
Subject: Re: 48 variable limit?

Posted by [R.G. Stockwell](#) on Tue, 06 May 2003 14:39:23 GMT

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

"Shane Mayor" <shane@ucar.edu> wrote in message
news:4d5b78c4.0305051207.34690816@posting.google.com...

> Hi,
> I've been given an ASCII file of numbers 50 columns wide
> by 6600 rows long, blank-space separated values.
> I've tried both read_ascii and readf, reads and have concluded
> that these routines can only read up to 48 variables in a row.
> Can anybody confirm this and/or suggest alternatives?
> Thanks much,
> Shane Mayor
> NCAR
> Boulder, CO

Hi Shane,

As a wild guess as to why it doesn't work, are there some columns that
are not separated by a space (like a -9999 used for a bad data file, in a
column only 4 digits wide)?

I did a quicky test, and it works fine.

Here is the idl code:

```
filename = 'C:\junktest.txt'  
d = read_ascii(filename)  
help,d  
end
```

Here is the txt file (54 columns, 3 rows)

```
1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2  
3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9  
1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2  
3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9  
1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2  
3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9
```

And here is the command log

```
IDL> .GO
```

```
% Compiled module: UNIQ.
```

```
D STRUCT = -> <Anonymous> Array[1]
```

```
IDL> help,d,/st
```

```
** Structure <17039a8>, 1 tags, length=648, data length=648, refs=1:
```

```
FIELD01 FLOAT Array[54, 3]
```

```
IDL> print,d.field01
```

```
1.00000 2.00000 3.00000 4.00000 5.00000 6.00000 7.00000 8.00000 9.00000  
1.00000 2.00000 3.00000 4.00000 5.00000 6.00000
```

```
7.00000 8.00000 9.00000 1.00000 2.00000 3.00000 4.00000 5.00000 6.00000  
7.00000 8.00000 9.00000 1.00000 2.00000 3.00000
```

```
4.00000 5.00000 6.00000 7.00000 8.00000 9.00000 1.00000 2.00000 3.00000  
4.00000 5.00000 6.00000 7.00000 8.00000 9.00000
```

```
1.00000 2.00000 3.00000 4.00000 5.00000 6.00000 7.00000 8.00000 9.00000
```

```
1.00000 2.00000 3.00000 4.00000 5.00000 6.00000 7.00000 8.00000 9.00000  
1.00000 2.00000 3.00000 4.00000 5.00000 6.00000
```

```
7.00000 8.00000 9.00000 1.00000 2.00000 3.00000 4.00000 5.00000 6.00000  
7.00000 8.00000 9.00000 1.00000 2.00000 3.00000
```

```
4.00000 5.00000 6.00000 7.00000 8.00000 9.00000 1.00000 2.00000 3.00000  
4.00000 5.00000 6.00000 7.00000 8.00000 9.00000
```

```
1.00000 2.00000 3.00000 4.00000 5.00000 6.00000 7.00000 8.00000 9.00000
```

```
1.00000 2.00000 3.00000 4.00000 5.00000 6.00000 7.00000 8.00000 9.00000  
1.00000 2.00000 3.00000 4.00000 5.00000 6.00000
```

```
7.00000 8.00000 9.00000 1.00000 2.00000 3.00000 4.00000 5.00000 6.00000  
7.00000 8.00000 9.00000 1.00000 2.00000 3.00000
```

```
4.00000 5.00000 6.00000 7.00000 8.00000 9.00000 1.00000 2.00000 3.00000  
4.00000 5.00000 6.00000 7.00000 8.00000 9.00000
```

```
1.00000 2.00000 3.00000 4.00000 5.00000 6.00000 7.00000 8.00000 9.00000
```
