
Subject: Re: Suppress return with readf ?

Posted by [davidf](#) on Wed, 04 Nov 1998 08:00:00 GMT

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

John Kwiatkowski (johnmk@shell.clark.net) writes:

```
> I have an ASCII data set that is giving me problems. I need to suppress
> the return when using readf since I only want to read in part of the
> line. Depending on the value of one of the fields there may be more
> info to read on that line. Anyone know how to stop readf from moving
> to the next line?
>
> Here is an example of 2 lines in the data file
> 0 58 2 94 2 0 0 0 0
> 0 57 7 85 44 0 0 0 2 0 0 0 2 0 0 0
>
> The first line has the 9 fields that are on every line. Depending on the
> value of the 9th field there may be 7 more fields to read on that line.
>
> psuedo code:
> A ; structure with required 9 fields
> B ; structure with optional 7 fields
>
> readf, unit, A
>   if (A.field9 neq 0) then
>     readf, unit B
>
> After the first read the file pointer gets placed at the next line in
> the file. How do I suppress this?
```

Ooohh, good question, John! I'm gonna come out of retirement to have a go at this one. :-)

Suppose you have a data set that looks like this:

```
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15
```

You want to read the first five values on the line, and if the last value read is "5", read 5 more values.

You certainly can't do it like this:

```
firstfive = IntArr(5)
nextfive = IntArr(5)
OpenR, lun, 'example.dat', /Get_Lun
ReadF, lun, firstfive
IF firstfive[4] EQ 5 THEN ReadF, lun, nextfive
```

Print, firstfive, nextfive

Because, as you have noticed the file pointer starts on the next line when you do the second read. A FORMAT keyword doesn't help either, I'm afraid (although I *always* get in trouble when I open my mouth about format statements.)

What you have to do instead is read the whole line into a STRING variable, and then read the actual values out of that string with the READS command. Your code might look like this:

```
firstfive = IntArr(5)
nextfive = IntArr(5)
line = "
OpenR, lun, 'example.dat', /Get_Lun
ReadF, lun, line
ReadS, line, firstfive
IF firstfive[4] EQ 5 THEN ReadS, line, firstfive, nextfive
Print, firstfive, nextfive
```

Ah, it feels good to be back in the saddle again. :-)

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438, Toll-Free Book Orders: 1-888-461-0155
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Note: A copy of this article was e-mailed to the original poster.
