
Subject: Re: OPENR F77 SWAP_IF_LITTLE_ENDIAN
Posted by [Paul Van Delst\[1\]](#) on Mon, 06 Apr 2015 20:00:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 04/06/15 15:43, fawltlanguage@gmail.com wrote:
> On Monday, April 6, 2015 at 8:04:05 PM UTC+2, Edward Hyer wrote:
>> ; open and write a file using F77_UNFORMATTED
>> IDL> openw,lun,/get_lun,/f77,'/tmp/test_f77'
>> IDL> writeu,lun,long(2015032518),float(6),long(0)
>> IDL> free_lun,lun
>> ; read the file back in
>> IDL> openr,lun,/get_lun,/f77,'/tmp/test_conc'
>> IDL> qq1=long(0) & qq2=float(0) & qq3=long(0)
>> IDL> readu,lun,qq1,qq2,qq3
>> IDL> print,qq1,qq2,qq3
>> 2013052306 6.00000 0
>> IDL> free_lun,lun
>> ; now try reading it with /SWAP_IF_LITTLE_ENDIAN
>> IDL> openr,lun,/get_lun,/f77,'/tmp/test_conc',/swap_if_little_end_ian
>> IDL> readu,lun,qq1,qq2,qq3
>> % READU: Corrupted f77 unformatted file detected. Unit: 100
>> File: /tmp/test_conc
>> % Execution halted at: \$MAIN\$
>>
>>
>> Has anyone encountered this behavior?
>>
>> --Edward H.
>
> F77 unformatted files contain record length data. Byte swapping them
> will result in corrupted records.

Oh, it didn't even occur to me that the OP didn't know about the
(ubiquitous but non-standard) format of Fortran unformatted sequential
files.

For the OP (apolgoies if you do know this):

Fortran unformatted sequential files have the following structure:

```
[next record length]
[record]
[previous record length]
```

repeated for each record. That is, each record is bookended with a
record marker indicating the length of the record. Having the record
length at the end of the record as well is to allow code to BACKSPACE
records like one would have had to do on magnetic tapes back in old day.

At any rate, when you byte swap the data via the `"/swap_if_little_endian"` keyword, you are also byteswapping the record markers. And that leads to potentially screwed up reads.

Doing an octal dump of the original `*little-endian*` file:

```
$ od -t d4 tmp.test_f77
00000000      12 2015032518 1086324736      0
00000020      12
00000024
```

Note the "12" at the beginning and end. The record is 12 bytes long.

Now lets do the same, but for a purposefully written big-endian file:

```
IDL> openw,lun,/get_lun,/f77,'be.tmp.test_f77',/swap_endian
IDL> writeu,lun,long(2015032518),float(6),long(0)
IDL> free_lun,lun
```

```
$ od -t d4 be.tmp.test_f77
00000000 201326592 -957080968 49216      0
00000020 201326592
00000024
```

Because of the byteswapping, this is the equivalent of the little-endian file when the `"/swap_if_little_endian"` keyword is used. When this file is read IDL thinks the record should be 201326592 bytes long. Which is isn't.

Thus, error.

cheers,

paulv
