
Subject: Re: How to read (large) ASCII arrays into PV-Wave
Posted by [David Fanning](#) on Mon, 17 Jun 2002 14:09:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Surendar Jeyadev (jeyadev@wrc.xerox.bounceback.com) writes:

> I have some floating point, ASCII arrays that I am trying to
> read into PV-Wave. They are not particularly large:
> 232 columns x 50 rows. All numbers are of the form bbbxxx.xxx
> where the 'b' are blank(white) space and the 'x' are digits.
> I have not read such arrays of such size in the past and I was
> surprised to see PV-Wave choke on them. Here is the [error]
>
> % Input line is too long for input buffer of 2048 characters.
> % READF: Error encountered reading from file. Unit: 1
> File: tt
> % Execution halted at \$MAIN\$ (READF).
>
> Any help would be greatly appreciated.

Are you transferring these files from someplace else?
I think that they don't have the required carriage return,
line feed characters on the end of the line. So IDL thinks
this file contains one line. This can happen when UNIX
ASCII files are transferred improperly to Windows machines,
for example. Alternatively, you could use a FORMAT keyword
on your READF statement to read the file.

Cheers,

David

--

David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: How to read (large) ASCII arrays into PV-Wave
Posted by [thompson](#) on Mon, 17 Jun 2002 15:46:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

jeyadev@wrc.xerox.bounceback.com (Surendar Jeyadev) writes:

> I have some floating point, ASCII arrays that I am trying to
> read into PV-Wave. They are not particularly large:
> 232 columns x 50 rows. All numbers are of the form bbbxxx.xxx

> where the 'b' are blank(white) space and the 'x' are digits.
> I have not read such arrays of such size in the past and I was
> surprised to see PV-Wave choke on them. Here is the session log:

(stuff deleted)

```
> WAVE> d = fltarr(233,50)
> WAVE> openr, 1, 'tt'
> WAVE> readf, 1, d
> % Input line is too long for input buffer of 2048 characters.
> % READF: Error encountered reading from file. Unit: 1
>     File: tt
> % Execution halted at $MAIN$ (READF).
> WAVE>exit
```

> 'wc' tells me that each line is 2085 characters long.

(rest deleted)

Evidently IDL no longer has this limitation, or else it's a much bigger number. I was able to read in a simulation of your data file without any problem in IDL/v5.4.

Here's a simple idea. Try using a text editor to reformat the "paragraphs" into shorter lines. For example, in Emacs this could be done by selecting the entire file, and then executing the command "fill-region". The numbers won't line up into nice distinct columns anymore, but you should still be able to read the array correctly with the above commands, because the numbers are in the right order.

I tried using the "\$" qualifier on the format specifier, but apparently that only works on output, not input.

If reformatting the input file doesn't appeal to you, it should be possible to read in each line of the file as a byte array, convert it to strings in sections, and then read in the data from these substrings. (Does Wave have the READS command?) That's the sort of thing that makes one refer to the euphemism "I leave it as an exercise for the reader." ;^)

William Thompson

Subject: Re: How to read (large) ASCII arrays into PV-Wave
Posted by [richard hilton](#) on Tue, 18 Jun 2002 08:42:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

This might not be the problem but you have stated that:

> They are not particularly large: 232 columns x 50 rows

But have then declared the array as:

```
> d = ftarr(233,50)
```

wouldn't this array then be looking past the end of the ascii data when the readf command is used!!

I'm relatively new to this so maybe I've missed something obvious. Or maybe this was just a typo in the original message.

Rich

"William Thompson" <thompson@orpheus.nascom.nasa.gov> wrote in message news:ael09i\$noe\$1@skates.gsfc.nasa.gov...

> jeyadev@wrc.xerox.bounceback.com (Surendar Jeyadev) writes:

>

>> I have some floating point, ASCII arrays that I am trying to

>> read into PV-Wave. They are not particularly large:

>> 232 columns x 50 rows. All numbers are of the form bbbxxx.xxx

>> where the 'b' are blank(white) space and the 'x' are digits.

>> I have not read such arrays of such size in the past and I was

>> surprised to see PV-Wave choke on them. Here is the session log:

>

> (stuff deleted)

>

```
>> WAVE> d = ftarr(233,50)
```

```
>> WAVE> openr, 1, 'tt'
```

```
>> WAVE> readf, 1, d
```

```
>> % Input line is too long for input buffer of 2048 characters.
```

```
>> % READF: Error encountered reading from file. Unit: 1
```

```
>>     File: tt
```

```
>> % Execution halted at $MAIN$ (READF).
```

```
>> WAVE>exit
```

>

>> 'wc' tells me that each line is 2085 characters long.

>

> (rest deleted)

>

> Evidently IDL no longer has this limitation, or else it's a much bigger number.

> I was able to read in a simulation of your data file without any problem

in

> IDL/v5.4.

>

> Here's a simple idea. Try using a text editor to reformat the "paragraphs" into shorter lines. For example, in Emacs this could be done by selecting the entire file, and then executing the command "fill-region". The numbers won't line up into nice distinct columns anymore, but you should still be able to read the array correctly with the above commands, because the numbers are in the right order.

> I tried using the "\$" qualifier on the format specifier, but apparently that only works on output, not input.

> If reformatting the input file doesn't appeal to you, it should be possible to read in each line of the file as a byte array, convert it to strings in sections, and then read in the data from these substrings. (Does Wave have the READS command?) That's the sort of thing that makes one refer to the euphemism "I leave it as an exercise for the reader." ;^)

> William Thompson
