
Subject: Whats up in opening files: Unix vs. Windows
Posted by [Sean Heukels](#) on Tue, 13 Mar 2001 09:09:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

I use this code on a Unix System and it has been working fine for 4 years.

```
file_header={file_header, nblocks: 0L, ntraces: 0L, np: 0L, ebytes: 0L,  
tbytes: 0L, bbytes: 0L, transf: 0, status: 0, spare1: 0L}  
data_header={data_header, scale: 0, status: 0, index: 0, spare3: 0, ctcount:  
0L, lpval: 0.0, rpval: 0.0, lvl: 0.0, rvl: 0.0}
```

```
;----Open the data file  
openr, unit, infile, /get_lun ; Open file  
point_lun, unit, 0  
readu, unit, file_header ; Read the file header (only occurs once)
```

```
help, file_header, /st  
;----Define variables to store data and determine nature of data (int or  
lon)  
nb = FIX(file_header.nblocks) ; Total number of data blocks  
ns = FIX(file_header.ntraces) ; Number of slices  
nro = FIX(file_header.np)/2 ; NUmber of read-out point : Real and  
imaginery  
npe = FIX(READ_PROCPAR('nv', infile)) ; Get number of Phase-encode  
points
```

```
if (npe eq 0) then npe = FIX(READ_PROCPAR('ni', infile)) ; or 'ni' for  
previous versions  
nt = nb/npe ; Number of time points, for repeated measurements  
----> if (file_header.ebytes eq 2) then temp=intarr(nro*2) else  
temp=lonarr(nro*2); Adjust the complex data for Integer or Float form
```

The only thing is that the header is not read properly. It crashes on
file_header.ebytes, which is set to 0 (0*2=> array size ?? no way)
A piece of what it does read:

```
** Structure FILE_HEADER, 9 tags, length=32:  
NBLOCKS      LONG      3276800  
NTRACES      LONG      16777216
```

Should be more like 164 and 220.
Is there a difference in 0L in Windows and Unix ?? Or structures, or anything
lese?

Thanks Sean

Subject: Re: Whats up in opening files: Unix vs. Windows

Posted by [R.Bauer](#) on Mon, 19 Mar 2001 21:21:33 GMT

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

Sean Heukels wrote:

```
>
> I use this code on a Unix System and it has been working fine for 4 years.
>
> file_header={file_header, nblocks: 0L, ntraces: 0L, np: 0L, ebytes: 0L,
> tbytes: 0L, bbytes: 0L, transf: 0, status: 0, spare1: 0L}
> data_header={data_header, scale: 0, status: 0, index: 0, spare3: 0, ctcount:
> 0L, lpval: 0.0, rpval: 0.0, lvl: 0.0, rvl: 0.0}
>
> ;----Open the data file
> openr, unit, infile, /get_lun    ; Open file
> point_lun, unit, 0
> readu, unit, file_header    ; Read the file header (only occurs once)
>
```

I got some experience about point_lun problems with unix and windows.

I believe it's not possible to read an unix ascii file correctly with
usage

of point_lun if it's accessed over the network directly from a
unix file server. The point_lun action will be different used by windows
because a line ends by two bytes for windows and one byte for unix.

I always use a get_file and put_file method to transfer the file
into the windows %TEMP% directory and then I read from this file.

This works fine.

regards

Reimar

--

Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich

email: R.Bauer@fz-juelich.de
<http://www.fz-juelich.de/icg/icg1/>

=====

a IDL library at Forschungszentrum Jülich
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.html

<http://www.fz-juelich.de/zb/text/publikation/juel3786.html>
