Subject: Re: Help with reading structure from file
Posted by btt on Wed, 07 Feb 2001 15:01:33 GMT
View Forum Message <> Reply to Message

Hello,

I haven't used PV-Wave in a very long time but I do recall the very nice DC_Read* functions (I might not have the syntax quite right) that were great for reading column-oriented flat files. Have you tried those?

Ben

Surendar Jeyadev wrote:

> In the loosing battle with the Excel users, the latest direct hit was reading
> in a file that contained strings and numeric data in each line. Having given
> up (and as the only hold out against Excel!), I need HEEELLLPPP.
>
> This is the simplified problem. I am trying to read data in the following
> format:
>
> 001a 312.194 76.922 296.301 21.462 0.453 289.515 0.957
> 001b 363.748 106.090 506.188 19.430 0.528 347.252 1.176
> 001c 398.248 138.541 724.470 17.152 0.578 383.534 1.701
> 002a 294.593 28.525 248.744 8.532 0.428 290.497 1.268
> 002b 353.415 46.290 449.015 7.974 0.513 349.565 2.011
> 002c 401.279 80.260 661.701 3.341 0.582 395.403 4.529
> ....
> ....
>
> i.e. in the format "(4x,a4,7f9.3)". I would like it to go into a 2 dimensional
> structure.
>
> I cannot find a way of reading it as a entire array. At present, all I can
> come up with is
>
> nsets = 108 ; number of lines of data
> f1 = "(4x,a4,7f9.3)"
> a = string(4)
> y = fltarr(7)
> dpt = { fullrow, id: ' ', x: fltarr(7) }
> fulldata = replicate({fullrow}, nsets)
>
> openr, 1, 'data'
> for i=0,nsets-1 do begin
> readf, 1, format = f1, a, y
> fulldata(i).id = a
> fulldata(i).x = y

> endfor
> close, 1
>
> Is there any way of avoiding the temporary variables and the loop? I am
> using PV-Wave CL, Ver 6.
> --
>
> Surendar Jeyadev jeyadev@wrc.xerox.com

--
Ben Tupper
Bigelow Laboratory for Ocean Sciences
180 McKown Point Rd.
W. Boothbay Harbor, ME 04575
btupper@bigelow.org

Subject: Re: Help with reading structure from file
Posted by Pavel A. Romashkin on Wed, 07 Feb 2001 17:11:32 GMT
View Forum Message <> Reply to Message

Sure does work, Martin :-)

```
;**************************
pro test
buffer = {a : '', b: fltarr(7)}
result = replicate(buffer, 6)
temp = dialog_pickfile()
; Pick the file with Surendar's data
openr, unit, temp, /get_lun
readf, unit, result, format='(4x,a4,7f9.3)'
free_lun, unit
; No need in subscripting
help, result.(1)
end
;**************************
```

Cheers,
Pavel

Martin Schultz wrote:
>
> The solution is to use a structure. SOmething like this (no time for
> testing though):
>
> template = { a:'', x:fltarr(7) }
> result = replicate(template, NLINES)
> readf, lun, result, format='(4x,a4,7f9.3)'

>
>   data = result[*].x   ; not sure if this even works ;-(
>
> Hope this helps you a little bit,
>
> Martin

Subject: Re: Help with reading structure from file
Posted by jeyadev on Wed, 07 Feb 2001 17:24:22 GMT
View Forum Message <> Reply to Message

In article <3A810F7C.B69C6E28@dkrz.de>,
Martin Schultz <martin.schultz@dkrz.de> wrote:
> The solution is to use a structure. SOmething like this (no time for
> testing though):
>
>   template = { a:'', x:fltarr(7) }
>   result = replicate(template, NLINES)

This is exactly what I tried -- see the code fragment included below.

>   readf, lun, result, format='(4x,a4,7f9.3)'
>   data = result[*].x   ; not sure if this even works ;-(

No, it does not! Here is the sesssion from PV Wave:

```
WAVE> nsets = 108
WAVE> f1 = "(4x,a4,7f9.3)"
WAVE> dpt = { fullrow, id: ' ', x: fltarr(7) }
WAVE> fulldata = replicate( {fullrow}, nsets)
WAVE>
WAVE> openr, 1, 'data'
WAVE> readf, 1, fulldata, format = f1
% End of input record encountered on file unit: 1.
% Execution halted at $MAIN$   (READF).
WAVE>
```

The reason is the the format statement is being ignored. The entire
first line of data is being read as the string variable, the first
7 fields of the *second* line are read as x, the last column of the
second line is read as the string variable of the second element of
the structure array ..... need I say more! This is what should happen
is the READF did *not* have a format statement.

Curiously, the PV Wave manual has a very similar example with an
unformatted read. What they do is

```
   fulldata = replicate( {fullrow, string(' ', format=f1),  $
              fltarr(7) } , nsets)
   readu, lun, fulldata
```

I tried this with the formatted read and it did not work. I guess
the Excel fans have won another round :-(To avoid this, I normally
put the string labels at the end of the data rows, but this was a
"How will you do it fast?" challenge.

> Hope this helps you a little bit,
>
> Martin

Thanks for the suggestion ...

Dave (or anyone else!), Any suggestion why the format is being ignored?

> Surendar Jeyadev wrote:
>>
>> In the loosing battle with the Excel users, the latest direct
>>   hit was reading
>> in a file that contained strings and numeric data in each line.
>>   Having given
>> up (and as the only hold out against Excel!), I need HEEELLLPPP.
>>
>> This is the simplified problem. I am trying to read data in the following
>> format:
>>
>>   001a 312.194   76.922 296.301   21.462   0.453 289.515   0.957
>>   001b 363.748 106.090 506.188   19.430   0.528 347.252   1.176
>>   001c 398.248 138.541 724.470   17.152   0.578 383.534   1.701
>>   002a 294.593   28.525 248.744   8.532   0.428 290.497   1.268
>>   002b 353.415   46.290 449.015   7.974   0.513 349.565   2.011
>>   002c 401.279   80.260 661.701   3.341   0.582 395.403   4.529
>>   ....
>>   ....
>>
>> i.e. in the format "(4x,a4,7f9.3)". I would like it to go into a 2 dimensional
>> structure.
>>
>> I cannot find a way of reading it as a entire array. At present, all I can
>> come up with is
>>
>>   nsets = 108   ; number of lines of data
>>   f1 = "(4x,a4,7f9.3)"
>>   a = string(4)
>>   y = fltarr(7)
>>   dpt = { fullrow, id: ' ', x: fltarr(7) }

```
>>       fulldata = replicate( {fullrow}, nsets)
>>
>>       openr, 1, 'data'
>>       for i=0,nsets-1 do begin
>>          readf, 1, format = f1, a, y
>>          fulldata(i).id = a
>>          fulldata(i).x = y
>>       endfor
>>       close, 1
>>
>>  Is there any way of avoiding the temporary variables and the loop? I am
>>  using PV-Wave CL, Ver 6.
>>  --
>>
>>  Surendar Jeyadev        jeyadev@wrc.xerox.com
>
> --
>  [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[ [[[[[[[
> [[ Dr. Martin Schultz   Max-Planck-Institut fuer Meteorologie    [[
> [[                      Bundesstr. 55, 20146 Hamburg             [[
> [[                      phone: +49 40 41173-308                  [[
> [[                      fax:   +49 40 41173-298                  [[
> [[ martin.schultz@dkrz.de                                        [[
>  [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[ [[[[[[[
```

--

Surendar Jeyadev   jeyadev@wrc.xerox.com

Subject: Re: Help with reading structure from file
Posted by jeyadev on Wed, 07 Feb 2001 17:40:18 GMT

View Forum Message <> Reply to Message

In article <95s0c6$8ar$1@news.wrc.xerox.com>,
Surendar Jeyadev <jeyadev@wrc.xerox.com> wrote:
> In article <3A810F7C.B69C6E28@dkrz.de>,
> Martin Schultz <martin.schultz@dkrz.de> wrote:

```
>> The solution is to use a structure. SOmething like this (no time for
>> testing though):
>>
>>    template = { a:'', x:fltarr(7) }
>>    result = replicate(template, NLINES)
>
> This is exactly what I tried -- see the code fragment included below.
>
>>    readf, lun, result, format='(4x,a4,7f9.3)'
```

>>   data = result[*].x   ; not sure if this even works ;-(
>
> No, it does not! Here is the sesssion from PV Wave:
>
>   WAVE> nsets = 108
>   WAVE> f1 = "(4x,a4,7f9.3)"
>   WAVE> dpt = { fullrow, id: ' ', x: fltarr(7) }
>   WAVE> fulldata = replicate({fullrow}, nsets)
>   WAVE>
>   WAVE> openr, 1, 'data'
>   WAVE> readf, 1, fulldata, format = f1
>   % End of input record encountered on file unit: 1.
>   % Execution halted at $MAIN$ (READF).
>   WAVE>
>
> The reason is the the format statement is being ignored. The entire
> first line of data is being read as the string variable, the first
> 7 fields of the *second* line are read as x, the last column of the
> second line is read as the string variable of the second element of
> the structure array ..... need I say more! This is what should happen
> is the READF did *not* have a format statement.

Sorry, that is not correct. I tried so many variations and got so
many different errors, I am all mixed up now. The problem is a really
weird one, now and I cannot figure it out.

After doing the above, ...

```
WAVE> print, fulldata(0)
{ 001a      312.194      76.9220      296.301      21.4620
    0.453000      289.515    0.957000 }
WAVE> print, fulldata(1)
{ 7         0.00000      0.00000      0.00000      0.00000
     0.00000      0.00000    0.00000 }
{       0.00000      0.00000      0.00000      0.00000
     0.00000      0.00000     0.00000 }
```

What is going on here? The first line of data is read correctly.
Why are the second, third ..... data rows messed up? On the other
hand, line by line reading via a loop works just fine.

> Curiously, the PV Wave manual has a very similar example with an
> unformatted read. What they do is
>
>   fulldata = replicate({fullrow, string(' ', format=f1), $
>   fltarr(7) } , nsets)
>   readu, lun, fulldata
>

> I tried this with the formatted read and it did not work. I guess
> the Excel fans have won another round :-(To avoid this, I normally
> put the string labels at the end of the data rows, but this was a
> "How will you do it fast?" challenge.
>
>> Hope this helps you a little bit,
>>
>> Martin
>
> Thanks for the suggestion ...
>
> Dave (or anyone else!), Any suggestion why the format is being ignored?
>
>> Surendar Jeyadev wrote:
>>>
>>> In the loosing battle with the Excel users, the latest direct
>>>   hit was reading
>>> in a file that contained strings and numeric data in each line.
>>>   Having given
>>> up (and as the only hold out against Excel!), I need HEEELLLPPP.
>>>
>>> This is the simplified problem. I am trying to read data in the following
>>> format:
>>>
>>>   001a 312.194   76.922 296.301   21.462   0.453 289.515   0.957
>>>   001b 363.748 106.090 506.188   19.430   0.528 347.252   1.176
>>>   001c 398.248 138.541 724.470   17.152   0.578 383.534   1.701
>>>   002a 294.593   28.525 248.744   8.532   0.428 290.497   1.268
>>>   002b 353.415   46.290 449.015   7.974   0.513 349.565   2.011
>>>   002c 401.279   80.260 661.701   3.341   0.582 395.403   4.529
>>>   ....
>>>   ....
>>>
>>> i.e. in the format "(4x,a4,7f9.3)". I would like it to go into a 2 dimensional
>>> structure.
>>>
>>> I cannot find a way of reading it as a entire array. At present, all I can
>>> come up with is
>>>
>>>   nsets = 108   ; number of lines of data
>>>   f1 = "(4x,a4,7f9.3)"
>>>   a = string(4)
>>>   y = fltarr(7)
>>>   dpt = { fullrow, id: ' ', x: fltarr(7) }
>>>   fulldata = replicate({fullrow}, nsets)
>>>
>>>   openr, 1, 'data'
>>>   for i=0,nsets-1 do begin

>>>   readf, 1, format = f1, a, y
>>>   fulldata(i).id = a
>>>   fulldata(i).x = y
>>>   endfor
>>>   close, 1
>>>
>>> Is there any way of avoiding the temporary variables and the loop? I am
>>> using PV-Wave CL, Ver 6.
>>> --
>>>
>>> Surendar Jeyadev   jeyadev@wrc.xerox.com
>>
>> --
>> [[[[[[[[[
>> [[Dr. Martin Schultz   Max-Planck-Institut fuer Meteorologie   [[
>> [[Bundesstr. 55, 20146 Hamburg   [[
>> [[phone: +49 40 41173-308   [[
>> [[fax: +49 40 41173-298   [[
>> [[martin.schultz@dkrz.de   [[
>> [[[[[[[[[
>
>
> --
>
> Surendar Jeyadev   jeyadev@wrc.xerox.com

--

Surendar Jeyadev   jeyadev@wrc.xerox.com

Subject: Re: Help with reading structure from file
Posted by jeyadev on Wed, 07 Feb 2001 17:43:26 GMT
View Forum Message <> Reply to Message

In article <95s1a2$8f8$1@news.wrc.xerox.com>,
Surendar Jeyadev <jeyadev@wrc.xerox.com> wrote:
>
> Sorry, that is not correct. I tried so many variations and got so
> many different errors, I am all mixed up now. The problem is a really
> weird one, now and I cannot figure it out.
>
> After doing the above, ...
>
> WAVE> print, fulldata(0)
> { 001a   312.194   76.9220   296.301   21.4620
>   0.453000   289.515   0.957000 }

> WAVE> print, fulldata(1)
> { 7 0.00000 0.00000 0.00000 0.00000
> 0.00000 0.00000 0.00000 }

EEEEks! Now my Unix is going ....

> { 0.00000 0.00000 0.00000 0.00000
> 0.00000 0.00000 0.00000 }

It should have been ..

WAVE> print, fulldata(2)
{ 0.00000 0.00000 0.00000 0.00000
0.00000 0.00000 0.00000 }

Anyway, the point is the same.

Thanks again.
--

Surendar Jeyadev jeyadev@wrc.xerox.com

Subject: Re: Help with reading structure from file
Posted by Liam E. Gumley on Thu, 08 Feb 2001 16:11:18 GMT
View Forum Message <> Reply to Message

Surendar Jeyadev wrote:
> In the loosing battle with the Excel users, the latest direct hit was reading
> in a file that contained strings and numeric data in each line. Having given
> up (and as the only hold out against Excel!), I need HEEELLLPPP.
>
> This is the simplified problem. I am trying to read data in the following
> format:
>
> 001a 312.194 76.922 296.301 21.462 0.453 289.515 0.957
> 001b 363.748 106.090 506.188 19.430 0.528 347.252 1.176
> 001c 398.248 138.541 724.470 17.152 0.578 383.534 1.701
> 002a 294.593 28.525 248.744 8.532 0.428 290.497 1.268
> 002b 353.415 46.290 449.015 7.974 0.513 349.565 2.011
> 002c 401.279 80.260 661.701 3.341 0.582 395.403 4.529
> ....
> ....
>
> i.e. in the format "(4x,a4,7f9.3)". I would like it to go into a 2 dimensional
> structure.
>
> I cannot find a way of reading it as a entire array. At present, all I can

> come up with is
>
> nsets = 108 ; number of lines of data
> f1 = "(4x,a4,7f9.3)"
> a = string(4)
> y = fltarr(7)
> dpt = { fullrow, id: ' ', x: fltarr(7) }
> fulldata = replicate({fullrow}, nsets)
>
> openr, 1, 'data'
> for i=0,nsets-1 do begin
> readf, 1, format = f1, a, y
> fulldata(i).id = a
> fulldata(i).x = y
> endfor
> close, 1
>
> Is there any way of avoiding the temporary variables and the loop? I am
> using PV-Wave CL, Ver 6.

Before I begin, I must say that I have no experience with PV-Wave. My comments are based on my experience with IDL.

A loop is not necessarily a bad thing when dealing with I/O. To read data from ASCII text file as an array you need to know in advance how many lines are in the file. I usually try and avoid this problem, because you end up reading the whole file anyway just to count the number of lines.

Here's an example program I adapted in a couple of minutes from one I had laying around. All I modified was the format string (FMT) and the definition of a single record (RECORD). This function is able to skip bad input records, but of course I never have bad records in my data ;-)

Here is the test data file (note there are five spaces at the beginning of each line):

```
    001a  312.194   76.922  296.301   21.462    0.453  289.515    0.957
    001b  363.748  106.090  506.188   19.430    0.528  347.252    1.176
    001c  398.248  138.541  724.470   17.152    0.578  383.534    1.701
    002a  294.593   28.525  248.744    8.532    0.428  290.497    1.268
    002b  353.415   46.290  449.015    7.974    0.513  349.565    2.011
    002c  401.279   80.260  661.701    3.341    0.582  395.403    4.529
```

Here is the function:

FUNCTION READ_FILE, FILE, MAXREC=MAXREC

```
;- Check arguments
if (n_elements(file) eq 0) then message, 'Argument FILE is undefined'
if (n_elements(maxrec) eq 0) then maxrec = 10000L

;- Open input file
openr, lun, file, /get_lun

;- Define record format and structure, and create data array
fmt = '(5x, a4, 6f9.3)'
record = {site:'', values:fltarr(6)}
data = replicate(record, maxrec)

;- Read records until end-of-file reached
nrecords = 0L
recnum = 1L
while (eof(lun) ne 1) do begin

  ;- Read this record (jumps to bad_rec: on error)
  on_ioerror, bad_rec
  error = 1
  readf, lun, record, format=fmt
  error = 0

  ;- Store data for this record
  data[nrecords] = record
  nrecords = nrecords + 1L

;- Check if maximum record count exceeded
  if (nrecords eq maxrec) then begin
    free_lun, lun
    message, 'Maximum record count reached: increase MAXREC'
  endif

  ;- Check for bad input record
  bad_rec:
  if (error eq 1) then print, 'Bad data at record ', recnum
  recnum = recnum + 1

endwhile

;- Close input file
free_lun, lun

;- Trim data array and return it to caller
data = data[0 : nrecords - 1]
return, data

END
```

Here's how it works:

```
IDL> data = read_file('test.dat')
IDL> help, data
DATA            STRUCT    = -> <Anonymous> Array[6]
IDL> print, data.site
001a 001b 001c 002a 002b 002c
IDL> print, data.values
      312.194      76.9220      296.301      21.4620     0.453000
289.515
      363.748      106.090      506.188      19.4300     0.528000
347.252
      398.248      138.541      724.470      17.1520     0.578000
383.534
      294.593      28.5250      248.744      8.53200     0.428000
290.497
      353.415      46.2900      449.015      7.97400     0.513000
349.565
      401.279      80.2600      661.701      3.34100     0.582000
395.403
```

If there is a bad record in the input file, such as

```
    001a  312.194   76.922  296.301   21.462    0.453  289.515    0.957
    001b  363.748  106.090  506.188   19.430    0.528  347.252    1.176
    001c  398.248  138.541  724.470   17.152    0.578  383.534    1.701
    002a  294.593   28.525  248.744    8.532    0.428  290.497    1.268
    xxxx
    002b  353.415   46.290  449.015    7.974    0.513  349.565    2.011
    002c  401.279   80.260  661.701    3.341    0.582  395.403    4.529
```

you would see this

```
IDL> data = read_file('test.dat')
Bad data at record            5
IDL> help, data
DATA            STRUCT    = -> <Anonymous> Array[6]
```

Cheers,
Liam.
http://cimss.ssec.wisc.edu/~gumley

Subject: Re: Help with reading structure from file
Posted by jeyadev on Thu, 08 Feb 2001 17:42:28 GMT
View Forum Message <> Reply to Message

In article <95q2fs$m1c$1@news.wrc.xerox.com>,
Surendar Jeyadev <jeyadev@wrc.xerox.com> wrote:
> In the loosing battle with the Excel users, the latest direct hit was reading
> in a file that contained strings and numeric data in each line. Having given
> up (and as the only hold out against Excel!), I need HEEELLLPPP.
>
> This is the simplified problem. I am trying to read data in the following
> format:
>
>   001a 312.194   76.922 296.301   21.462   0.453 289.515   0.957
>   001b 363.748 106.090 506.188   19.430   0.528 347.252   1.176
>   001c 398.248 138.541 724.470   17.152   0.578 383.534   1.701
>   002a 294.593   28.525 248.744   8.532   0.428 290.497   1.268
>   002b 353.415   46.290 449.015   7.974   0.513 349.565   2.011
>   002c 401.279   80.260 661.701   3.341   0.582 395.403   4.529
>   ....
>   ....
>
> i.e. in the format "(4x,a4,7f9.3)". I would like it to go into a 2 dimensional
> structure.
>
> I cannot find a way of reading it as a entire array. At present, all I can
> come up with is
>
>
>   nsets = 108   ; number of lines of data
>   f1 = "(4x,a4,7f9.3)"
>   a = string(4)
>   y = fltarr(7)
>   dpt = { fullrow, id: ' ', x: fltarr(7) }
>   fulldata = replicate({fullrow}, nsets)
>
>   openr, 1, 'data'
>   for i=0,nsets-1 do begin
>   readf, 1, format = f1, a, y
>   fulldata(i).id = a
>   fulldata(i).x = y
>   endfor
>   close, 1
>
> Is there any way of avoiding the temporary variables and the loop? I am
> using PV-Wave CL, Ver 6.

Thanks to everyone for all the help. Just got a call from PV Wave support.
Looks like it *is* a Wave thing -- the man assures me that there is no
other way to do it. Now, one would expect them to document why that should
be so, wouldn't one? I asked if this was true for any kind of mixed record
but the man was not sure -- thought that it may be something limited to

strings. So much for that.

Thanks again.
--

Surendar Jeyadev jeyadev@wrc.xerox.com

Subject: Re: Help with reading structure from file
Posted by R.Bauer on Sun, 18 Feb 2001 22:04:53 GMT
View Forum Message <> Reply to Message

Surendar Jeyadev wrote:
>
> In the loosing battle with the Excel users, the latest direct hit was reading
> in a file that contained strings and numeric data in each line. Having given
> up (and as the only hold out against Excel!), I need HEEELLLPPP.
>
> This is the simplified problem. I am trying to read data in the following
> format:
>
> 001a 312.194 76.922 296.301 21.462 0.453 289.515 0.957
> 001b 363.748 106.090 506.188 19.430 0.528 347.252 1.176
> 001c 398.248 138.541 724.470 17.152 0.578 383.534 1.701
> 002a 294.593 28.525 248.744 8.532 0.428 290.497 1.268
> 002b 353.415 46.290 449.015 7.974 0.513 349.565 2.011
> 002c 401.279 80.260 661.701 3.341 0.582 395.403 4.529
> ....
> ....
>
Dear Surendar,

this is my solution of this problem.

You will need additional:

http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_source/idl_ html/dbase/download/fileline.tar.gz
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_source/idl_ html/dbase/download/n_separator.tar.gz

Please have a look for further routines and licensing at
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml

FUNCTION read_excel

  n_lines=fileline('excel.txt')

```
  txt=STRARR(n_lines)
  OPENR,lun,'excel.txt' ,/GET_LUN
  READF,lun,txt
  FREE_LUN,lun

  txt=STRCOMPRESS(txt)

  sep=' '
  n_sep=(n_separator(txt[1],sep))[0]
  pos_sep=STRPOS(txt[1],sep)

  asc=STRMID(txt,0,pos_sep)
  txt=STRMID(txt,pos_sep+1,10000)

  data=FLTARR(n_sep,n_lines)
  READS, txt, data

  RETURN,{asc:asc,data:data}

END
```

--
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 Jï¿½lich
 http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml

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