Subject: Reading unformatted fortran files on Windows platform
Posted by Richard Eckman on Mon, 10 Aug 1998 07:00:00 GMT
View Forum Message <> Reply to Message

I just moved over to using IDL on a Windows NT platform. I'm running Fortran applications using Visual Digital Fortran which maintains compatibility with our array of Digital Alpha workstations. The output from Fortran is written with format="unformatted". On a UNIX box, these files may be easily opened and read with the /f77 in IDL, which is missing from the Windows version of IDL.

In order to read these fortran files (produced on our Windows box), I've found no better way than to read a "junk" longword before and after each record. It's a bit of mess modifying existing code to run on both Unix and Windows platforms. Some months ago, I posted a message asking if anyone had any easier method of reading these unformatted fortran files on a Windows box. I received no really encouraging replies at that time. I'm wondering if anyone recently has run into this same problem and has a good solution for it? Ideally, it would be nice for RSI to enable the use of the /f77 keyword in Windows-based IDL since there are Fortran compilers that will write this type of unformatted file under Windows.

Thanks, in advance, for any assistance.

Richard Eckman
NASA Langley
Hampton, VA

r.s.eckman@larc.nasa.gov

Subject: Re: reading unformatted fortran
Posted by Liam Gumley on Mon, 13 Dec 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Morwenna Grifiths wrote:
> I'm having difficulty reading unformatted fortran in IDL. I think I am
> doing just what the manual tells me to, but I assume I'm not! Here's a
> sample program.
>
> The fortran code to write the unformatted data:
>   program write_out
>   integer bb
>   real aa

>
>   write(6,*) 'in write_out now'
>   open(unit=15,file='t1',form='unformatted',status='unknown')
>   bb = 23
>   write(15) bb
>   close(unit=15)
>
>   open(unit=16,file='t2',form='unformatted',status='unknown')
>   aa=47.0
>   write(16) aa
>
>   close(unit=16)
>
>   stop
>   end
>
> The IDL code that doesn't work:
> a=1 & b=1 & c=1 & d=1 & e=1 & f=1 & aa=1.0
>
> openr,1,'t1',/f77_unformatted
> readu,1,a
> close,1
> print,'should be',a ; gives an answer of 0
>
> openr,1,'t1',f77_unformatted
> readu,1,a,b,c,d,e,f
> print,'but it is: ',a,b,c,d,e,f ; gives answers of 0 4 0 23 0 4, so the
> 4th number is correct
> close,1
>
> openr,2,'t2',f77_unformatted
> readu,2,aa
> print,'real number ',aa ; gives an answer of 5.60519e-45, no idea why!
>
> close,/all
>
> So it seems that IDL is reading some extra characters before and after
> the integer, and I have no idea what's happening with the real data.
>
> I'm running both the fortran and IDL on the same machine (a silicon
> graphics).

I compiled and ran your program on my SGI (Origin2000, Irix 6.5) using

% f77 write_out.f
% a.out

The following IDL commands read the file just fine:

```
openr, 1, 't1', /f77_unformatted
bb = 0L
readu, 1, bb
print, bb
close, 1

openr, 1, 't2', /f77_unformatted
aa = 0.0
readu, 1, aa
print, aa
close, 1
```

Note that
(1) /f77_unformatted is a keyword (not an argument), hence the leading slash,
(2) bb is defined as a long integer (0L),
(3) aa is defined as a 32 bit float (0.0).

Cheers,
Liam.

--
Liam E. Gumley
Space Science and Engineering Center, UW-Madison
http://cimss.ssec.wisc.edu/~gumley

Subject: Re: reading unformatted fortran
Posted by Nigel Wade on Mon, 13 Dec 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Morwenna Grifiths wrote:
>
> I'm having difficulty reading unformatted fortran in IDL. I think I am
> doing just what the manual tells me to, but I assume I'm not! Here's a
> sample program.
>
> The fortran code to write the unformatted data:
>
```
>       program write_out
>       integer bb
>       real aa
>
>       write(6,*) 'in write_out now'
>       open(unit=15,file='t1',form='unformatted',status='unknown')
>       bb = 23
>       write(15) bb
>       close(unit=15)
```

>
>   open(unit=16,file='t2',form='unformatted',status='unknown')
>   aa=47.0
>   write(16) aa
>
>   close(unit=16)
>
>   stop
>   end
>
> The IDL code that doesn't work:
> a=1 & b=1 & c=1 & d=1 & e=1 & f=1 & aa=1.0
>
> openr,1,'t1',/f77_unformatted
> readu,1,a
> close,1
> print,'should be',a ; gives an answer of 0

Here you are reading a 16 bit integer from the file
rather than a 32 bit integer. IDL defaults to 16 bit
integers whereas IRIX defaults to 32bit (unless you are
running IRIX64). To get IDL to create a 32 bit integer
define a=1L.

>
> openr,1,'t1',f77_unformatted
> readu,1,a,b,c,d,e,f
> print,'but it is: ',a,b,c,d,e,f ; gives answers of 0 4 0 23 0 4, so the
> 4th number is correct
> close,1

Here you have read the file as plain binary (the f77_unformatted is
not a keyword, it is an argument as there is no /). So you read the file
as it was written by FORTRAN - here it has a 32 bit byte count followed
by the data followed by a trailing 32 bit byte count.

>
> openr,2,'t2',f77_unformatted
> readu,2,aa
> print,'real number ',aa ; gives an answer of 5.60519e-45, no idea why!
>

Here again, f77_unformatted is an argument not a keyword so IDL is not
reading FORTRAN data. It's loading the first 4 bytes (0x00000004), the
byte count, into the floating point value.

Change it to /f77_unformatted and you should get the 47.0 you expect.

> close,/all
>
> So it seems that IDL is reading some extra characters before and after
> the integer, and I have no idea what's happening with the real data.

Binary data is alway fun. Especially when it is created by FORTRAN.
AFAIK those byte counts at the start and end of the FORTRAN unformatted records are implementation dependant - the FORTRAN standard leaves it up to the compiler writers as to how they store unformatted data.
So you can't reliably transfer unformatted FORTRAN data from one machine to another; maybe not even on the same machine if you have executables created with different compilers.

>
> I'm running both the fortran and IDL on the same machine (a silicon
> graphics).
>
> Can anyone help me, please?

If you really have to read FORTRAN unformatted data into non-FORTRAN programs be very careful.

>
> Morwenna

--

Nigel Wade, System Administrator, Space Plasma Physics Group,
University of Leicester, Leicester, LE1 7RH, UK
E-mail : nmw@ion.le.ac.uk
Phone : +44 (0)116 2523568, Fax : +44 (0)116 2523555