Subject: Re: A (too?) simple question about importing data
Posted by promashkin on Thu, 22 Jun 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Hi Michael,
If the data bearing strings are well-defined (e.g., data or filling with
"bad" number are present always), then the following would work:

```
; create a file with dummy data first...
temp = '0330 00 00 00 00 00000  50.60   03.40 000 0.0 USGS_EU_Catalogue'
; make 100 rows in that file
temp = replicate(temp, 100)
openw, unit, 'temp_junk.txt', /get_lun
printf, unit, temp
free_lun, unit
; now we have a file to try to read.
; open the file for reading
openr, unit, 'temp_junk.txt', /get_lun
; Create STR_FORM that reflects format of data in one file row
str_form = {data:fltarr(10), note:''}
; create array of STR_FORMs big enough to read the whole file at once.
; lets pretend we don't know file length in advance.
data_array = replicate(str_form, 2000)
; in this case it is way too big. Not to worry.
readf, unit, data_array
;% READF: End of file encountered. Unit: 100
;       File: IDE data:idl:ukmo:temp_junk.txt
;% Execution halted at:  $MAIN$
; Sure enough, reading failed. But we know file size now.
; The number of fields (10 values and a string) is 11, so we do:
print, (fstat(unit)).transfer_count / 11
;        100
; this means we had 100 rows in the file. Resize the array:
data_array = replicate(str_form, 100)
; start over in the file:
point_lun, unit, 0
; read the array:
readf, unit, data_array
print, data_array[2]
;{      330.000      0.00000      0.00000      0.00000      0.00000
;      0.00000      50.6000      3.40000      0.00000      0.00000
; USGS_EU_Catalogue}
```

I discovered (for myself - the Pros knew that all along, I'd think :-)
that reading past the end of file and then resizing the read buffer is a
lot faster than reading accurately line by line inside a WHILE NOT EOF
loop. IDL can read a 100x100000 FLTARR directly a thousand times faster
than going through a 100000 line loop, reading a 1000 point vector at a time.

Will this work?
Cheers,
Pavel

Michael Spranger wrote:
>
> Hi,
> another beginner's question, this time about reading data:
> I want to read data from ASCII files into a structure. The data look
> as follows:
>
> YYYY MM DD HH II SSSSS PPPPPP LLLLLLL KKK RRR
> 0330 00 00 00 00 00000 50.60   03.40 000 0.0 USGS_EU_Catalogue
>
> the structure, type, and length of variables are always the same, only
> the the order might change and some data might be missing. The last
> row (without header) contains comments only.
>
> Sounds easy, is (probably) easy - but (still) too difficult for me.
>
> Thanks for any help/ hints in advance,
> Michael
