## Subject: Re: Reading and Plotting big txt. File Posted by incognito.me on Thu, 02 Aug 2007 08:55:42 GMT

View Forum Message <> Reply to Message

```
On 1 Aug., 18:15, Conor <cmanc...@gmail.com> wrote:
> On Aug 1, 10:49 am, "incognito.me" <incognito...@gmx.de> wrote:
>
>
>
>
>
>> On 1 Aug., 14:44, Conor <cmanc...@gmail.com> wrote:
>>> On Aug 1, 6:25 am, greg.a...@googlemail.com wrote:
>>> On Aug 1, 11:33 am, "incognito.me" <incognito...@gmx.de> wrote:
>>> > I'm trying to read and plot (surface) a very big text (.txt) file
>>> > (1020, 1024) with a 5 line string Header in IDL. My file looks like a
>>>> circle made of numbers!!!. That means in some lines and colums there
>>>> are no numbers only blanks!!!for example my file contains integers
>>> > between rows 633 and 390 and between columns 650 and 406. At the left
>>> > side of the file, there are the numbers of rows (1023,1022,1021,....0)
>>>> > my code should not read, but it does. And I also notice, that my code
>>>> > don't begin to read where the data starts!!By running the code I have
>>>> the following error message: READF: End of file encountered. Unit: 1.
>>>> > Can someone help me?
>>>> > This is how my code looks like
>>>> > pro readfile, filename
>>>> >; file=strupcase(filename)
>>> > rows=file_lines(file)
>>> > ;open the file and read the five line header.
>>>> > openr,1,file
>>>> header=strarr(5)
>>>> readf,1,header
>>>> >; Find the number of columns in the file
>>> > cols=fix(strmid(header(3),14,4))
>>>> >; Number of rows of the data
>>>> rows data=rows-n elements(header)
>
>>>> Create a big array to hold the data
>>>> data=intarr (cols, rows_data)
>>>> > ; All blanks should be replaced by zero
>>>> data[where(data eq ' ')]=0
>>>> ; A small array to read a line
>>>> > s=intarr(cols)
>>>> n=0
```

```
while (~ eof(1) and (n lt rows_data -1 )) do begin
           ; Read a line of data
>>>> >
           readf,1,s
>>>> >
            ; Store it in data
>>>> >
             data[*,n]=s
>>>> >
             n=n+1
>>>> >
         end
>>>> >
>>> > data=data[*,0:n-1]
>>>> > CLOSE,1
>>>> > Shade_surf, data
>>>> > end
>>>> > thanks
>>>> > incognito
>>>> I'm suspicious of the line converting blanks to zeros before you've
>>> even read them. I don't think the blanks will come out the way you're
>>> expecting, anyway. I'd suggest you write a program to correctly read
>>> your first line of data before you go for the whole thing.
>>>> Greg
>>> For starters, I'm not sure why you are converting blanks to zeroes
>>> there at all. As far as I can tell, you haven't even initialized any
>>> data yet. It seems like you are trying to convert blanks to zeros on
>>> an integer array which is already filled with zeroes anyway. When I
>>> tried to do that, I got this error:
>>> % Type conversion error: Unable to convert given STRING to Integer.
>
>>> Which isn't a fatal error, so your code would still run but the line
>>> 'data[where(data eq ' ')]=0' wouldn't actually do anything. As for
>>> the rest of your problem, I think what you need is a format
>>> statement. I believe what is happening is that because you haven't
>>> included an explicit format statement (telling it how many columns are
>>> on each line) it simply reads in entries until it fills up a row in
>>> your data array. For instance, look at this file:
>>> 12 34 698 934
>>> 16
              18
>>> 17 20
               13
>>> 14 23 234 123
>>> being read by this pseudo-code:
>>> readf,lun,file,/get lun
```

```
>>> data = intarr(4)
>>> readf,lun,data
>>> print,data
>>> ; 12 34 698
                    934
>>> readf,lun,data
>>> print,data
>>> : 16 13 17
                    20
>>> readf,lun,data
>>> print,data
>>> ; 14 23
              234 123
>>> readf,lun,data
>>> % READF: End of file encountered. Unit: 100, File: test
>
>>> See, because you have no format specified, each readf keeps reading
>>> data in until the data array is filled. You are assuming that readf
>>> reads one line at a time, but that's not happening, which is why your
>>> data isn't where it's supposed to be. Also, because it is reading
>>> faster than one line at a time, you are reading to the end of the file
>>> before you call readf (rows data) times, and then you get the EOF
>>> error. The solution is to give it a format:
>
>>> IDL> openr,lun,'test',/get lun
>>> IDL> format = '(i3, 1x, i3, 1x, i3, 1x, i3)'
>>> IDL> readf,lun,test,format=format
>>> IDL> print,test
                    698
        12
               34
                           934
>>>
>>> IDL> readf,lun,test,format=format
>>> IDL> print,test
>>>
        16
               0
                     0
                          18
>>> IDL> readf,lun,test,format=format
>>> IDL> print.test
        17
               20
                      0
                           13
>>>
>>> IDL> readf,lun,test,format=format
>>> IDL> print,test
                           123- Zitierten Text ausblenden -
        14
               23
                    234
>>>
>>> - Zitierten Text anzeigen -
>
>> Hi Conor,
>> Thanks for your suggestions! I muss agree, to fill the blanks with
>> zeroes was not so cute!! have to read how one uses the keyword format
>> with readf again, because I should confest I haven't unsterstood
>> yet.Could you please give me a hint?
>> Thanks a lot.
>> Kind regards
>> C.
>
```

```
> Unfortunately, I'm not so great with format statements, I don't use
> them so much, and I've never used them for reading files. The general
> idea for reading floats is that you specify the total number of
> characters to read, and how many numbers come after the decimal
  place. So, for instance the number:
  123.456789
>
  would be specified by the statement:
>
 (f10.6)
>
>
  There are ten characters that must be read (9 digits, plus the decimal
  point) and there are 6 digits after the period. For spaces you use
  '1x' (or '2x' for two spaces, etc...). So for instance the line:
  134.367 123.45 123.92
 would be specified by:
>
  (f7.3, 1x, f6.2, 1x, f6.2)
>
 Also, you can specify that IDL should "repeat" a format statement.
  For instance, you could also represent the last one with:
>
  (f7.3, 2(1x, f6.2))
>
>
> This last part is very important to you because you won't want to
> write out the format statement for all 1000 of your columns. In fact,
> IDL won't let you specify that many anyway. With any luck, all the
> columns have the same fixed width (or at least a repeating pattern) so
> you can do something like this:
>
  (f10.5, 999(1x, f12.1))
>
> Exactly how it will work I don't know. You might just have to play
> around with it. As I said, I'm not terribly familiar with format
> statements myself, so this might not be the best way to do it. Maybe
 someone else has some suggestions?- Zitierten Text ausblenden -
> - Zitierten Text anzeigen -
Hi Conor,
I'm still having trouble .I did many tries with the format statement
and I'm not so successfull. Let's suppose my file ist not (1020, 1024)
```

Measurement results

but only (14,10). Here is how my data looks like:

```
Row=14 Col=10
Row\Col 0 1 2 3 4 5 6 7 8 9
13
12
11
10
9
     -1193 -1230 -1236 -1242 -1190 -1134 -1097
8
     -1570 -1545 -1557 -1588 -1591 -1604 -15767 -1539
7
     -1848 -1792 -1718 -1678 -1638 -1576 -1517 -1446 -1372 -1322
6
     -306 -312 -300 -318 -309 -278 -272 -241 -250 -222
5
     -596 -599 -584 -556 -501 -457 -420 -386 -349
4
     158 154 164 161 158 179 195 210 154
3
     284 306 346 334 315 334
2
        485 513 513 504 494 491
1
0
```

By using the following statement to read a line: readf,lun,test,format='((11x,(9(/,i+4.4,1x)),i+4.4))' and I'm having the following error message:End of input record encountered on file unit: 1. (I'm using actually the version 6.3 of IDL on a windows machine)

Can you please tell me what I'm doing wrong this time?

Can you please tell me what I'm doing wrong this time? Kind regards

C.