Subject: Problems reading binary files - pointer at 4096 gives EOF
Posted by Oliver Smith on Wed, 22 Mar 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Hi,

I'm working on a program which loads data from a structured binary file.
Each
file contains many sets of different data types, each data field is
preceeded by a header(int) and fieldlength (long) before the data itself.
In order to read the files, I use a WHILE NOT EOF(file) loop as there is no
indication of the last field in the file. I've hit a major problem with
this, the EOF test reports end of file whenever the file pointer is at 4096.
The
problem is recreated below using a binary file consisting of only integers
created using findgen.

```
***********************************
binary file creation procedure

***********************************
PRO createbinary

OPENW, file, 'c:\temp\test.bin', /GET_LUN

data=INDGEN(2560)

WRITEU, file, data

FREE_LUN, file

END
***********************************
```

This first method of reading the binary file most closely represents the
actual methods used to read the data files.

```
********************************************************

PRO readbinary

OPENR, file, 'c:\temp\test.bin', /GET_LUN

data=INTARR(2560)
temp=0
c=0
```

```
WHILE NOT EOF(file) DO BEGIN
 POINT_LUN, -file, pos
 READU, file, temp
 PRINT, pos, temp, c
 c=c+1
ENDWHILE

FREE_LUN, file

END
```

****

Note the very first iteration of the loop reads the correct data value, but reports a file pointer of 3584!

There is a dicontinuity when the pointer reaches 4096, the data skips from 25 to 2048, then carries on reading until finding the true end of file.

****

IDL Version 5.3.1 (Win32 x86). (c) 2000, Research Systems, Inc.

```
IDL> .COMPILE "C:\MyWork\Damson\IDL\readbinary.pro"
% Compiled module: READBINARY.
IDL> readbinary
    pointer(pos),   data(temp),  loop(c)
        3584      0      0
        4046      1      1
        4048      2      2
        4050      3      3
Snip...........................
        4090     23     23
        4092     24     24
        4094     25     25
        4096   2048     26
        4098   2049     27
        4100   2050     28
Snip...........................
        5114   2557    535
        5116   2558    536
        5118   2559    537
```

****

This alternative method of reading the file also produces errors

****

```
PRO readbinary

OPENR, file, 'c:\temp\test.bin', /GET_LUN

data=INTARR(2560)
temp=0
c=0

WHILE NOT EOF(file) DO BEGIN

 READU, file, data

ENDWHILE

FREE_LUN, file

END
```

 *** ****************
*****

Note when the procedure halts upon detecting EOF at pointer=4096, it is possible to read another integer (and it's correct)
It is also possible to wind the pointer on one byte, and the test for EOF returns false.

 *** ****************
*****

```
IDL Version 5.3.1 (Win32 x86). (c) 2000, Research Systems, Inc.

IDL> .COMPILE "C:\MyWork\Damson\IDL\readbinary.pro"
% Compiled module: READBINARY.
IDL> readbinary
% READU: End of file encountered. Unit: 100, File: c:\temp\test.bin
% Execution halted at:  READBINARY         11
C:\MyWork\Damson\IDL\readbinary.pro
%                       $MAIN$
IDL> point_lun, -file, pos
IDL> print, pos
        4096
IDL> print, eof(file)
       1
IDL> readu, file, temp
IDL> print, temp
    2048
```

```
*********************************************************

IDL Version 5.3.1 (Win32 x86). (c) 2000, Research Systems, Inc.

IDL> readbinary
% Compiled module: READBINARY.
% READU: End of file encountered. Unit: 100, File: c:\temp\test.bin
% Execution halted at:  READBINARY        11
C:\MyWork\Damson\IDL\readbinary.pro
%                    $MAIN$
IDL> point_lun, -file, pos
IDL> print, pos
       4096
IDL> print, eof(file)
      1
IDL> point_lun, file, pos+1
IDL> print, eof(file)
      0

**********************************************************
```

This simple procedure doesn't test for EOF, and works correctly. Whilst it is possible to read this example data file without using the WHILE NOT EOF(file) loop, this would not be feasible with the real-world data files I'm working with, as I never know the length of the file, or when I'm reading the last field.

```
*******************************************
PRO readbinary

OPENR, file, 'c:\temp\test.bin', /GET_LUN

data=INTARR(2560)

READU, file, data

FREE_LUN, file

END

*****************************************************
```

So, has anybody else experienced this behaviour in other versions of IDL (I've tried 5.3 and 5.31 on an Intel NT box)

Does anybody know of a fix / work around?

I'm currently waiting for a reply from RS inc support.

Regards,

Oliver Smith