

---

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 HEEELLLLPPP.
>>>
>>> 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