Subject: Q: About reading files into an array without knowing the size.
Posted by David van Kuijk on Thu, 02 Nov 1995 08:00:00 GMT
View Forum Message <> Reply to Message

Hi

One of the nice things of IDL is that it is possible to read whole
ASCII-files of data (e.g. floats) into an array in one swoop, without
having to go through a while loop which reads all of these numbers one by
one. E.g.:

```
OPENR,1, "filename"
floatss=FLTARR(10000)
READF, 1, floatss
```

What is not so nice is that IDL has to know exactly how many datapoints
there are in the file, otherwise not all the data are read, or you get
sth like an "End of file encountered"-error. So the size of _floatss_ in
the example above should be equal to the number of floats in the file.

I know that Mathlab is capable of processing files with an unknown number
of data.

Does anybody know a way to achieve this in IDL (maybe I missed something)?

David,

```
 ************************************************************ ***********
* David van Kuijk       | Max-Planck-Institute for Psycholinguistics *
* E-mail: kuijk@mpi.nl   | Wundtlaan 1                              *
*                    | 6525 XD Nijmegen                         *
*                    | The Netherlands                          *
* Tel: +31 (0)24 3521523 |                                          *
* Fax: +31 (0)24 3521213 | Snail-mail: P.O. Box 310, 6500 AH Nijmegen *
*              "Prots the whoblem?"                              *
 ************************************************************ ***********
```

Subject: Re: Q: About reading files into an array without knowing the size.
Posted by rsmith on Thu, 02 Nov 1995 08:00:00 GMT
View Forum Message <> Reply to Message

David,

Yes, this is annoying.   I wrote a little routine a while back to deal
with just this problem, and posted it to this group. Today I added a few
new features, based on suggestions by Boyd Blackwell, such as
returning row and column numbers in keywords, and working on comma

separated data as well as tab or spaces separated data.

Anyway, the new features seem to work on my test files, and I've been using a slightly older version (no commas, basically) for a long time with no problems. Use at your own risk, of course, and feel free to redistribute as needed.

Randall Smith
rsmith@wisp5.physics.wisc.edu

```
 ------------------------------------------------------------ ------
PRO  read_array,filename,array,SKIP=skip,INTEGER=integer,ROWS=row s,$
             COLUMNS=cols,SILENT=silent
;+
; NAME:
; READ_ARRAY
;
; PURPOSE:
;      Reads a file into an array variable.  The file is assumed to
;      consist of lines of numbers, separated by tabs or spaces,
;      with the same number of values on each line.  The file length
;      is arbitrary.
;
; CATEGORY:
;      PROG
;
; CALLING SEQUENCE:
;      READ_ARRAY, filename, array
;
; INPUTS:
;      filename: The name of the file to be read.
;      array:    The variable to hold the data.
;
; KEYWORD PARAMETERS:
; SKIP:    The number of lines of "header" information in the
;          file to skip.
;
; INTEGER: If the data in the file is integer.  The default is
;          floating point.
;
;      ROWS:    Returns the number of rows
;
;      COLUMNS: Returns the number of columns
;
;      SILENT: If set, do not output to the screen
;
; OUTPUTS:
;      Returns a two-dimensional array whose first index is the
;       number of elements per line, and second index is the number of
;       lines in the file.  Also outputs these numbers to the screen.
;
; RESTRICTIONS:
;       Not tested all that much.  Does not read double precision data.
;
; EXAMPLE:
;       READ_ARRAY, 'spectra.dat', spectrum, SKIP=3, /SILENT
;       Reads the file spectra.dat, skipping the first 3 lines, creating
;       the array variable spectrum.  Don't print out anything.
;
; MODIFICATION HISTORY:
;  Written by: Randall Smith, 6/19/95
;       Modified  :     RKS,            11/2/95
;-
if (N_params() lt 2) then begin
    print,'Call with'
    print,'"Read_Array,"filename",array,[skip=n],[/integer]'
    print,'where "filename" is the file to be read'
    print,'      "array" is the variable to put the data into and'
    print,'      /skip=n where n is the number of lines at the beginning ' + $
      'to skip and '
    print,'      /integer is used if the data is integer, not float.'
    return
endif
;
; Check to see if file exists and open file
;
result = findfile(filename,count=ct)
if (ct eq 0) then begin
    print,'File : '+filename+' not found.'
    return
endif
if (ct gt 1) then begin
    print,'Multiple files match that name:'
    print,result
    return
endif
get_lun,lun
openr,lun,filename
;
; Skip any lines?
;
if (keyword_set(skip) ne 0) then begin
    line = 'string'
    for i=0,skip-1 do readf,lun,line
endif
;
```

```
; Calculate the number of elements per line
;
tab = string(9B)
space = ' '
comma = ','
first = 1
line = ' string '
readf,lun,line
pos = strpos(line,tab)
while (pos ne -1) do begin
   strput,line,space,pos       ; Convert tabs
   pos = strpos(line,tab)
endwhile
line = strtrim(line,2)  ; Remove extra spaces
line = line+' '    ; Guarantee at least one space found
while (strlen(line) gt 0) do begin
   pos = strpos(line,space)
   cpos = strpos(line,comma)
   if ((cpos ne -1) and (cpos le pos)) then pos = cpos
   if (pos ne -1) then begin
      if (keyword_set(integer)) then begin
         number = fix(strmid(line,0,pos))
      endif else begin
         number = float(strmid(line,0,pos))
      endelse
      if (first eq 1) then begin
         first = 0
         arrayline = number
      endif else begin
         arrayline = [arrayline,number]
      endelse
      while ((pos lt strlen(line)) and $
            ((strmid(line,pos,1) eq space) or $
             (strmid(line,pos,1) eq comma))) do begin
         pos = pos+1
      endwhile
      nline = strmid(line,pos,strlen(line))
      line = nline
   endif
   line = strtrim(line,1)      ; Get rid of excess white space
endwhile
array = arrayline
nperline = n_elements(arrayline)
if (keyword_set(silent) eq 0) then begin
   print,'Number of elements per line: ',strtrim(string(nperline),2)
endif
numline = 1
;
```

```
; Read the file
;
if (keyword_set(integer)) then begin
   a = intarr(nperline)
endif else begin
   a = fltarr(nperline)
endelse
while (not(eof(lun))) do begin
   readf,lun,a
   array = [[array],[a]]
   numline = numline + 1
endwhile
if (keyword_set(silent) eq 0) then begin
   print,'Number of lines in file: ',strtrim(string(numline),2)
endif
;
; Clean up
;
rows=fix(numline)
cols=fix(nperline)
free_lun,lun

return
end
```

Subject: Re: Q: About reading files into an array without knowing the size.
Posted by Pierre Maxted on Mon, 06 Nov 1995 08:00:00 GMT

View Forum Message <> Reply to Message

David van Kuijk <kuijk@mpi.nl> wrote:
> Hi
>
> One of the nice things of IDL is that it is possible to read whole
> ASCII-files of data (e.g. floats) into an array in one swoop, without
> having to go through a while loop which reads all of these numbers one by
> one. E.g.:
>
> OPENR,1, "filename"
> floatss=FLTARR(10000)
> READF, 1, floatss
>
> What is not so nice is that IDL has to know exactly how many datapoints
> there are in the file, otherwise not all the data are read, or you get
> sth like an "End of file encountered"-error. So the size of _floatss_ in
> the example above should be equal to the number of floats in the file.
>
> I know that Mathlab is capable of processing files with an unknown number

> of data.
>
> Does anybody know a way to achieve this in IDL (maybe I missed something)?
>
> David,
>
> ** ***********
> * David van Kuijk | Max-Planck-Institute for Psycholinguistics *
> * E-mail: kuijk@mpi.nl | Wundtlaan 1   *
> * | 6525 XD Nijmegen   *
> * | The Netherlands   *
> * Tel: +31 (0)24 3521523 | *
> * Fax: +31 (0)24 3521213 | Snail-mail: P.O. Box 310, 6500 AH Nijmegen *
> *   "Prots the whoblem?"   *
> ** ***********
>

This routine, due to Michael Andersen in Copenhagen, should do the trick if you
are on a UNIX machine. The trick here is to use the command "wc" to get the
number of lines and "words" in the file. I'm sure it can be converted to VMS
without to much trouble.

Included below are the function itself (READ_FLTARR) that reads in the array
and below that is the function ACCESS that checks that the file exists.
--

```
  _-_-_                                        _-_-_
 |     Dr Pierre Maxted (pflm@star.maps.susx.ac.uk)      |
| |       Astronomy Centre, University of Sussex        | |
 |           Falmer, Brighton, BN1 9QH                  |
  -_-_-         Procrastinate now!!!              -_-_-
```

```
FUNCTION READ_FLTARR, FILE , COLLS , LINES , SILENT = silent
;+
; NAME:
;       READ_FLTARR
;
; PURPOSE:
;       read a 2D float array from an ascii file
;
; CALLING SEQUENCE:
;       array = READ_FLTARR( FILE [ , COLLS , LINES , /SILENT ] )
;
; INPUTS:
;       FILE    String giving file from which to read array
;
```

```
; KEYWORDS;       SILENT  If set, the size of ARRAY will not be displayed
;
; OUTPUTS:
;       ARRAY   Float array
;
; OPTIONAL OUTPUTS
;       COLLS   Number of collumns in ARRAY
;
;       LINES   Number of lines in ARRAY
;
; RESTRICTIONS
;       files with header or empty lines cannot be handled
;       uses non standard routine 'ACCESS'
;
; PROCEDURE
;       Uses SPAWN and UNIX 'wc' to establish size of array
;
; MODIFICATION HISTORY:
;       WRITTEN, Michael Andersen CUOBS, August, 1994
;-

  ON_ERROR, 2

  IF NOT ( ACCESS( file ) ) THEN $
  MESSAGE, 'Cannot access file:' + file

  SPAWN, "wc -lw " + file + $
         " | sed s/" + file + "//" + $
         " | awk '"+'{printf("%s\n%s\n",$1,$2)}'+"'", lw
  lines = LONG( lw( 0 ) )
  colls = LONG( lw( 1 ) ) / lines
  IF ( lw( 0 ) eq 0 ) THEN MESSAGE, 'File with no lines:' + file

  array = fltarr( colls , lines )
  IF( NOT KEYWORD_SET( SILENT ) ) THEN BEGIN
    PRINT, 'Reading array of ' + STRTRIM( colls ) + ' columns'
    PRINT, '             and ' + STRTRIM( lines ) + ' lines'
  ENDIF

  OPENR, l , file , /get
  READF, l , array
  CLOSE, l & FREE_LUN, l

  RETURN, array

END
```

```
 ------------END OF READ_FLTARR--------------


FUNCTION ACCESS, file , UNIQ = uniq

;+
;NAME:
;       ACCESS
;
;PURPOSE:
;       Boolean function returning true, if file exist
;
;CATEGORY:
;       Files, IO
;
;CALLING SEQUENCE:
;       a = ACCESS( file [, UNIQUE = unique ] )
;
;INPUTS:
;       file    Scalar string, giving the name of the file to be searched fore
;
;OPTIONAL KEYWORDS
;       UNIQ,   if set, ACCESS returns true only if file is unique
;
;OUTPUTS:
       a       boolean, true if file exsist, else false
;
;COMMON BLOCKS:
;       none
;
;SIDE EFFECTS:
;       none
;
;RESTRICTIONS:
;       only a single file can be handled at a time
;
;MODIFICATION HISTORY:
;       Written, Michael Andersen, December 1992
;-


  ON_ERROR, 2                           ;return to caller

  IF( STRLEN( file ) EQ 0 ) THEN RETURN, 0
  a = FINDFILE( file , COUNT = nfiles )
  IF( nfiles EQ 1 OR nfiles GT 1 AND NOT KEYWORD_SET( UNIQUE ) ) THEN RETURN, 1
  RETURN, 0
```

END

Subject: Re: Q: About reading files into an array without knowing the size.
Posted by rivers on Mon, 06 Nov 1995 08:00:00 GMT
View Forum Message <> Reply to Message

In article <DHMELp.JDM@news.dlr.de>, Hermann Mannstein <H.Mannstein@dlr.de> writes:
> David van Kuijk <kuijk@mpi.nl> wrote:
>> Hi
>>
>> One of the nice things of IDL is that it is possible to read whole
>> ASCII-files of data (e.g. floats) into an array in one swoop, without
>> having to go through a while loop which reads all of these numbers one by
>> one. E.g.:
>>
>> OPENR,1, "filename"
>> floatss=FLTARR(10000)
>> READF, 1, floatss
>>
>> What is not so nice is that IDL has to know exactly how many datapoints
>> there are in the file, otherwise not all the data are read, or you get
>> sth like an "End of file encountered"-error. So the size of _floatss_ in
>> the example above should be equal to the number of floats in the
>
> with
> OPENR,1, "filename"
> a=fstat(1)
> floatss=fltarr(a.size/4)
> READF, 1, floatss
>
> you will get what you want, but you have to know, what type of data "filename"
> contains.
> --

This will not work. "a.size/4" would be the number of elements in the array in
a BINARY file, but the original question concerned ASCII files. In and ASCII
file the number of bytes/element is not known in advance, and it may not even
be the same for each array element.

Mark Rivers	(312) 702-2279 (office)
CARS	(312) 702-9951 (secretary)
Univ. of Chicago	(312) 702-5454 (FAX)
5640 S. Ellis Ave.	(708) 922-0499 (home)
Chicago, IL 60637	rivers@cars3.uchicago.edu (Internet)

Subject: Re: Q: About reading files into an array without knowing the size.
Posted by Hermann Mannstein on Mon, 06 Nov 1995 08:00:00 GMT
View Forum Message <> Reply to Message

David van Kuijk <kuijk@mpi.nl> wrote:
> Hi
>
> One of the nice things of IDL is that it is possible to read whole
> ASCII-files of data (e.g. floats) into an array in one swoop, without
> having to go through a while loop which reads all of these numbers one by
> one. E.g.:
>
> OPENR,1, "filename"
> floatss=FLTARR(10000)
> READF, 1, floatss
>
> What is not so nice is that IDL has to know exactly how many datapoints
> there are in the file, otherwise not all the data are read, or you get
> sth like an "End of file encountered"-error. So the size of _floatss_ in
> the example above should be equal to the number of floats in the

with
```
OPENR,1, "filename"
a=fstat(1)
floatss=fltarr(a.size/4)
READF, 1, floatss
```

you will get what you want, but you have to know, what type of data "filename"
contains.
--
Regards,

```
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~
 Hermann Mannstein           Tel.: +49 8153 28-2503
 Institut fuer Physik der Atmosphaere    or -2558
 DLR - Oberpfaffenhofen            Fax.: +49 8153 28-1841
 Postfach 1116        \       mailto:H.Mannstein@dlr.de
 D-82230 Wessling        \ 0    http://www.op.dlr.de/~pa64
 Germany    ________\/|________
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~`--------\--------'~ ~~~~~~~~~~~~~~~~~~~~
                                         \
```

Subject: Re: Q: About reading files into an array without knowing the size.
Posted by meron on Tue, 07 Nov 1995 08:00:00 GMT
View Forum Message <> Reply to Message

In article <309FD207.167E@pfc.mit.edu>, "Thomas W. Fredian" <twf@pfc.mit.edu> writes:

```
> David van Kuijk wrote:
>>
>>  Hi
>>
>>  One of the nice things of IDL is that it is possible to read whole
>>  ASCII-files of data (e.g. floats) into an array in one swoop, without
>>  having to go through a while loop which reads all of these numbers one by
>>  one. E.g.:
>>
>>  OPENR,1, "filename"
>>  floatss=FLTARR(10000)
>>  READF, 1, floatss
>>
> ...
>
> I just tried to write a simple procedure with an on_ioerror to trap the
> end of file error and it seems to work fairly well:
>
> function readarray,unit
>  f = fstat(unit)
>  maxvals=f.size
>  answer=fltarr(maxvals) ; Allocate an array guaranteed to be big enough
>  on_ioerror,finish      ; Trap end of file error
>  readf,unit,answer      ; Read in what values are there
>  finish:
>  f = fstat(unit)
>  return,answer(0:f.transfer_count-1) ; return the truncated array
> end
>
> I tested this with IDL Version 4.0.1(vms alpha) and it seems to work.
> --
```

Years ago I wrote a routine called read_ascii which reads data from an
ASCII file. The routine finds on the go the number of rows and
columns, as long as the number of entries on each line is the same.
It is possible to have lines of text imbedded in the file, the routine
will ignore them. Finally, it is called as a function so it returns
the data as an array.

Since the routine calls on other routines from my library, it is
advisable to copy the whole library and sort things out later. The
library, in tar form is accessible on cars.uchicago.edu in the pub/idl
directory. The file name is cars_library.tar . You can log in as
anonymous and get it.

Mati Meron | "When you argue with a fool,
meron@cars3.uchicago.edu | chances are he is doing just the same"

Subject: Re: Q: About reading files into an array without knowing the size.
Posted by mirko.vukovic on Tue, 07 Nov 1995 08:00:00 GMT
View Forum Message <> Reply to Message

In article <DHMELp.JDM@news.dlr.de>, H.Mannstein@dlr.de says...
>
> David van Kuijk <kuijk@mpi.nl> wrote:

> with
> OPENR,1, "filename"
> a=fstat(1)
> floatss=fltarr(a.size/4)
> READF, 1, floatss
>
> you will get what you want, but you have to know, what type of data
"filename"
> contains.
> --
> Regards,
>
> ~~~~~~
~~~~~~~~~~~~~
~~~~~~~~~
>   Hermann Mannstein   Tel.: +49 8153
Well, I tried that and it does not quite work, since fstat returns the
file size in bytes. If you are reading ascii, the division by 4 (4 bytes
for floating point variables) is not correct. How many bytes/character
(0.5?)
Now, if fstat returned the number of records in a file, that would be
something usefull, provided one knew how many columns are in the file.

--
Mirko Vukovic, PhD   mirko.vukovic@grc.varian.com
Varian Research Center   Phone: (415) 424-4969
3075 Hansen Way, M/S K-109   Fax: (415) 424-6988
Palo Alto, CA 94304-1025

Subject: Re: Q: About reading files into an array without knowing the size.
Posted by Thomas W. Fredian on Tue, 07 Nov 1995 08:00:00 GMT
View Forum Message <> Reply to Message

David van Kuijk wrote:
>
> Hi
>
> One of the nice things of IDL is that it is possible to read whole
> ASCII-files of data (e.g. floats) into an array in one swoop, without

```
>  having to go through a while loop which reads all of these numbers one by
>  one. E.g.:
>
>  OPENR,1, "filename"
>  floatss=FLTARR(10000)
>  READF, 1, floatss
>
...
```

I just tried to write a simple procedure with an on_ioerror to trap the
end of file error and it seems to work fairly well:

```
function readarray,unit
 f = fstat(unit)
 maxvals=f.size
 answer=fltarr(maxvals) ; Allocate an array guaranteed to be big enough
 on_ioerror,finish      ; Trap end of file error
 readf,unit,answer      ; Read in what values are there
 finish:
 f = fstat(unit)
 return,answer(0:f.transfer_count-1) ; return the truncated array
end
```

I tested this with IDL Version 4.0.1(vms alpha) and it seems to work.
--
/*************************************/
Thomas W. Fredian
Plasma Fusion Center
Massachusetts Institute of Technology
email: twf@pfc.mit.edu
/*************************************/

Subject: Re: Q: About reading files into an array without knowing the size.
Posted by chris on Sat, 11 Nov 1995 08:00:00 GMT

View Forum Message <> Reply to Message

I often spawn the unix wc (word count) program to asses the
length of an ascii file. In fact I (and probably others as well)
have written a simple routine which employs "getwrd.pro" from
the JHUALP library:

```
function n_lines,file

;this will tell you how many lines a text file is
; Usage: N = n_lines("myfile.dat")

spawn,'wc '+ file, output       ;unix word count command
```

```
return,getwrd(output(0),0)
end
```