Subject: Re: Text output line too short, width = val not working
Posted by laura.hike on Thu, 04 Jun 2015 20:12:30 GMT
View Forum Message <> Reply to Message

Drat, you're right. How embarrassing! I should've tried counting before messing with all of the individual formats. I knew about that behavior.

In the meantime, I did some experiments with output from IDL 8.4.1 and found some interesting results. I'll post them below. The 80 character limit only seems to apply in some cases....

Thanks!

Laura

On Thursday, June 4, 2015 at 12:46:46 PM UTC-7, Dick Jackson wrote:
> Hi Laura,
>
> It looks to me like you have more data in 'outdata' than you have formatting elements for, and the Print statement is reusing items from the last available *repeated* segment, being the "2(1X, F7.2)". This might look awful, but here goes:
>
> ; This is your original
> IDL> print, format = '(I4, 1X, 2(I3, 1X), I4, 1X, (A4, 1X), 2(F7.2, 1X), F7.2, 1X, 2(I4, 1X), F8.3, 1X, F8.3, 1X, F6.1, 1X, F6.1, 1X, F7.2, 1X, F7.2, 1X, F6.1, 2(1X, F7.2))', findgen(21)
> 0 1 2 3 5.00 6.00 7.00 8 9 10.000 11.000 12.0 13.0 14.00 15.00 16.0 17.00 18.00
> 19.00 20.00
>
> ; This is the same, but expanding the 2(1X, F7.2)... note how it's different now!
> IDL> print, format = '(I4, 1X, 2(I3, 1X), I4, 1X, (A4, 1X), 2(F7.2, 1X), F7.2, 1X, 2(I4, 1X), F8.3, 1X, F8.3, 1X, F6.1, 1X, F6.1, 1X, F7.2, 1X, F7.2, 1X, F6.1, 1X, F7.2, 1X, F7.2)', findgen(21)
> 0 1 2 3 5.00 6.00 7.00 8 9 10.000 11.000 12.0 13.0 14.00 15.00 16.0 17.00 18.00
> 19 20
>
> ; Giving more data:
> IDL> print, format = '(I4, 1X, 2(I3, 1X), I4, 1X, (A4, 1X), 2(F7.2, 1X), F7.2, 1X, 2(I4, 1X), F8.3, 1X, F8.3, 1X, F6.1, 1X, F6.1, 1X, F7.2, 1X, F7.2, 1X, F6.1, 1X, F7.2, 1X, F7.2, 1X, F7.2, 1X, F7.2)', findgen(21)
> 0 1 2 3 5.00 6.00 7.00 8 9 10.000 11.000 12.0 13.0 14.00 15.00 16.0 17.00 18.00 19.00 20.00
>
> ; Or, even better, perhaps:
> IDL> print, format = '(I4, 1X, 2(I3, 1X), I4, 1X, (A4, 1X), 2(F7.2, 1X), F7.2, 1X, 2(I4, 1X), F8.3, 1X, F8.3, 1X, F6.1, 1X, F6.1, 1X, F7.2, 1X, F7.2, 1X, F6.1, 4(1X, F7.2))', findgen(21)
> 0 1 2 3 5.00 6.00 7.00 8 9 10.000 11.000 12.0 13.0 14.00 15.00 16.0 17.00 18.00 19.00 20.00

>
> Hope this is helpful!
>
> This behaviour of IDL (going back to reuse only the last repeated segment) can be confusing, but likely helpful in many cases. In IDL Help on "Using Formatted Input and Output", it reads:
>
> Format Reversion
>
> In format reversion, the current record is terminated, a new one is initiated, and format control reverts to the group repeat specification whose opening parenthesis matches the next-to-last closing parenthesis of the format string. If the format does not contain a group repeat specification, format control returns to the initial opening parenthesis of the format string.
>
> Cheers,
> -Dick
>
> Dick Jackson Software Consulting Inc.
> Victoria, BC, Canada --- http://www.d-jackson.com
>
> On Thursday, 4 June 2015 12:07:07 UTC-7, Laura H. wrote:
>> Hi David,
>>
>> Thanks for the reply. I agree that the format statement is the next thing to look at, so I will do that and post the results. In response to your points, though,
>>
>> 1) The format statement only contains data formats (A, F, or I) and spaces (X), no carriage returns or other funky things. (I never use carriage returns.)
>>
>> 2) I read the file in vi. vi wraps the text to the size of the window, breaking at whatever character is at the edge of the screen. If you resize the window, the wrapping changes accordingly. In this case, the header line is longer than the line of data and only wraps when the window is small. The last two values in the data line always appear on a separate line, regardless of the size of the window. In vi, characters have uniform width, so there's no cheating there.
>>
>> 3) According to the Exelis documentation, there is a limit to the number of characters in a line for printf (80), and this can be overcome by using a WIDTH parameter in the openw statement. It's not obvious that this would only apply to lines consisting of a single string variable vs a number of them written one after the other. A punch card is a punch card is a punch card. You hit the end and it's over, if not, then not. I'm baffled by this.
>>
>> Interestingly, I often run into the problem of a single printf statement printing to multiple lines when I DON'T use a FORMAT statement. Putting in an explicit format usually helps, and allows me to make the line longer than when I rely on IDL standard formatting.....
>>
>>
>>
>> On Thursday, June 4, 2015 at 7:11:42 AM UTC-7, David Fanning wrote:
>>> Laura H. writes:

>>>
>>>> I'm writing an ascii file of data and some of the numbers are wrapping to a second line. On a couple of web pages, I find a suggested solution of adding width = somevalue to the end of my openw statement. I tried several different values, all the way up to 400, and found no change in my output. Any idea why this is? The relevant code and output lines are below. Interestingly, the header line (just text) prints out fine regardless of whether I use a width parameter.
>>>
>>> Well, you are using a FORMAT statement. So, I wouldn't expect a change
>>> when the WIDTH of the line is changed. If the FORMAT statement doesn't
>>> allow a single line of output, EVEN when the WIDTH of the line is
>>> changed, I would immediately suspect something is up with the FORMAT
>>> statement.
>>>
>>> Yours is so complicated, however, that I can't tell what the problem
>>> might be. Try a very simple FORMAT and see if that doesn't shed some
>>> light on this problem, before you retreat to your more complicated
>>> design.
>>>
>>> Cheers,
>>>
>>> David
>>>
>>> P.S. You ARE sure that the wrapping you are seeing is not related to the
>>> configuration of the software you are using to view the output file,
>>> right? For example, I have to turn Word Wrap off in my Notepad
>>> application to view these kinds of text files properly.
>>> --
>>> David Fanning, Ph.D.
>>> Fanning Software Consulting, Inc.
>>> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>>> Sepore ma de ni thue. ("Perhaps thou speakest truth.")
