Subject: Re: Data management... but Execution halts!!! T.T
Posted by Rick Towler on Tue, 14 Nov 2006 19:28:05 GMT
View Forum Message <> Reply to Message

And that last time I checked (a few years ago) the structure tags returned by read_ascii differed depending on if you had < 10 or >= 10 columns of data. This makes it less than ideal of your files have "optional" columns of data that occasionally bring the column count to 10 or more.

< 10 -> data.field1 data.field2 ...
>=10 -> data.field01 data.field02 ...

-Rick

greg michael wrote:
> The problem with read_ascii is the template. If your file has anything
> other than a trivial format (as this one does) - i.e. a header of some
> sort, or mixed types, you need to use a template. And you can't
> (reasonably) generate this programmatically, but only via the
> ascii_template GUI function. Try it, then you'll want to learn the
> other way...
>
> regards,
> Greg
>
>
> Jeff N. wrote:
>> Do people not use READ_ASCII()? Is it really slow, or tempermental or
>> something? I saved the original poster's data to a text file, and this
>> seemed to work for me:
>>
>> data = read_ascii('file.txt', delimiter='|')
>>
>> There's a NaN on the end that I'd have to chop off, but that's easily
>> done, but it also makes me wonder if the function doesn't work in a lot
>> of cases maybe? I confess i've never used it myself before now, but
>> that's b/c i never knew it existed until a few minutes ago :)
>>
>> Jeff
>>
>>
>>
>> kim20026@gmail.com wrote:
>>> I requested some meteological data to a governmental institute. To
>>> reduce the file size, they sent me this way...
>>>
>>> 90|2000|1|1|25|25|26|26|26|25|26|28|31|34|36|39|40|38|38|37|

```
35|33|35|37|34|34|35|29|90|2000|1|2|30|32|29|28|31|33|55|65|
74|68|72|75|71|83|85|77|69|60|51|43|41|34|34|29|90|2000|1|3|
25|20|12|9|8|5|-2|-3|14|27|31|42|51|54|48|51|37|28|15|11|6|-
12|0|-13|90|2000|1|4|-7|-5|-8|-10|-21|-21|-13|-18|-2|23|36|4
0|50|49|42|37|29|22|9|3|0|0|5|4|90|2000|1|5|1|4|12|10|15|15|
9|6|0|-1|1|3|5|4|4|6|8|8|10|9|9|13|13|17|90|2000|1|6|17|18|2
3|23|24|28|28|27|25|26|45|73|87|90|81|72|58|54|39|19|-2|-11|
-17|-26|90|2000|1|7|-31|-39|-43|-55|-67|-77|-85|-86|-78|-71|
-63|-63|-52|-43|-39|-41|-49|-51|-53|-54|-58|-53|-54|-57|90|2
000|1|8|-57|-52|-49|-48|-45|-39|-36|-30|-25|-18|-11|-8|0|2|1 2|7|-3|-7|-7|-7|-6|-16|-15|-17|
>>>  ...
>>>
>>>  This is only one line, but contains a matrix of 28 columns * 157824L
>>>  rows. As you guys may know, Excel can only 65,000 rows approximately,
>>>  and I need to divide this into three to read this file correctly.
>>>
>>>  I have tried like this so far, but everytime I try, I have the same
>>>  error message and the program stops.
>>>
>>>
>>>  ----------------------------------------
>>>  Pro temp1
>>>
>>>  ; Main working directory (location of IDL procedures and functions)
>>>  WorkDir         = 'D:\MODIS_Documents\MetData\IDLPractice'
>>>
>>>  s=''
>>>  Openr, 1, 'temp1.txt'
>>>  Readf, 1, s
>>>  print, strlen(s)
>>>  close, 1
>>>
>>>  data=fix(strsplit(s, '|', /extract))
>>>  help, data
>>>  data=reform(data, 28, 157824L) ; <- Execution halts right here
>>>  everytime!!!
>>>  print, data[*, 0]
>>>  print, data[*, 1]
>>>
>>>  for i=0L, 157824-1 do begin
>>>          if i mod 52608L eq 0 then begin
>>>                  close, /all
>>>                  file_no=i / 52608L + 1   ; 1, 2, 3
>>>                  openw, 1, 'out'+string(file_no, format='(I1)')+'.txt'
>>>  ;ex) out1.txt
>>>          endif
>>>          printf, 1, strjoin(string(data[*, i], format='(I4)'), '')
>>>
```

>>> endfor
>>> close, /all
>>> end
>>> ---------- ----------
>>>
>>> Please let me know if you have any solutions (or suggestions at least.
>>> It's urgent!! T.T)
>