Subject: Re: ascii_template advanced
Posted by David Fanning on Thu, 07 Sep 2006 13:40:48 GMT
View Forum Message <> Reply to Message

maye writes:

> The problem I have with the ascii_template/read_ascii combination is
> really the unknown amount of comment lines at the beginning plus the
> fact that I want to read the header line in AFTER those unknown amount
> of lines.
> Because the template let's me ignore the comment lines, but then the
> first line of my data structure will contain the variable names.
> Also, when I use the DATA_START keyword of read_ascii, it counts again
> from the beginning of the file, not from where I already defined the
> data to be of value (meaning withOUT the comment lines).
> If it would work like that, I could define the template to ignore
> comment lines, and then skip one extra line at the read_ascii command,
> using it for the HEADER keyword to read in the variable names. Would it
> not make more sense if things work like that?
> I hope I made myself clear? :-/ By the way, if somebody wonders: I want
> the variables names for a GUI to be clickable. ;)
> Does anybody have a solution for this?

I don't have a solution, but I'm certain there must
be one, since I've never had to use READ_ASCII in
20+ years of IDL programming. Perhaps you have to
write a program. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui.
(Opata Indian saying, meaning "Perhaps thou speakest truth.")

Subject: Re: ascii_template advanced
Posted by Jo Klein on Thu, 07 Sep 2006 14:53:31 GMT
View Forum Message <> Reply to Message

If you don't feel confident enough to program a filter for this, you can
simply preprocess your input data before feeding it into IDL with e.g. sed.
You could do:
sed 's/^#//' mydata.txt > mydatawithoutcomments.txt
to remove all lines starting with #.

Cheers,
Jo

Subject: Re: ascii_template advanced
Posted by Mike Wallace on Thu, 07 Sep 2006 15:05:12 GMT
View Forum Message <> Reply to Message

Create the template with ascii_template() as you normally would for one
of your data files. For each of the data files, open it and count the
the number of lines that you want to skip. Then change the datastart
value in the template structure to start at the correct line. Now call
read_ascii with your updated template.

So, in short, hack the template structure returned from ascii_template()
for each file so that it starts reading on the correct line.

-Mike

Subject: Re: ascii_template advanced
Posted by maye on Thu, 07 Sep 2006 20:02:31 GMT
View Forum Message <> Reply to Message

Mike Wallace wrote:
> Create the template with ascii_template() as you normally would for one
> of your data files. For each of the data files, open it and count the
> the number of lines that you want to skip. Then change the datastart
> value in the template structure to start at the correct line. Now call
> read_ascii with your updated template.
>
> So, in short, hack the template structure returned from ascii_template()
> for each file so that it starts reading on the correct line.
>
> -Mike

Thanks guys, I will try these two last hints out, although I then
almost will reach David's suggestion, which I tried to avoid by
fiddling with ascii_template. ;)
Regards,
Michael

Subject: Re: ascii_template advanced
Posted by greg michael on Fri, 08 Sep 2006 16:04:17 GMT
View Forum Message <> Reply to Message

read_ascii is an IDL joke to scare away new users... try something easier like:

```
 s=strarr(file_lines(filename))
 openr,1,filename
 readf,1,s
 close,1
 s=(s[where(strmid(s,0,1) ne '#')])[1:*] ;get rid of the comments and
field names
 time=strmid(s,0,8) ;time is a tricky format - take it as a string for
now
 s=strmid(s,9) ;chop the time off
 m=replicate({voltage:0.,current:0.,temp:0.},n_elements(s)) ;make your
structure array
 s=strjoin(strsplit(strjoin(s,','),';',/extract),',') ;swap those pesky
semicolons for commas
 reads,s,m ;make the conversion
```

regards,
greg