Subject: Feature request: printing very long arrays
Posted by Helder Marchetto on Mon, 08 Jun 2015 12:48:44 GMT
View Forum Message <> Reply to Message

Hi,
I don't know if this happens only to me, but sometimes while debugging I like to look at what's inside a variable. Most of the times I use the command:

help, variable

and sometimes

print, variable

However, sometimes I'm too eager to look at what's hidden under the name and I go directly for the print option. And if I'm so stupid to do that on array of say 4096 x 4096 elements... well it takes a while and the only way to stop this useless overflow of data is to kill the IDL process.

Is there a chance we a print command that looks like this:

```
IDL> print, veryBigVariable
[      0      1      ...  999998  999999]
```

and

```
IDL> print, veryBigVariable, /fullPrint
       0       1       2       3       4       5       6       7       8       9      10      11      12      13      14      15
16      17      18      19      20      21      22      23      24      25      26      27      28      29      30      31
   32
      33      34      35      36      37      38      39      40      41      42      43      44      45      46      47
48      49      50      51      52      53      54      55      56      57      58      59      60      61      62      63
   64      65
....
```

well you got the point.

Any chance of this showing up in the future?

Cheers,
Helder

Subject: Re: Feature request: printing very long arrays
Posted by optiksguy on Tue, 09 Jun 2015 13:43:31 GMT
View Forum Message <> Reply to Message

On Monday, June 8, 2015 at 8:48:47 AM UTC-4, Helder wrote:
> Hi,
> I don't know if this happens only to me, but sometimes while debugging I like to look at what's

inside a variable. Most of the times I use the command:
>
> help, variable
>
> and sometimes
>
> print, variable
>
> However, sometimes I'm too eager to look at what's hidden under the name and I go directly for the print option. And if I'm so stupid to do that on array of say 4096 x 4096 elements... well it takes a while and the only way to stop this useless overflow of data is to kill the IDL process.
>
> Is there a chance we a print command that looks like this:
>
> IDL> print, veryBigVariable
> [0   1   ... 999998 999999]
>
> and
> IDL> print, veryBigVariable, /fullPrint
>   0   1   2   3   4   5   6   7   8   9   10   11   12   13   14   15
   16   17   18   19   20   21   22   23   24   25   26   27   28   29   30
31   32
>   33   34   35   36   37   38   39   40   41   42   43   44   45   46   47
   48   49   50   51   52   53   54   55   56   57   58   59   60   61   62
63   64   65
> ....
>
> well you got the point.
>
> Any chance of this showing up in the future?
>
> Cheers,
> Helder

+1 to this request, as I have made the same mistake too many times to count. I would guess there are backwards compatibility issues here though.

John

Subject: Re: Feature request: printing very long arrays
Posted by Lajos Foldy on Tue, 09 Jun 2015 15:57:19 GMT

View Forum Message <> Reply to Message

On Tuesday, June 9, 2015 at 3:43:33 PM UTC+2, john.c...@gmail.com wrote:
> On Monday, June 8, 2015 at 8:48:47 AM UTC-4, Helder wrote:
>> Hi,

>> I don't know if this happens only to me, but sometimes while debugging I like to look at what's inside a variable. Most of the times I use the command:
>>
>> help, variable
>>
>> and sometimes
>>
>> print, variable
>>
>> However, sometimes I'm too eager to look at what's hidden under the name and I go directly for the print option. And if I'm so stupid to do that on array of say 4096 x 4096 elements... well it takes a while and the only way to stop this useless overflow of data is to kill the IDL process.
>>
>> Is there a chance we a print command that looks like this:
>>
>> IDL> print, veryBigVariable
>> [0   1   ... 999998 999999]
>>
>> and
>> IDL> print, veryBigVariable, /fullPrint
>>   0   1   2   3   4   5   6   7   8   9   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32
>>   33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48   49   50   51   52   53   54   55   56   57   58   59   60   61   62   63   64   65
>> ....
>>
>> well you got the point.
>>
>> Any chance of this showing up in the future?
>>
>> Cheers,
>> Helder
>
>
> +1 to this request, as I have made the same mistake too many times to count. I would guess there are backwards compatibility issues here though.
>
> John

You can write your own print procedure, something like:

```
pro myprint, x, fullprint=full
help, x
n=n_elements(x)
if n le 10 or keyword_set(full) then print, x $
else print, x[0:4], '...', x[n-5:n-1]
```

end

regards,
Lajos

Subject: Re: Feature request: printing very long arrays
Posted by wlandsman on Tue, 09 Jun 2015 17:03:20 GMT

View Forum Message <> Reply to Message

On Monday, June 8, 2015 at 8:48:47 AM UTC-4, Helder wrote:

> However, sometimes I'm too eager to look at what's hidden under the name and I go directly for the print option. And if I'm so stupid to do that on array of say 4096 x 4096 elements... well it takes a while and the only way to stop this useless overflow of data is to kill the IDL process.

You should be able to interrupt the display with Control^C without killing the IDL process.

At the terminal, I still like to use the MORE capability, and instead of PRINT I use a procedure that tests for the value of !MORE, e.g.

http://idlastro.gsfc.nasa.gov/ftp/pro/misc/forprint.pro

But this doesn't work from the IDL workbench.

Subject: Re: Feature request: printing very long arrays
Posted by Helder Marchetto on Wed, 10 Jun 2015 07:56:13 GMT

View Forum Message <> Reply to Message

On Tuesday, June 9, 2015 at 5:57:20 PM UTC+2, fawltyl...@gmail.com wrote:
> On Tuesday, June 9, 2015 at 3:43:33 PM UTC+2, john.c...@gmail.com wrote:
>> On Monday, June 8, 2015 at 8:48:47 AM UTC-4, Helder wrote:
>>> Hi,
>>> I don't know if this happens only to me, but sometimes while debugging I like to look at what's inside a variable. Most of the times I use the command:
>>>
>>> help, variable
>>>
>>> and sometimes
>>>
>>> print, variable
>>>
>>> However, sometimes I'm too eager to look at what's hidden under the name and I go directly for the print option. And if I'm so stupid to do that on array of say 4096 x 4096 elements... well it takes a while and the only way to stop this useless overflow of data is to kill the IDL process.
>>>

```
>>> Is there a chance we a print command that looks like this:
>>>
>>> IDL> print, veryBigVariable
>>> [      0      1      ...  999998  999999]
>>>
>>> and
>>> IDL> print, veryBigVariable, /fullPrint
>>>        0      1      2      3      4      5      6      7      8      9     10     11     12     13     14
15     16     17     18     19     20     21     22     23     24     25     26     27     28     29     30
  31     32
>>>       33     34     35     36     37     38     39     40     41     42     43     44     45     46
47     48     49     50     51     52     53     54     55     56     57     58     59     60     61     62
  63     64     65
>>> ....
>>>
>>> well you got the point.
>>>
>>> Any chance of this showing up in the future?
>>>
>>> Cheers,
>>> Helder
>>
>>
>> +1 to this request, as I have made the same mistake too many times to count. I would guess
there are backwards compatibility issues here though.
>>
>> John
>
> You can write your own print procedure, something like:
>
> pro myprint, x, fullprint=full
> help, x
> n=n_elements(x)
> if n le 10 or keyword_set(full) then print, x $
> else print, x[0:4], '...', x[n-5:n-1]
> end
>
> regards,
> Lajos
```

Thanks Lajos,
I didn't think of that easy solution... Just made my "p" (=print) like this:

```
pro p, inVar, fullprint=fullprint
n=size(inVar)
if n[-1] eq 0 then print, 'variable undefined' $
else begin
  if n[-1] le 10 || keyword_set(fullprint) then print, inVar $
  else begin
    if n[0] eq 1 then f = '(i0)' $
               else f = '('+strtrim(n[0]-1,2)+'(i0,","),'+'(i0))'
    print, 'variable has '+strtrim(n[0],2)+' dimensions with ('+string(n[1:-3], format=f)+') elements
and a total of '+strtrim(n[-1],2)+' elements'
    print, inVar[0:1], '...', inVar[n[-1]-2:n[-1]-1]
  endelse
endelse
end
```

Not very elegant, but does the job.

Cheers,
Helder

Subject: Re: Feature request: printing very long arrays
Posted by Paul Van Delst[1] on Tue, 23 Jun 2015 18:22:12 GMT

View Forum Message <> Reply to Message

Hello,

I do this a lot too but my approach is:

```
IDL> verybigvariable=dindgen(1000000)
IDL> print, verybigvariable[0:10]
      0.0000000       1.0000000       2.0000000       3.0000000
      4.0000000       5.0000000       6.0000000       7.0000000
      8.0000000       9.0000000       10.000000
IDL> print, verybigvariable[-10:-1]
      999990.00       999991.00       999992.00       999993.00
      999994.00       999995.00       999996.00       999997.00
      999998.00       999999.00
```

That seems a lot simpler than requesting/supplying a keyword for a PRINT
statement.

What if you want to look at the middle part of the array, e.g.

```
IDL> n=n_elements(verybigvariable)
IDL> print, verybigvariable[n/2-5:n/2+5]
```

What would the PRINT keyword be?

```
IDL> print, veryBigVariable, /TruncatedPrint, $
          Location="middle", NumberToPrint=20
```

(ha ha)

Why not write you own "Inspect" procedure to implement this type of
thing? Then simply teach yourself to type "Inspect" rather than "Print",

IDL> Inspect, verybigvariable

?

cheers,

paulv

On 06/08/15 08:48, Helder wrote:
> Hi, I don't know if this happens only to me, but sometimes while
> debugging I like to look at what's inside a variable. Most of the
> times I use the command:
>
> help, variable
>
> and sometimes
>
> print, variable
>
> However, sometimes I'm too eager to look at what's hidden under the
> name and I go directly for the print option. And if I'm so stupid to
> do that on array of say 4096 x 4096 elements... well it takes a while
> and the only way to stop this useless overflow of data is to kill the
> IDL process.
>
> Is there a chance we a print command that looks like this:
>
> IDL> print, veryBigVariable [0   1   ... 999998
> 999999]
>
> and IDL> print, veryBigVariable, /fullPrint 0   1   2   3
> 4   5   6   7   8   9   10   11   12
> 13   14   15   16   17   18   19   20   21
> 22   23   24   25   26   27   28   29   30
> 31   32 33   34   35   36   37   38   39   40
> 41   42   43   44   45   46   47   48   49
> 50   51   52   53   54   55   56   57   58
> 59   60   61   62   63   64   65 ....
>
> well you got the point.
>
> Any chance of this showing up in the future?
>
> Cheers, Helder

```
>
```

Subject: Re: Feature request: printing very long arrays
Posted by Helder Marchetto on Wed, 24 Jun 2015 11:13:18 GMT
View Forum Message <> Reply to Message

On Tuesday, June 23, 2015 at 8:22:15 PM UTC+2, Paul van Delst wrote:

```
> Hello,
>
> I do this a lot too but my approach is:
>
> IDL> verybigvariable=dindgen(1000000)
> IDL> print, verybigvariable[0:10]
>        0.0000000       1.0000000       2.0000000       3.0000000
>        4.0000000       5.0000000       6.0000000       7.0000000
>        8.0000000       9.0000000       10.000000
> IDL> print, verybigvariable[-10:-1]
>        999990.00       999991.00       999992.00       999993.00
>        999994.00       999995.00       999996.00       999997.00
>        999998.00       999999.00
>
> That seems a lot simpler than requesting/supplying a keyword for a PRINT
> statement.
>
> What if you want to look at the middle part of the array, e.g.
>
> IDL> n=n_elements(verybigvariable)
> IDL> print, verybigvariable[n/2-5:n/2+5]
>
> What would the PRINT keyword be?
>
> IDL> print, veryBigVariable, /TruncatedPrint, $
>           Location="middle", NumberToPrint=20
>
> (ha ha)
>
> Why not write you own "Inspect" procedure to implement this type of
> thing? Then simply teach yourself to type "Inspect" rather than "Print",
>
> IDL> Inspect, verybigvariable
>
> ?
>
> cheers,
>
> paulv
>
```

> On 06/08/15 08:48, Helder wrote:
>> Hi, I don't know if this happens only to me, but sometimes while
>> debugging I like to look at what's inside a variable. Most of the
>> times I use the command:
>>
>> help, variable
>>
>> and sometimes
>>
>> print, variable
>>
>> However, sometimes I'm too eager to look at what's hidden under the
>> name and I go directly for the print option. And if I'm so stupid to
>> do that on array of say 4096 x 4096 elements... well it takes a while
>> and the only way to stop this useless overflow of data is to kill the
>> IDL process.
>>
>> Is there a chance we a print command that looks like this:
>>
>> IDL> print, veryBigVariable [0   1   ... 999998
>> 999999]
>>
>> and IDL> print, veryBigVariable, /fullPrint 0   1   2   3
>> 4   5   6   7   8   9   10   11   12
>> 13   14   15   16   17   18   19   20   21
>> 22   23   24   25   26   27   28   29   30
>> 31   32 33   34   35   36   37   38   39   40
>> 41   42   43   44   45   46   47   48   49
>> 50   51   52   53   54   55   56   57   58
>> 59   60   61   62   63   64   65 ....
>>
>> well you got the point.
>>
>> Any chance of this showing up in the future?
>>
>> Cheers, Helder
>>

Hi Paul,
thanks for the heads up. I wrote down this procedure and called it p. It works pretty well for now.
The reason I did this in the first place, was to avoid

```
help, unknownVar
print, unknownVar[0:10]
```

Your approach works only if you know that it has "at least" 11 parameters.
try

```
a = 0
```

```
print, a[0:10]
```

So that's why I don't want to use the a completely different print pro.
It seems like modifying the print pro would have tooooooo many consequences.

Cheers,
Helder