Subject: Re: WHILE NOT EOF(lun) wierdness under IDL 4.0.1 on Win NT
Posted by Joseph M Zawodny on Tue, 18 Jun 1996 07:00:00 GMT
View Forum Message <> Reply to Message

Peter Mason wrote:
>
> Since Idl version 4, I found I have to use the /BINARY switch (and
> /NOAUTOMODE, to be safe) on my OPEN calls under Windows, to get things
> to work as expected.
> I first noticed a problem while using READU/WRITEU with structures and
> a binary file opened with /XDR - the reads and writes would sometimes
> (unpredictably) work incorrectly - I sometimes couldn't read back into
> the structure from which I had written (and it contained no strings :).
> Using the above switches sorted it out.
>
> I didn't notice problems like the one you described above (still can't
> understand why the code would work from the command line and not from
> within a routine), but try OPEN,/BINARY,/NOAUTOMODE - it's worth a shot.
>
> Peter Mason

Well this worked for me, but where did you find out about these keywords?
I did not see them in the manual under the OPEN statement, nor in either
the online help or the little "IDL For Windows" booklet. It is hard to
solve a problem when you do not know what tools are at your disposal.

Thanks again,

--
Work: Dr. Joseph M. Zawodny   Play: Joe Zawodny
      NASA Langley Research Center   KO4LW@amsat.org
      E-mail: J.M.Zawodny@LaRC.NASA.gov   zawodny@exis.net

Subject: Re: WHILE NOT EOF(lun) wierdness under IDL 4.0.1 on Win NT
Posted by Peter Mason on Tue, 18 Jun 1996 07:00:00 GMT
View Forum Message <> Reply to Message

On Mon, 17 Jun 1996, Joseph M. Zawodny wrote:
> I just ran across a very strange "feature" on IDL 4.0.1
> running on WIN-NT V3.51 and a Pentuim Pro. Basically the problem
> goes like this. I have a routine which reads from a previously
> opened file. It read in a structure and then 2 variable length
> records. The file is unformatted and access using READU. One of
> the fields in the structure contains the length of the arrays in
> the following records. If I call the routine (I called it READ_REC)
> manually, things are fine and I can read through the entire file
> without any problem. Similarly, I can put READ_REC in a FOR loop

> and work my way through the file. My intent has always been to
> use a WHILE NOT EOF(lun) construct with a READ_REC call within
> the WHILE block. This does not work! I can read the first two
> sets of three records correctly and on the third call, I read
> just the first three fields in the structure before hitting an EOF.
> IDL really thinks that it is at the end of the file and I cannot
> read any other data from the file. This is very reproducible.
>
> Has anyone else run into this? I certainly have not seen this
> happen on any of the UNIX versions of IDL I've used. Is this
> just another IDL 4 Windows "feature" (like the lack of support
> for 32 bit file systems - ie. long file names)?

Since Idl version 4, I found I have to use the /BINARY switch (and
/NOAUTOMODE, to be safe) on my OPEN calls under Windows, to get things to
work as expected.
I first noticed a problem while using READU/WRITEU with structures and a
binary file opened with /XDR - the reads and writes would sometimes
(unpredictably) work incorrectly - I sometimes couldn't read back into
the structure from which I had written (and it contained no strings :).
Using the above switches sorted it out.

I didn't notice problems like the one you described above (still can't
understand why the code would work from the command line and not from
within a routine), but try OPEN,/BINARY,/NOAUTOMODE - it's worth a shot.

Peter Mason

Subject: Re: WHILE NOT EOF(lun) wierdness under IDL 4.0.1 on Win NT
Posted by Peter Mason on Wed, 19 Jun 1996 07:00:00 GMT

View Forum Message <> Reply to Message

> Well this worked for me, but where did you find out about these keywords?
> I did not see them in the manual under the OPEN statement, nor in either
> the online help or the little "IDL For Windows" booklet. It is hard to
> solve a problem when you do not know what tools are at your disposal.

The documentation for the Windows-specific BINARY and NOAUTOMODE Open
keywords is stuck away in the readme.wri file which comes with IDL 4.x.
I thought I'd post this reply to the group because I'm sure many people
don't generally read "readme"s in search of topics which should be dealt
with in the manuals.   (I just happened to read this one by chance.)

While I'm at it, I'd like to add that I find said documentation a little
confusing: The default mode is text-mode, which reportedly passes
<CR><LF> pairs intact when "text" is written (presumably via printf but

in practice via any means and with any kind of data containing adjacent [13B,10B], apparently?), while binary mode reportedly strips out the <CR> from a <CR><LF> pair. In other words, funny things can happen to your data in binary mode, according to the documentation.
I think that things actually work the other way round, and that binary mode is the one to use to avoid data filtering.

Peter Mason