
Subject: Re: Reading a very large ascii data file
Posted by [mvukovic](#) on Fri, 24 Aug 2001 22:22:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

Paul van Delst <paul.vandelst@noaa.gov> wrote in message
news:<3B8698E0.B3F13251@noaa.gov>...

> Mirko Vukovic wrote:

>>

>> I am reading some large ascii data files in csv (comma separated
>> fields) format, and would like to speed the process up.

>>

>> I recall someone discussing reading such files as binaries and then
>> converting to ascii after finding line breaks, but was un-able to find
>> the discussion on the group.

>>

>> Can anyone offer pointers, code, or suggestions on who might have
>> discussed it (so that I can look again on the newsgroup).

>

> Can you provide more information about your data files? E.g. are the number of columns
> fixed? Are the number of lines fixed? If not, is there a maximum number of lines which the
> files won't exceed?

>

> Try the DDREAD.PRO and associated IDL code. Have a look at

>

> http://www.dfanning.com/tips/unknown_rows.html

>

> for some issues and a link to the source code.

>

> paulv

Thanks for the comments,

The file format is variable. The file contains a log of data of a
variable number of channels, and of arbitrary duration. It is
generated by the TrendLink software from Fluke.

The file consists of a header, which has as many lines as diagnostics.
Next comes the data, with one column for the time and date, and a
column each for each channel.

I therefore use a two-pass system. In the first, I read all the
lines, and count their number, and from the last line also extract the
number of channels.

With this info, I then initialize the header and data structures, and
then go again through the file, and store the stuff.

In that sense, I am not using the very slow procedure noted by martin

(appending a line to the matrix). However, I am going explicitly through a very long loop, twice.

One methode may be to open the file in binary mode, get info about the number of bytes, initialize a byte vector to appropriate size, and then read the file into it. Now, with the file stored in memory (although it can be megabytes in size), go through it, ``reading" line by line.

This actually looks to be a quite generic procedure. Any idea whether it has been implemented already?

Any more suggestions?

Thanks,

Mirko
