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
>
```