Subject: File pointer problem on PC-IDL
Posted by landsman on Thu, 11 Mar 1999 08:00:00 GMT
View Forum Message <> Reply to Message

I am trying to debug a program (a "FITS" reader) that is failing on PC-IDL
Unfortunately, I don't have access to PC-IDL myself so I apologizing for not
doing as much debugging as I would have liked before posting my problem.

Basically, then internal file position is getting screwed up. I open a
binary file, readu 2880 bytes, and find myself located 4095 bytes into the
file. Below is the simple script (as done by a remote user with PC-IDL).

```
IDL> print, !version
{ x86 Win32 Windows 5.2 Oct 30 1998}

IDL>  openr,lun,'C:\burst_image_1.fit',/block,error=error,/get_lun
IDL> if eof(lun) then message, 'Reached end of file'
IDL> buf = replicate(32b,80,36)                 ;Total of 2880 bytes
IDL> readu,lun,buf
IDL> help,/str,fstat(lun)
** Structure FSTAT, 12 tags, length=36:
   UNIT            LONG              100
   NAME            STRING    'C:\ROTSE\GRB 990123\burst_image_1.fit'
   OPEN            BYTE         1
   ISATTY          BYTE         0
   ISAGUI          BYTE         0
   INTERACTIVE     BYTE         0
   READ            BYTE         1
   WRITE           BYTE         0
   TRANSFER_COUNT  LONG           2880
   CUR_PTR         LONG           4095
   SIZE            LONG          25920
   REC_LEN         LONG              0
```

How did the current pointer get at 4095 bytes??? Calling EOF() is not
supposed to change the file position. The situation seems reminiscent of
a VMS mode where one always reads in multiple of 512 bytes. (To avoid this
problem is why the VMS-only /BLOCK keyword is added to the OPENR statement
above.) But I wasn't aware of such fixed block I/O on a PC.

Anyone have any ideas? (The file burst_image_1.fit is available at
ftp://idlastro.gsfc.nasa.gov/landsman/fits)

thanks, --Wayne Landsman landsman@mpb.gsfc.nasa.gov

Subject: Re: File pointer problem on PC-IDL
Posted by menakkis on Thu, 18 Mar 1999 08:00:00 GMT
View Forum Message <> Reply to Message

landsman@stars.gsfc.nasa.gov (Wayne Landsman) wrote:
<...>
> Basically, then internal file position is getting screwed up. I open a
> binary file, readu 2880 bytes, and find myself located 4095 bytes into the
> file.
<...>

Wayne, don't ask me why, but ask your remote user to include the switch
/NOAUTOMODE in the OPEN call.

Oh well, it's hard to turn down any opportunity to pick on RSI for this
little Windows feature of theirs. :-) It has bothered me since the day it
sidled onto the scene, announced - as it were - somewhere in the bowels of
the readme file. Basically IDL's I/O on Windows has a "text mode" and a
"binary mode". One of these (the "binary" mode, would you believe?) fools
around with <CR><LF> pairs. For some reason your EOF check is triggering IDL
to switch to "binary" mode or something. (I haven't checked, but it seems
unlikely to me that your file would have enough <CR><LF> pairs to fool with
to account for the discrepancy - it looks more like a bug.) Another strange
thing is that (in this case) if you use the /BINARY switch in the OPEN call
it will also fix the problem. (I don't even want to speculate on this one.)

Peter Mason

Subject: Re: File pointer problem on PC-IDL
Posted by Craig Markwardt on Thu, 18 Mar 1999 08:00:00 GMT
View Forum Message <> Reply to Message

landsman@stars.gsfc.nasa.gov (Wayne Landsman) writes:
>
> ... discussion of binary files under Windows ...
>

On a side note, I have found that ON_IOERROR resets the file pointer,
at least in reading from pipes under Unix. This was a real
show-stopper until I realized what was going on, since you can't use
POINT_LUN on a pipe. I had to remove my ON_IOERROR calls.

Has anybody else experienced the same?

Craig

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@astrog.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------

Subject: Re: File pointer problem on PC-IDL
Posted by landsman on Thu, 18 Mar 1999 08:00:00 GMT
View Forum Message <> Reply to Message

In article <7cpol7$k4h$1@nnrp1.dejanews.com>, Peter Mason <menakkis@my-dejanews.com> writes...
> landsman@stars.gsfc.nasa.gov (Wayne Landsman) wrote:
> <...>
>> Basically, then internal file position is getting screwed up. I open a
>> binary file, readu 2880 bytes, and find myself located 4095 bytes into the
>> file.
> <...>
>
> Wayne, don't ask me why, but ask your remote user to include the switch
> /NOAUTOMODE in the OPEN call.
>
Peter, thanks for the info. The IDL documentation certainly gives the
impression that the Windows-only /NOAUTOMODE and /BINARY keywords to OPEN
have an effect only when *writing* to a file. I guess my example shows
that one has to be aware of flipping between "text" mode and "binary"
mode, even when reading a existing file opened with OPENR.

--Wayne Landsman landsman@mpb.gsfc.nasa.gov

Subject: Re: File Pointer
Posted by David Fanning on Mon, 06 Dec 2004 21:28:34 GMT
View Forum Message <> Reply to Message

Pravesh writes:

> HOw do i implement a feature, that on clicking of a button will
> automatically display the next file in the directory. the files are
> images.

Well, you just go to the next file in the list of files

you are keeping, read the file, and display the image.
Then you probably update your file position pointer because
you always want to know where you are in your list.

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: File Pointer
Posted by Paul Van Delst[1] on Mon, 06 Dec 2004 22:50:53 GMT
View Forum Message <> Reply to Message

Pravesh wrote:
> Hi:
> HOw do i implement a feature, that on clicking of a button will
> automatically display the next file in the directory. the files are
> images.

Some sort of widget code that, e.g., after using findfile to get a list of the requested files (*.png or *.gif or whatever), displays the first one in the list. The button widget then just goes through the list with its event handler displaying the file.

paulv

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC
