
Subject: Re: Reading F77 on Windows platforms

Posted by [Paul Van Delst\[1\]](#) on Wed, 24 Apr 2002 13:14:44 GMT

[View Forum Message](#) <> [Reply to Message](#)

Sverre Solberg wrote:

>
> I have problems reading F77-unformatted files when running idl on a
> Windows 2000
> platform. Opening the file (with /f77) works, but as soon as I try to
> read something, an error message claiming 'corrupted f77 file' shows
> up. The program works fine though on a Unix station, so the f77-file
> should be ok. According to the manual (for idl 5.5) the F77 should
> work both on Unix and Windows.
>
> In Fortran the file is created like this:
> open (<unit>,file=<name>, form='unformatted')
> write (<unit>) var1, var2, ...
>
> If opening the file in idl without the '/f77' keyword I dont get any
> complaints about corrupt file, but I then get problems retrieving the
> data. On Unix I am able to trace the file content in 4-bytes words
> (word by word) and it looks all fine. On Windows, however, I dont get
> the same results. Why??

Play around with the "swap endian" keywords on the IDL open statement. Moving from a big-endian

(most unix) to little-endian (PC) also swaps the record length markers at the beginning and end of each record that is written in the Fortran code. So, by opening with /f77_unformatted and no byte swapping keyword, your first record of length, say for e.g., 16 bytes, will end up being interpreted as 238573485837 bytes. (or something like that....or maybe even negative!).

If you must use Fortran unformatted sequential access files across platforms then my advice - in your fortran code, the first thing you should write is a 4-byte unique "magic" number (I use 123456789). In IDL when you open the file, open it *without* the /f77_unformatted keyword. Read the first 8-bytes, toss the first four (the record length of the first record), and check that the value of the remaining 4-bytes is 123456789. If it is, set a variable swap = 0. If not, check if the value is SWAP_ENDIAN(123456789). If this is true, set the variable swap = 1. Close the file and then open it again using /F77_unformatted, SWAP_ENDIAN = swap. If the magic number value is neither 123456789 or SWAP_ENDIAN(123456789) then you have a file that requires your personal intervention to read. :o)

Alternatively, you could adopt the convention of always writing in big-endian format and use the SWAP_IF_LITTLE_ENDIAN keyword on the IDL open.

The best method (I think) - output in netCDF format. Then your read and write functions are the same across platforms. A bit more work up front, but you'll be laughing later on.

paulv

--

Paul van Delst Religious and cultural
CIMSS @ NOAA/NCEP purity is a fundamentalist
Ph: (301)763-8000 x7274 fantasy
Fax:(301)763-8545 V.S.Naipaul

Subject: Re: Reading F77 on Windows platforms
Posted by [SSO](#) on Thu, 25 Apr 2002 09:36:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

Many thanks!

The addition of the 'swap_if_little_endian' keyword (in all open statements) solved the whole problem. However, as I understand you this may differ from platform to platform, so that exporting the program now working on windows-2000 (thats NT) to e.g. a Windows-98 may still create trouble(?).

(My problem is reading big (100Mb) binary fortran files efficiently, and this now works perfectly using ASSOC, by adding the longword before/after each block, although the manual states that binary fortran files should not be read with assoc)

Sverre

Subject: Re: Reading F77 on Windows platforms
Posted by [Paul Van Delst\[1\]](#) on Thu, 25 Apr 2002 12:32:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

Sverre Solberg wrote:

>
> Many thanks!
>
> The addition of the 'swap_if_little_endian' keyword (in all open
> statements) solved the whole problem. However, as I understand you
> this may differ from platform to platform, so that exporting the
> program now working on windows-2000 (thats NT) to e.g. a Windows-98
> may still create trouble(?).

No - unless someone somewhere has ported Windows to run on big-endian platforms (!?!??). By using the swap_if_little_endian keyword you are implicitly assuming that the data files will *always* be in big-endian "format". If you run the Fortran code that generates the binary files on a little-endian platform (PC, and maybe Dec Alpha?) and also run your IDL reader on a

little-endian platform, you will get the same problem because your little-endian data is being swapped into big-endian format. That's why I generally avoid the `swap_if_XXX_endian` keywords and test the data directly to determine if it needs swapping. Of course, that's not always possible.

- > (My problem is reading big (100Mb) binary fortran files efficiently,
- > and this now works perfectly using ASSOC, by adding the longword
- > before/after each block, although the manual states that binary
- > fortran files should not be read with assoc)

Well, I think by using ASSOC in IDL you are assuming that the record length of what you want to read is fixed, i.e. the records are all the same length. In that case, why not output the binary data in the Fortran code as DIRECT access, UNFORMATTED form? Then you won't have the pesky (and unneeded) record markers around each record and you also won't have to specify them in the ASSOC set-up.

Also, and probably more importantly, the DIRECT, UNFORMATTED type of Fortran output file is simply a stream of bytes - no record info whatever. So, apart from the byte swap issue, it's pretty portable. The SEQUENTIAL, UNFORMATTED type of Fortran output file *typically* has 4-bytes at the begin and end of each record *but that is not guaranteed by the Fortran standard*. So you could conceivably come across a Fortran compiler that uses only 2-bytes as record markers (quite unlikely, but not inconceivable...particularly for f77 compilers). That will create output files that will choke your IDL reader.

The amount of effort one should put into this sort of thing is directly proportional to the number of platforms the code has to work on. The more portable the solution, the more effort is required.

paulv

--

Paul van Delst Religious and cultural
CIMSS @ NOAA/NCEP purity is a fundamentalist
Ph: (301)763-8000 x7274 fantasy
Fax:(301)763-8545 V.S.Naipaul
