
Subject: Re: reading ASCII Data

Posted by [David Foster](#) on Tue, 13 Apr 1999 07:00:00 GMT

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

Mirko wrote:

>
> Hello
>
> We have ASCII files (floats) with a different number of lines and
> different number of data in each line. The data was generated by a mass
> spectrometer software under Win 3.11. Each measurement line ends with
> '0D'xb instead of '0D'xb and '0A'xb. Is there a way to find the END OF
> LINE with an IDL function (like EOF) because each line starts with some
> "header information" which has to be skipped before the measured data
> starts.
> How can you handle a variable number of floats?
>
> Cheers
>
> Mirko

A good start would be to use FILE_STRARR.PRO to read the file into an array of strings. This could be parsed fairly easily. I thought it'd be trivial using IDL's STR_SEP(), but you can't get rid of the extra space characters easily (it gets confused when two delimiting characters are next to each-other...dreadful programming!).

Anyways, FILE_STRARR.PRO and FILE_STRARR.DOC are included below.

Dave

--

```
~~~~~  
David S. Foster      Univ. of California, San Diego  
Programmer/Analyst  Brain Image Analysis Laboratory  
foster@bial1.ucsd.edu  Department of Psychiatry  
(619) 622-5892      8950 Via La Jolla Drive, Suite 2240  
La Jolla, CA 92037  
~~~~~
```

```
; FILE_STRARR.PRO 5-27-94  
;  
; Routine to read text file and return a STRARR containing  
; the lines of text. Returns a STRARR(2) containing the null  
; string " and the value of !err_string if an I/O error is encountered  
;  
; This code adapted from XDISPLAYFILE
```

```
FUNCTION file_strarr, fname
```

```
ON_IOERROR, IO_ERROR
```

```
openr, unit, fname, /get_lun, error=err
if (err ne 0) then begin
  return, ['ERROR', !err_string]
endif else begin
  max_lines = 1000
  a = strarr(max_lines)
  i = 0
  c = ""
  while (not eof(unit)) do begin
    readf, unit, c
    a[i] = c
    i = i + 1
    if (i eq max_lines - 2) then begin
      a = [a, strarr(max_lines)]
      max_lines = max_lines + max_lines
    endif
  endwhile
  a = a[0:i-1]
  free_lun, unit
  return, a
endelse
```

```
IO_ERROR:
```

```
message, 'Error reading file: ' + fname, /continue
print, !err_string
return, ['ERROR', !err_string]
```

```
END
```

FILE_STRARR

Use this routine to read a text file and return a STRARR variable containing the lines of text. This is useful when displaying text-files in text widgets. If an I/O error such as file-not-found occurs then returns a STRARR(2) containing the string 'ERROR' and the value of the IDL system variable !ERR_STRING.

Calling Sequence

```
Text = FILE_STRARR(Filename)
```

Arguments

Filename

The name of the file containing the text you wish to return as a STRARR.

Outputs

Returns the lines of text from the file as a STRARR variable. If an I/O error occurs then this will contain two elements: the string 'ERROR' and the value of the IDL system variable !ERR_STRING when the error occurred.

Example

```
; Locate an IDL program file

Ret = FIND_PROCEDURE('mrsegreg.pro',Filename)

; Put this file into a STRARR

Text = FILE_STRARR(Filename)

if (Text(0) eq 'ERROR' and $
    n_elements(Text) eq 2) then
    message, 'Error reading file ' + Filename
```

File Attachments

- 1) [file_strarr.pro](#), downloaded 96 times
 - 2) [file_strarr.doc](#), downloaded 115 times
-

Subject: Re: reading ASCII Data
Posted by [R.Bauer](#) on Tue, 13 Apr 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mirko wrote:

```
> Was in a hurry this mornig an missed to put an example on.
> 1  08.04.1999  15:27:12:40  11.04  1.05722E-007  2.14 9.9E-009  16.05
> 3.59375E-009  28.02 4.20938E-009  44.03 1.2875E-009
> 2  08.04.1999  15:27:21:60  20.6  1.07047E-007  2.19 1.04469E-008  16.05
> 3.66875E-009  17.06 1.89781E-008  18.09 6.69937E-008  28.02
> 4.11563E-009  31.94 1.49062E-009  43.97 1.35312E-009
```

```
> 3    08.04.1999    15:27:30:44    29.44    1.14278E-007    ... and so on for the next
> 700 ranges
>
> Any suggestions?
>
> Cheers
```

```
file='test.dat'
IF file_exist(file) THEN A = fileline('test.dat')
```

R.Bauer

```
; Copyright (c) 1998, Forschungszentrum Juelich GmbH ICG-1
; All rights reserved.
; Unauthorized reproduction prohibited.
; This software may be used, copied, or redistributed as long as it is not
; sold and this copyright notice is reproduced on each copy made. This
; routine is provided as is without any express or implied warranties
; whatsoever.
;
;+
; NAME:
;   file_exist
;
; PURPOSE:
;   The result of this function is 1 if a file exist and 0 if not
;
; CATEGORY:
;   DATAFILES
;
; CALLING SEQUENCE:
;   Result=file_exist(file_name)
;
; INPUTS:
;   file_name: The name of the File
;
; OUTPUTS:
;   This function returns 1 if the file exist and 0 if not
;
; EXAMPLE:
;   result=file_exist('otto.nc')
;
; MODIFICATION HISTORY:
```

; Written by: R.Bauer (ICG-1), 1998-May-18

;-

FUNCTION file_exist,file_name

if n_params() lt 1 then begin

 help,call=call

 help_of_interest=within_brackets(call[0],brackets=['<','('])

 message,help_calling_sequence(help_of_interest),/cont

 return,-1

endif

 OPENR,lun,file_name,err=err,/GET_LUN

 IF n_elements(lun) GT 0 THEN FREE_LUN,lun

 IF err NE 0 THEN RETURN,0 ELSE RETURN,1

END

;

; Copyright (c) 1997, Forschungszentrum Juelich GmbH ICG-1

; All rights reserved.

; Unauthorized reproduction prohibited.

; This software may be used, copied, or redistributed as long as it is not

; sold and this copyright notice is reproduced on each copy made. This

; routine is provided as is without any express or implied warranties

; whatsoever.

;

;+

; NAME:

; fileline

;

; PURPOSE:

; This function returns the number of lines of an ascii file

;

; CATEGORY:

; DATAFILES/FILE

;

; CALLING SEQUENCE:

; Result=fileline(file_name)

;

; INPUTS:

; file_name: the name of an ascii file

```

;
; KEYWORD PARAMETERS:
; bytarr: optional output
;
; EXAMPLE:
; Result=fileline('test.asc')
;
; MODIFICATION HISTORY:
; Written by: R.Bauer (ICG-1), Oct. 1996
; R.Bauer 1998-11-10 added opt output bytarr
;-

```

FUNCTION fileline, filename, BYTARR=lesefeld

```

IF N_PARAMS() LT 1 THEN BEGIN
  HELP:HELP, call=call
  help_of_interest=within_brackets(call[0], brackets=['<', '('])
  MESSAGE, help_calling_sequence(help_of_interest), /cont
  RETURN, -1
  help_open: MESSAGE, 'File: '+filename+' NOT found', /cont
  RETURN, -1
  help_size: MESSAGE, 'File: '+filename+' has a SIZE OF 0 bytes', /cont
  RETURN, -1
ENDIF
byt=filesize(filename)
IF byt EQ 0 THEN GOTO, help_size

IF byt EQ -1 THEN GOTO, help_open

lesefeld=BYTARR(byt)

OPENR, lun, filename, /GET_LUN, error=err
IF err NE 0 THEN GOTO, help_open
READU, lun, lesefeld

FREE_LUN, lun
IF lesefeld(byt-1) NE 10b THEN lesefeld=[lesefeld, 10b]
line=WHERE(lesefeld EQ 10B, count_line)

RETURN, count_line

END

```

```

;
; Copyright (c) 1996, Forschungszentrum Juelich GmbH ICG-1
; All rights reserved.
; Unauthorized reproduction prohibited.
; This software may be used, copied, or redistributed as long as it is not
; sold and this copyright notice is reproduced on each copy made. This
; routine is provided as is without any express or implied warranties
; whatsoever.
;+
; NAME:
; filesize
;
; PURPOSE:
; The result of this function is the bytelength of an ascii file
;
; CATEGORY:
; DATAFILES/FILE
;
; CALLING SEQUENCE:
; Result=filesize(file_name)
;
; INPUTS:
; file_name: the name of an ascii file
;
; OUTPUTS:
; This function returns the number of bytes of an ascii file
;
; EXAMPLE:
; Result=filesize('test.asc')
;
; MODIFICATION HISTORY:
; Written by: R.Bauer (ICG-1), Oct. 1996
;-

```

FUNCTION filesize, filename

```

if n_params() lt 1 then begin
    help:help,call=call
    help_of_interest=within_brackets(call[0],brackets=['<','()])
    message,help_calling_sequence(help_of_interest),/cont
    return,-1
    help_open: message, 'File: '+filename+' not found',/cont
    return,-1
endif

```

```
OPENR, lun, filename, /GET_LUN,error=err
IF err NE 0 THEN GOTO, help_open
stats = FSTAT(lun)
FREE_LUN, lun
```

```
RETURN, stats.size
END
```

File Attachments

- 1) [file_exist.pro](#), downloaded 111 times
 - 2) [fileline.pro](#), downloaded 91 times
 - 3) [filesize.pro](#), downloaded 90 times
-

Subject: Re: reading ASCII Data
Posted by [Mirko](#) on Tue, 13 Apr 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Was in a hurry this mornig an missed to put an example on.

```
1 08.04.1999 15:27:12:40 11.04 1.05722E-007 2.14 9.9E-009 16.05
3.59375E-009 28.02 4.20938E-009 44.03 1.2875E-009
2 08.04.1999 15:27:21:60 20.6 1.07047E-007 2.19 1.04469E-008 16.05
3.66875E-009 17.06 1.89781E-008 18.09 6.69937E-008 28.02
4.11563E-009 31.94 1.49062E-009 43.97 1.35312E-009
3 08.04.1999 15:27:30:44 29.44 1.14278E-007 ... and so on for the next
700 ranges
```

Any suggestions?

Cheers
