Subject: Breaking files
Posted by fgg on Thu, 13 May 2010 21:05:14 GMT
View Forum Message <> Reply to Message

Hello,

I have a code that is designed to read and process text files with
this format:

=xxx
id = 1
a = ...
b = ...
c = ...

It starts by reading the data in using file_lines and a string array:

infile = '/path/filename'
n = file_lines(infile)
data = strarr(n)
openr, inunit, infile, /get_lun
readf, inunit, data

... and ends by exporting stuff to another text file.

Now I need to deal with input text files that have a slightly
different format:

=xxx
id = 1
a = ...
b = ...
c = ...
=xxx
id = 2
a = ...
b = ...
c = ...
=xxx
id = 3
a = ...
b = ...
c = ...

They are just multiple files appended together. I was wondering if
there is a simple way of breaking this new big file into multiple
files with the old format so I don't have to change my code
considerably (maybe using the "=xxx" lines to indicate where each file

starts).

Sorry if this is too obvious. I'm new to IDL.

Thank you.

Subject: Re: Breaking files
Posted by Gray on Fri, 14 May 2010 21:18:32 GMT
View Forum Message <> Reply to Message

On May 14, 4:31 pm, fgg <fabioguimaraesgoncal...@gmail.com> wrote:
> Thanks for the suggestions. I can use WHERE to flag the those lines...
> I'm just not really sure how to create a loop to run the entire script
> for the first block of lines, then for the second and so one.
>
> I use a Mac so I probably have the split utility. I'll take a look.
> Thanks!

All you do is pick out the data you want.

```
these = [where(stregex(data_array,'=xxx',/
boolean),nthese),n_elements(data_array)-1]
for i=0L,nthese-1 do begin
  these_data = data_array[these[i]:these[i+1]]
  ;operate on these_data
  ...
endfor
```

Subject: Re: Breaking files
Posted by fgg on Mon, 17 May 2010 18:37:54 GMT
View Forum Message <> Reply to Message

> All you do is pick out the data you want.
>
> these = [where(stregex(data_array,'=xxx',/
> boolean),nthese),n_elements(data_array)-1]
> for i=0L,nthese-1 do begin
> these_data = data_array[these[i]:these[i+1]]
> ;operate on these_data
> ...
> endfor

This is all I needed to finish the script! Thank you, Gray.
