
Subject: Re: Extracting variables from ascii files that are not in columnar format
Posted by [natha](#) on Fri, 09 Apr 2010 19:21:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

I'll try something like that

```
aux=""
OPENR, un$lun, file, /GET_LUN

WHILE ~EOF(un$lun) DO BEGIN

  READF, un$lun, aux

  aux_str=STRSPLIT(aux,' ',/EXTRACT)

  PRINT, 'label', aux_str[0:1]
  PRINT, 'values', aux_str[2:*]

  ;; Then you can concatenate, etc...

ENDWHILE

FREE_LUN, un$lun
```

Subject: Re: Extracting variables from ascii files that are not in columnar format
Posted by [Gray](#) on Fri, 09 Apr 2010 19:51:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Apr 9, 3:21 pm, nata <bernat.puigdomen...@gmail.com> wrote:

```
> I'll try something like that
>
> aux=""
> OPENR, un$lun, file, /GET_LUN
>
> WHILE ~EOF(un$lun) DO BEGIN
>
>   READF, un$lun, aux
>
>   aux_str=STRSPLIT(aux,' ',/EXTRACT)
>
>   PRINT, 'label', aux_str[0:1]
>   PRINT, 'values', aux_str[2:*]
>
>   ;; Then you can concatenate, etc...
>
> ENDWHILE
>
```

```
> FREE_LUN, un$lun
```

Or, you can read everything at once with `file_lines` and a string array, then loop through the array and `strsplit`. If you know that every line has the same format, you can use `astrolib`'s `gettok` to pick out the header names. Printing in columns like that is a little more complicated, and requires padding. Example:

```
n = file_lines(file)
data = strarr(n)
heads = strarr(n)
openr, unit, file, /get_lin
readf, unit, data
close, unit & free_lun, unit
heads = gettok(data, '=')
print, heads, format='('+n+'A12)'
lens = intarr(n)
for i=0L,n-1 do lens[i] = n_elements(strsplit(data[i],/regex))
npad = max(lens)
padded_data = strarr(n,npad)
for i=0L,n-1 do padded_data[i,0] = strsplit(data[i],/regex)
print, transpose(padded_data), format='('+n+'A12)'
```

Subject: Re: Extracting variables from ascii files that are not in columnar format
Posted by [Gray](#) on Fri, 09 Apr 2010 20:31:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Apr 9, 3:51 pm, Gray <grayliketheco...@gmail.com> wrote:

> On Apr 9, 3:21 pm, nata <bernat.puigdomen...@gmail.com> wrote:

>

>

>

>

>

>> I'll try something like that

>

>> aux=""

>> OPENR, un\$lun, file, /GET_LUN

>

>> WHILE ~EOF(un\$lun) DO BEGIN

>

>> READF, un\$lun, aux

>

>> aux_str=STRSPLIT(aux, ' ',/EXTRACT)

>

>> PRINT, 'label', aux_str[0:1]

>> PRINT, 'values', aux_str[2:*

```

>
>> ;; Then you can concatenate, etc...
>
>> ENDWHILE
>
>> FREE_LUN, un$lun
>
> Or, you can read everything at once with file_lines and a string
> array, then loop through the array and strsplit. If you know that
> every line has the same format, you can use astrolib's gettok to pick
> out the header names. Printing in columns like that is a little more
> complicated, and requires padding. Example:
>
> n = file_lines(file)
> data = strarr(n)
> heads = strarr(n)
> openr, unit, file, /get_lin
> readf, unit, data
> close, unit & free_lun, unit
> heads = gettok(data, '=')
> print, heads, format=('+' + n + 'A12')
> lens = intarr(n)
> for i=0L,n-1 do lens[i] = n_elements(strsplit(data[i],/regex))
> npad = max(lens)
> padded_data = strarr(n,npad)
> for i=0L,n-1 do padded_data[i,0] = strsplit(data[i],/regex)
> print, transpose(padded_data), format=('+' + n + 'A12')

```

Oops, forgot /extract in the 2nd strsplit call. Also, you may need to strtrim n for the format codes.
