Subject: Re: STRING with FORMAT keyword and REFORM
Posted by Bruce Bowler on Wed, 11 Jan 2006 15:38:31 GMT
View Forum Message <> Reply to Message

On Wed, 11 Jan 2006 10:11:47 -0500, Ben Tupper put fingers to keyboard and said:

```
>  Hello All,
>
>  I just bumped into a funny thing.  When calling
>
>  result =  STRING(array, FORMAT = '(something')
>
>  a 2d array is reformed into a 1d array.
>
>  Here's an example...
>
>  print, !VERSION
>  ;{ ppc darwin unix Mac OS X 6.2 Jun 20 2005      32      32}
>
>  number = REBIN(INDGEN(1,10), 10,10)
>  text = STRING(number, FORMAT = '(I2.2)')
>  help, number, text
>
>  ;NUMBER         INT       = Array[10, 10]
>  ;TEXT           STRING    = Array[100]    <<<<< reformed, nuts!
>
>  text = STRING(number)
>  help, number, text
>  ;NUMBER         INT       = Array[10, 10]
>  ;TEXT           STRING    = Array[10, 10] <<<< not reformed
>
>  text = STRING(text, format = '(I2.2)')
>  help, number, text
>
>  ;NUMBER         INT       = Array[10, 10]
>  ;TEXT           STRING    = Array[100]   <<< still reformed
>
>  I can re-reform the array, but nothing pops out at me from the online
>  help about this.  Is this expected behavior?
>
>  Thanks,
>  Ben
```

Well since IDL formats are similar to FORTRAN formats, it doesn't surprise me. If you were to use that format statement in FORTRAN, you'd get 100 lines of output. I think you really want to use a format of '(10I2.2)' or '10(I2.2)'. See the section in "Building IDL apps" called "Using

explicitly formatted I/O" starting at page 161 (at least that's where it
is in my V5 doc set :-)

Bruce

--
+------------------+------------------------------------ ------------+
Bruce Bowler | I am the woman who gave bad taste a good name. -
1.207.633.9600 | Bette Middler
bbowler@bigelow.org |
+------------------+------------------------------------ ------------+

Subject: Re: STRING with FORMAT keyword and REFORM
Posted by btt on Wed, 11 Jan 2006 15:55:33 GMT
View Forum Message <> Reply to Message

Bruce Bowler wrote:
> On Wed, 11 Jan 2006 10:11:47 -0500, Ben Tupper put fingers to keyboard and
> said:
>
>
>> Hello All,
>>
>> I just bumped into a funny thing. When calling
>>
>> result = STRING(array, FORMAT = '(something')
>>
>> a 2d array is reformed into a 1d array.
>>
>> Here's an example...
>>
>> print, !VERSION
>> ;{ ppc darwin unix Mac OS X 6.2 Jun 20 2005 32 32}
>>
>> number = REBIN(INDGEN(1,10), 10,10)
>> text = STRING(number, FORMAT = '(I2.2)')
>> help, number, text
>>
>> ;NUMBER INT = Array[10, 10]
>> ;TEXT STRING = Array[100] <<<<< reformed, nuts!
>>
>> text = STRING(number)
>> help, number, text
>> ;NUMBER INT = Array[10, 10]
>> ;TEXT STRING = Array[10, 10] <<<< not reformed
>>
>> text = STRING(text, format = '(I2.2)')

>> help, number, text
>>
>> ;NUMBER   INT   = Array[10, 10]
>> ;TEXT   STRING   = Array[100]   <<< still reformed
>>
>> I can re-reform the array, but nothing pops out at me from the online
>> help about this. Is this expected behavior?
>>
>> Thanks,
>> Ben
>
>
> Well since IDL formats are similar to FORTRAN formats, it doesn't surprise
> me.   If you were to use that format statement in FORTRAN, you'd get 100
> lines of output. I think you really want to use a format of '(10I2.2)' or
> '10(I2.2)'. See the section in "Building IDL apps" called "Using
> explicitly formatted I/O" starting at page 161 (at least that's where it
> is in my V5 doc set :-)
>
> Bruce
>

Hi Bruce,

Maybe we should have a cup-and-string communication system between our
desks instead of the newsgroup.

Those were good ideas - but neither of the options worked.

```
IDL> number = REBIN(INDGEN(1,10), 10,10)
IDL> text = STRING(number, FORMAT = '(10I2.2)')
IDL> help, number,text
NUMBER          INT       = Array[10, 10]
TEXT            STRING    = Array[10]
IDL> text = STRING(number, FORMAT = '(10(I2.2))')
IDL> help, number,text
NUMBER          INT       = Array[10, 10]
TEXT            STRING    = Array[10]
```

Here's what I found in the Building Apps guide regarding string
processing on non-scalars...

"If the argument is an array instead of a scalar, the function returns
an array result with the same structure as the argument. Each element of
the result corresponds to an element of the argument."

It makes me think it is a bug, but I'm not sure since the FORMAT keyword

has a lot more features than I can wrap my brain around. In fact, given my history, I am sure that it will turn out to NOT be a bug since I think it might be a bug. You follow me on that?

Thanks,
Ben
