Subject: operating on tab-/comma-delimited files
Posted by Seb on Wed, 28 Aug 2013 16:56:27 GMT
View Forum Message <> Reply to Message

Hi,

I am in the process of updating code I've inherited as I learn IDL, and noticed that it uses the following workflow for fixing errors in tab-/comma-delimited files:

1. Use file_search to generate a list of files to work on

2. Use FOR loop to iterate through each file

   2.1 Create a float array with the same dimensions as input file
   2.2 Load the data into a string array with 'openr' -> 'strarr' -> 'readf' -> 'close'
   2.3 Loop through each line doing various tests/operations
   2.4 Copy the fixed data into the float array in 2.1

3. Write the float array with 'openw' -> 'printf' -> 'close'

Based on what I'm reading, it seems this should be more efficient by using 'read_ascii' and then fixing the errors with array indexing and then writing the data back. Is this worth pursuing, or are there any recommendations for a module I should know about?

Thanks,

--
Seb

Subject: Re: operating on tab-/comma-delimited files
Posted by David Fanning on Wed, 28 Aug 2013 17:11:47 GMT
View Forum Message <> Reply to Message

Seb writes:

>
> Hi,
>
> I am in the process of updating code I've inherited as I learn IDL, and
> noticed that it uses the following workflow for fixing errors in
> tab-/comma-delimited files:
>
> 1. Use file_search to generate a list of files to work on
>

> 2. Use FOR loop to iterate through each file
>
> 2.1 Create a float array with the same dimensions as input file
> 2.2 Load the data into a string array with 'openr' -> 'strarr' ->
> 'readf' -> 'close'
> 2.3 Loop through each line doing various tests/operations
> 2.4 Copy the fixed data into the float array in 2.1
>
> 3. Write the float array with 'openw' -> 'printf' -> 'close'
>
> Based on what I'm reading, it seems this should be more efficient by
> using 'read_ascii' and then fixing the errors with array indexing and
> then writing the data back. Is this worth pursuing, or are there any
> recommendations for a module I should know about?

Normally you step through as file like this reading a line at a time if
the problems with the file are more difficult to deal with than the
simple problems READ_ASCII can handle. We can't say without seeing the
file itself.

Cheers,

David

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

Subject: Re: operating on tab-/comma-delimited files
Posted by wlandsman on Wed, 28 Aug 2013 17:21:40 GMT
View Forum Message <> Reply to Message

I'd say the code is already efficient in that it reads and writes each file with a single I/O operation. Any place for improvement would be in step 2.3.

On Wednesday, August 28, 2013 12:56:27 PM UTC-4, Seb wrote:
> Hi,
>
>
>
> I am in the process of updating code I've inherited as I learn IDL, and
>
> noticed that it uses the following workflow for fixing errors in
>

> tab-/comma-delimited files:
>
>
>
> 1. Use file_search to generate a list of files to work on
>
>
>
> 2. Use FOR loop to iterate through each file
>
>
>
> 2.1 Create a float array with the same dimensions as input file
>
> 2.2 Load the data into a string array with 'openr' -> 'strarr' ->
>
> 'readf' -> 'close'
>
> 2.3 Loop through each line doing various tests/operations
>
> 2.4 Copy the fixed data into the float array in 2.1
>
>
>
> 3. Write the float array with 'openw' -> 'printf' -> 'close'
>
>
>
> Based on what I'm reading, it seems this should be more efficient by
>
> using 'read_ascii' and then fixing the errors with array indexing and
>
> then writing the data back. Is this worth pursuing, or are there any
>
> recommendations for a module I should know about?
>
>
>
> Thanks,
>
>
>
> --
>
> Seb