
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
