Subject: Re: Read Total lines in an ASCII file
Posted by condor on Sat, 21 Dec 2002 00:14:51 GMT
View Forum Message <> Reply to Message

"Mark Hadfield" <m.hadfield@niwa.co.nz> wrote in message
news:<atqt2j$58j$1@newsreader.mailgate.org>...
> "Paul Woodford" <cpwoodford@spamcop.net> wrote in message
> news:cpwoodford-C4345E.22403817122002@corp.supernews.com...
>> Would it be possible to find the length of the file, read it into to
>> a byte array, and then convert it to text?
>
> Yes, but:
>
> - If you take the 1D byte array that would result from reading the
> file and convert it to a string, then you don't get a string array,
> you just get a string with line-separator characters in it. So
> there's a bit of splitting to be done, and you really should handle
> the various line separators supported by the different platforms.

As far as I recall, the OP just wanted to know the number of lines,
not necessarily try to convert them into anything. The only deviation
from the usual 10b linefeed out there on idl'ish platforms is the DOS
[10b,13b] LF/CR, right? Or do VMS systems do yet something different?
How do the various suggested methods hold up on VMS?

If the LF and CR/LF are the only two, the only thing you'd have to do
is counting the number of 10b in the byte-filed:

```
 f = read_binary('Big_honking_example_file')
 h = histogram(f)
 print,h[10]
    1479054
```

If you're really intent on accessing the individual data items in the
file, you could retain the reverse indices of the histogram for a
handy field of pointers to each individual line that can be converted
into a string at will...

> - It doesn't work on compressed files, because you don't know how
> many bytes there are in a compressed file until you've read it. So
> you have to read the byte data in chunks, trap the error when the
> final read hits the end of the file, and join the chunks together.
>
>> Paul, who is too lazy to figure it out himself

>
> Mark, who has thought about it, but can't be bothered actually trying
> it.