lun, /f77_unformatted
  print, infile[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