
Subject: Re: READU on IRX and Win95

Posted by [Chris Jengo](#) on Tue, 20 Jul 1999 07:00:00 GMT

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

- > When I use READU on the same file in Win95 and IRIX, I get different
- > results.
- > Is this possible? How would I read the file on a Win95 system?

Looks like a byte swapping problem. No matter how many times this happens to me, it always takes me a few minutes to figure out what the heck is going on! In ENVI I just switch it in the header, but in IDL it looks like BYTEORDER will do the trick (going from network (IEEE) to Host (Intel), use the XDRTOF keyword).

Chris

Chris Jengo

Imaging Scientist

Earth Satellite Corporation

(301)231-0660

cjengo@earthsat.com

Subject: Re: READU on IRX and Win95

Posted by [Liam Gumley](#) on Tue, 20 Jul 1999 07:00:00 GMT

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

Paul O Storaasli wrote:

- > When I use READU on the same file in Win95 and IRIX, I get different
- > results. Is this possible?

Oh yes. A quote from the link referenced below:

"Little and Big-Endian machines

Another practical complication arises when you store any multi-byte entity in memory.

Computer memory is referenced by addresses that are positive integers. It is 'natural' to store numbers with the LEAST SIGNIFICANT BYTE coming before the MOST SIGNIFICANT BYTE in the computer memory, however computer designers prefer sometimes to use a reversed order version of the representation.

The 'natural' order, where less significant binary digits comes before more significant digits in memory is called LITTLE-ENDIAN, many vendors like IBM, CRAY and Sun preferred the reverse order that of course is

called BIG-ENDIAN."

How would I read the file on a Win95 system?

```
> I am trying to read a binary file of 32-bit floats with the following
> code:
>
> function readfile
> ; This function will read the data in 'D:\data\thunderb.653'
> ; as an array of floating point (32-bit) numbers. It assumes
> ; the array has 716 columns, and calculates the number of rows.
> ; It returns the array.
>
> file='thunderb.653'
>
> openr,LUN,file,/Get_Lun
> result=fstat(LUN)
> nbytes=result.size
> nx=716L
> ny=nbytes / 4L /nx
> image=fltarr(nx,ny)
> readu,LUN,image
> free_lun, LUN
>
> return, image
> end
>
> The results I get are quite different:
>
> On Win95
> IDL> print,a(1:5,1:5)
> -6.32813e-023-4.27821e+008-6.32846e-023-4.27903e+008 1.08637e+024
> 9.04917e-041-4.27821e+008 8.09951e-043 3.67981e-042 4.16997e-008
> 9.24857e-044-6.32813e-023-4.27838e+008-6.33169e-023-6.33524e -023
> 1.08622e+024 8.09951e-043 4.16942e-008-6.33718e-023-4.28427e+008
> -4.27903e+008 3.67981e-042 1.44418e-041-4.28427e+008-6.34203e-023
> % Program caused arithmetic error: Floating underflow
>
> On an IRIX system:
> IDL> print,a(1:5,1:5)
> 32.4000 32.2000 32.9000 34.7000 35.1000
> 31.5000 32.2000 32.5000 34.5000 38.3000
> 32.0000 32.4000 32.7000 37.9000 43.4000
> 33.1000 32.5000 36.8000 46.4000 50.7000
> 34.7000 34.5000 42.0000 50.7000 53.9000
```

Under Win95, try typing

```
IDL> print, swap_endian(a[1:5,1:5])
```

If the data file is created under Irix in IDL, add the /XDR keyword to your OPENW and OPENR statements. This will cause the data to be written and read in a format which is portable between different IDL machine architectures.

If the data file is created under Irix in some other application, add the keyword /SWAP_IF_LITTLE_ENDIAN to the OPENR statement, which will cause the data to be swapped if the file is opened on a little-endian system, like a PC.

For a full explanation of this behavior (which is not specific to IDL), check out <http://metalab.unc.edu/pub/languages/fortran/ch4-3.html>

Cheers,
Liam.

--

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