Subject: Writing a modified .txt file issue
Posted by Ahmet Hakan BAYRAK on Mon, 29 Jul 2013 08:22:06 GMT
View Forum Message <> Reply to Message

Hi everyone I started to learn IDL 2 weeks ago and I'm at zero level. :) I think my problem must be easy but I couldn't solve anyway. :(I hope I can express my problem properly. I have a .txt file like this:

hole_id from to Pr_1 Pr_2
2006/1 6 8.2 39.23 0.09
2006/1 12.25 14 39.16 12.06
2006/2A 2 5.9 59.17 0.42
2006/2A 5.9 8.7 51.02 0.61
2006/2A 8.7 11.7 49.8 0.97
. . . . .
. . . . .

These are my problems: I can't read the first column as string. With "file_lines" I can read other columns as floating array after seperated the first column on excel so I can seperate them in IDL and do calculations on these columns. At the beginning my header line is 5 columns but after my calculations, modifications my header line must be 7-8 columns. How can I do write in my output file? Also when I wrote the header and data they don't match, seems like this:

from to pr_1 pr_2 int
6.00000 8.20000 39.2300 0.0900000 2.20000
12.2500 14.0000 39.1600 12.0600 1.75000
. . . . .
. . . . .

(The fifth column is difference of "to" and "from" columns)

I am at the beginning on IDL and I want to learn please help me. Forgive me for my English.

Cheers,

Hakan

Subject: Re: Writing a modified .txt file issue
Posted by Andy Sayer on Mon, 29 Jul 2013 18:27:53 GMT
View Forum Message <> Reply to Message

If you post the code you have so far, we might be able to help.

On Monday, July 29, 2013 4:22:06 AM UTC-4, Ahmet Hakan BAYRAK wrote:
> Hi everyone I started to learn IDL 2 weeks ago and I'm at zero level. :) I think my problem must be easy but I couldn't solve anyway. :(I hope I can express my problem properly. I have a .txt file like this:

>
>
>
> hole_id from to Pr_1 Pr_2
>
> 2006/1 6 8.2 39.23 0.09
>
> 2006/1 12.25 14 39.16 12.06
>
> 2006/2A 2 5.9 59.17 0.42
>
> 2006/2A 5.9 8.7 51.02 0.61
>
> 2006/2A 8.7 11.7 49.8 0.97
>
> . . . . .
>
> . . . . .
>
>
>
> These are my problems: I can't read the first column as string. With "file_lines" I can read other columns as floating array after seperated the first column on excel so I can seperate them in IDL and do calculations on these columns. At the beginning my header line is 5 columns but after my calculations, modifications my header line must be 7-8 columns. How can I do write in my output file? Also when I wrote the header and data they don't match, seems like this:
>
>
>
> from to pr_1 pr_2 int
>
> 6.00000 8.20000 39.2300 0.0900000 2.20000
>
> 12.2500 14.0000 39.1600 12.0600 1.75000
>
> . . . . .
>
> . . . . .
>
>
>
> (The fifth column is difference of "to" and "from" columns)
>
>
>
> I am at the beginning on IDL and I want to learn please help me. Forgive me for my English.
>
>

>
> Cheers,
>
>
>
> Hakan

Subject: Re: Writing a modified .txt file issue
Posted by David Fanning on Mon, 29 Jul 2013 18:35:40 GMT
View Forum Message <> Reply to Message

Ahmet Hakan BAYRAK writes:

>
> Hi everyone I started to learn IDL 2 weeks ago and I'm at zero level. :) I think my problem must be easy but I couldn't solve anyway. :(I hope I can express my problem properly. I have a .txt file like this:
>
> hole_id from to Pr_1 Pr_2
> 2006/1 6 8.2 39.23 0.09
> 2006/1 12.25 14 39.16 12.06
> 2006/2A 2 5.9 59.17 0.42
> 2006/2A 5.9 8.7 51.02 0.61
> 2006/2A 8.7 11.7 49.8 0.97
> . . . . .
> . . . . .
>
> These are my problems: I can't read the first column as string. With "file_lines" I can read other columns as floating array after seperated the first column on excel so I can seperate them in IDL and do calculations on these columns. At the beginning my header line is 5 columns but after my calculations, modifications my header line must be 7-8 columns. How can I do write in my output file? Also when I wrote the header and data they don't match, seems like this:
>
> from to pr_1 pr_2 int
> 6.00000 8.20000 39.2300 0.0900000 2.20000
> 12.2500 14.0000 39.1600 12.0600 1.75000
> . . . . .
> . . . . .
>
> (The fifth column is difference of "to" and "from" columns)
>
> I am at the beginning on IDL and I want to learn please help me. Forgive me for my English.

The person who wrote this data file is not your friend. You should avoid
associating with them from now on.

They gave you something that may look easy, but is, in fact, very

difficult. Plus, you don't know much about IDL yet, so I would say this task is nearly impossible for you. I'm glad it is not the first thing I had to do when I was starting out learning IDL. (I had to read the entire User Guide--twice--when I was starting out before I could do something useful in IDL, but the User Guide was much smaller in those days.)

In any case, this is going to be difficult to explain to you, mostly because you probably don't have the IDL vocabulary and knowledge for us to communicate effectively with you. It might be easiest if you would post or send me the first 15 lines or so of this file, so I can write an example program that reads it. I'll post it on my web page for others to learn from as well.

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: Writing a modified .txt file issue
Posted by Phillip Bitzer on Mon, 29 Jul 2013 20:03:02 GMT
View Forum Message <> Reply to Message

I also have an ASCII data file that contains both numeric and string columns. I'm not a fan. What I'm doing now is read the whole thing in as a string array:

```
nLines = FILE_LINES(filename)
OPENR, lun, filename, /GET_LUN
word = STRARR(nLines)
READF, lun, word
FREE_LUN, lun
```

Then, I loop through each line, using STR_SPLIT to divvy up the fields at the space that separates the columns, and throw them into the output structure (not defined here for simplicity)

```
FOR i=0L, nLines-1 DO BEGIN
  this_row = STRSPLIT(word[i], ' ', /extract)
  data[i].date = this_row[0]
  data[i].time = this_row[1]
  data[i].latitude = DOUBLE(this_row[2])
  data[i].longitude = DOUBLE(this_row[3])
  ....
ENDFOR
```

I have been looking for a better/more efficient way of doing this. Can't seem to find a good way, but maybe Coyote has cooked up something :-) ReadCol works about as quickly as my code.

I should mention that there are several string columns mixed up amongst the numeric columns. Yuck.

For the output, Hakan, I would say those numbers *do* match, although they might be formatted differently. You'll want to use the FORMAT keyword, however you're writing the file back out if you want to match the numbers/decimal places/etc.

A simple example:
IDL> a = 13.2
IDL> PRINT, a
     13.2000
IDL> PRINT, a, format='(F5.2)'
13.20
IDL> PRINT, a, format='(F4.1)'
13.2

You would do something similar using PRINTF.

Subject: Re: Writing a modified .txt file issue
Posted by David Fanning on Mon, 29 Jul 2013 20:13:53 GMT
View Forum Message <> Reply to Message

Phillip Bitzer writes:

> I have been looking for a better/more efficient way of doing this. Can't seem to find a good way, but maybe Coyote has cooked up something :-) ReadCol works about as quickly as my code.

I think this is what you have to do. READCOL (a NASA Astronomy Library file) is as good as any when your data is in well defined columns and all the data is there. If data is missing then about the only thing you can do is get your cousin Vinny to go after the guy that created the file. READCOL is probably as fast as your routine, Philip, because it's doing what you are doing. :-)

Cheers,

David

P.S. I have to admit, I think I've used READCOL once in my life, but that's only because I'm stubborn and like to roll my own. :-)

--

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: Writing a modified .txt file issue
Posted by Heinz Stege on Mon, 29 Jul 2013 20:39:03 GMT
View Forum Message <> Reply to Message

On Mon, 29 Jul 2013 13:03:02 -0700 (PDT), Phillip Bitzer wrote:

> I also have an ASCII data file that contains both numeric and string columns. I'm not a fan. What I'm doing now is read the whole thing in as a string array:
>
> nLines = FILE_LINES(filename)
> OPENR, lun, filename, /GET_LUN
> word = STRARR(nLines)
> READF, lun, word
> FREE_LUN, lun
>
> Then, I loop through each line, using STR_SPLIT to divvy up the fields at the space that separates the columns, and throw them into the output structure (not defined here for simplicity)
>
> FOR i=0L, nLines-1 DO BEGIN
> this_row = STRSPLIT(word[i], ' ', /extract)
> data[i].date = this_row[0]
> data[i].time = this_row[1]
> data[i].latitude = DOUBLE(this_row[2])
> data[i].longitude = DOUBLE(this_row[3])
> ....
> ENDFOR
>
> I have been looking for a better/more efficient way of doing this. Can't seem to find a good way, but maybe Coyote has cooked up something :-) ReadCol works about as quickly as my code.
>
This type of files typically have much more lines than columns.
Therefore it is more efficient to run the loop over a few columns than
over all the lines. The following code more deals with arrays instead
of scalars:

```
for i=0,nColumns-2 do begin
  pos=strpos(word,' ')
  data.(i)=strmid(word,0,transpose(pos))
  word=strtrim(strmid(word,transpose(pos)),1)
  end
data.(nColumns-1)=word
```

Try it.

Cheers, Heinz

Subject: Re: Writing a modified .txt file issue
Posted by wlandsman on Mon, 29 Jul 2013 20:55:21 GMT
View Forum Message <> Reply to Message

I haven't really tested this, but now STRSPLIT() can operate on a 2-d array and return the values in a list. So you can eliminate loops entirely

```
mylist = strsplit(word,' ',/extract)
myarr = mylist.toarray()
data.date = myarr[*,0]
data.time = myarr[*,1]
data.latitude = double( myarr[*,2])
data.longitude = double( myarr[*,3])
```

On Monday, July 29, 2013 4:03:02 PM UTC-4, Phillip Bitzer wrote:

```
>    FOR i=0L, nLines-1 DO BEGIN
>
>      this_row = STRSPLIT(word[i], ' ', /extract)
>
>      data[i].date = this_row[0]
>
>      data[i].time = this_row[1]
>
>      data[i].latitude = DOUBLE(this_row[2])
>
>      data[i].longitude = DOUBLE(this_row[3])
>
>      ....
>
>    ENDFOR
>
>
>
>  I have been looking for a better/more efficient way of doing this. Can't seem to find a good way, but maybe Coyote has cooked up something :-) ReadCol works about as quickly as my code.
>
>
>
>  I should mention that there are several string columns mixed up amongst the numeric columns. Yuck.
>
>
```

>
> For the output, Hakan, I would say those numbers *do* match, although they might be formatted differently. You'll want to use the FORMAT keyword, however you're writing the file back out if you want to match the numbers/decimal places/etc.
>
>
>
> A simple example:
>
> IDL> a = 13.2
>
> IDL> PRINT, a
>
>   13.2000
>
> IDL> PRINT, a, format='(F5.2)'
>
> 13.20
>
> IDL> PRINT, a, format='(F4.1)'
>
> 13.2
>
>
>
> You would do something similar using PRINTF.

Subject: Re: Writing a modified .txt file issue
Posted by David Fanning on Mon, 29 Jul 2013 22:31:15 GMT
View Forum Message <> Reply to Message

Heinz Stege writes:

Uh, let me remind you, this guy said he was an IDL "beginner". You guys
are losing me. :-(

Cheers,

David

>
> On Mon, 29 Jul 2013 13:03:02 -0700 (PDT), Phillip Bitzer wrote:
>
>> I also have an ASCII data file that contains both numeric and string columns. I'm not a fan. What I'm doing now is read the whole thing in as a string array:
>>
>>   nLines = FILE_LINES(filename)

>> OPENR, lun, filename, /GET_LUN
>> word = STRARR(nLines)
>> READF, lun, word
>> FREE_LUN, lun
>>
>> Then, I loop through each line, using STR_SPLIT to divvy up the fields at the space that separates the columns, and throw them into the output structure (not defined here for simplicity)
>>
>> FOR i=0L, nLines-1 DO BEGIN
>>   this_row = STRSPLIT(word[i], ' ', /extract)
>>   data[i].date = this_row[0]
>>   data[i].time = this_row[1]
>>   data[i].latitude = DOUBLE(this_row[2])
>>   data[i].longitude = DOUBLE(this_row[3])
>>   ....
>> ENDFOR
>>
>> I have been looking for a better/more efficient way of doing this. Can't seem to find a good way, but maybe Coyote has cooked up something :-) ReadCol works about as quickly as my code.
>>
> This type of files typically have much more lines than columns.
> Therefore it is more efficient to run the loop over a few columns than
> over all the lines. The following code more deals with arrays instead
> of scalars:
>
> for i=0,nColumns-2 do begin
>   pos=strpos(word,' ')
>   data.(i)=strmid(word,0,transpose(pos))
>   word=strtrim(strmid(word,transpose(pos)),1)
>   end
> data.(nColumns-1)=word
>
> Try it.
>
> Cheers, Heinz

--
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: Writing a modified .txt file issue

Posted by David Fanning on Mon, 29 Jul 2013 23:18:31 GMT
View Forum Message <> Reply to Message

Wayne Landsman writes:

> I haven't really tested this, but now STRSPLIT() can operate on a 2-d array and return the values in a list. So you can eliminate loops entirely
>
> mylist = strsplit(word,' ',/extract)
> myarr = mylist.toarray()
> data.date = myarr[*,0]
> data.time = myarr[*,1]
> data.latitude = double(myarr[*,2])
> data.longitude = double(myarr[*,3])

OK, I can get this to work if I change this line:

```
  mylist = strsplit(word,' ',/extract)
```

To this:

```
  mylist = strsplit(word,/extract)
```

I can't get this code by Heinz to work at all:

```
> for i=0,nColumns-2 do begin
>    pos=strpos(word,' ')
>    data.(i)=strmid(word,0,transpose(pos))
>    word=strtrim(strmid(word,transpose(pos)),1)
>   end
> data.(nColumns-1)=word
```

Does anyone have any less theoretical code? :-)

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: Writing a modified .txt file issue
Posted by Phillip Bitzer on Tue, 30 Jul 2013 00:07:40 GMT
View Forum Message <> Reply to Message

>
> Does anyone have any less theoretical code? :-)
>

Working on it.... Initial testing shows Heinz's implementation is about 33% faster. Trying to squeeze a little more out of it.

Subject: Re: Writing a modified .txt file issue
Posted by Heinz Stege on Tue, 30 Jul 2013 00:54:20 GMT
View Forum Message <> Reply to Message

On Mon, 29 Jul 2013 17:18:31 -0600, David Fanning wrote:

> I can't get this code by Heinz to work at all:
>

Below is an example with some Explanations.

Cheers, Heinz

```
; Here is a string array which could come from a file:
nLines=10
word='text'+strtrim(indgen(nLines)+1,2)+$
    string(findgen(nLines)*2.+100.)+$
    string(lindgen(nLines)*5L+200L)
print,'This is our input:'
print,word,form='(a)'
;
; Make a structure for the result:
nColumns=3
struct={text:'',float:0.,long:0L}
data=replicate(struct,nLines)
;
; Loop over all columns but the last one:
for i=0,nColumns-2 do begin
  ;
  ; Get the end positions of the first column
  pos=strpos(word,' ')
  ;
  ; Put the first column into the i-th tag of the structure
  data.(i)=strmid(word,0,transpose(pos))
  ;
  ; Remove the first column (and trailing blanks) from the input
  ; array
  word=strtrim(strmid(word,transpose(pos)),1)
  ;
```

```
   ; Continue the loop with the next column
   end
;
; Copy the last column into the structure
data.(nColumns-1)=word
;
; Check the result
print,'This is the result structure:'
for i=0,nLines-1 do print,data[i]
;
end
```

Subject: Re: Writing a modified .txt file issue
Posted by David Fanning on Tue, 30 Jul 2013 01:39:49 GMT
View Forum Message <> Reply to Message

Heinz Stege writes:

> Below is an example with some Explanations.

OK, here is my data file, save in a file bad_data.txt. If you put this
in a text editor, you will see the columns are all left aligned.

```
hole_id  from to Pr_1 Pr_2
2006/1  6 8.2 39.23 0.09
2006/1  12.25 14 39.16 12.06
2006/2A  2 5.9 59.17 0.42
2006/2A  5.9 8.7 51.02 0.61
2006/2A  8.7 11.7 49.8 0.97
```

Here is the code I am using. A combination of Philip's code and yours,
modified a bit to strip the header off.

```
;************************************************
nLines = FILE_LINES(filename)
OPENR, lun, filename, /GET_LUN
header = ""
word = STRARR(nLines-1)
READF, lun, header, word
FREE_LUN, lun

mylist = strsplit(word,/extract)
myarr = mylist.toarray()

struct = {date:"", from:0.0, to:0.0, pr1:0.0, pr2:0.0}
data = Replicate(struct, nLines-1)
```

```
; Loop over all columns but the last one:
for i=0,nColumns-2 do begin
   ;
   ; Get the end positions of the first column
   pos=strpos(word,' ')
   ;
   ; Put the first column into the i-th tag of the structure
   data.(i)=strmid(word,0,transpose(pos))
   ;
   ; Remove the first column (and trailing blanks) from the input
   ; array
   word=strtrim(strmid(word,transpose(pos)),1)
   ;
   ; Continue the loop with the next column
   end
;
; Copy the last column into the structure
data.(nColumns-1)=word
;
; Check the result
print,'This is the result structure:'
for i=0,nLines-2 do print,data[i]
;

END
;************************************************
```

Here are my results when I run the program:

```
This is the result structure:
{ 2006/1      0.000000     0.000000     0.000000      6.00000}
{ 2006/1      0.000000     0.000000     0.000000      12.2500}
{ 2006/2A      0.000000     0.000000     0.000000      2.00000}
{ 2006/2A      0.000000     0.000000     0.000000      5.90000}
{ 2006/2A      0.000000     0.000000     0.000000      8.70000}
```

What am I doing wrong?

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: Writing a modified .txt file issue
Posted by wlandsman on Tue, 30 Jul 2013 02:18:04 GMT
View Forum Message <> Reply to Message

Hmm, I just cut and pasted your code and got the expected output structure corresponding to the input file. The only change I had to make was to define Ncolumns which isn't defined in your code snippet. Perhaps you had an incorrect value of Ncolumns stored somewhere?

A few words here about READCOL. There are dozens of IDL programs to read ASCII files, and READCOL is certainly not the fastest in computer speed. But I think it requires the least number of brain cells. Here is the READCOL method to read the data file:

IDL> readcol, 'test.dat', hole_id, vfrom, vto, pr_1, pr_2,f='a'

Note that (1) one doesn't need to predefine any of the variables. (2) One needs to specify the column to be read as a string, but no specification is needed for the floating point columns (3) One doesn't need to worry about the header line (any line not meeting the specified format is ignored). Also, READCOL can handle missing columns with the use of the /PRESERVE_NULL keyword.

--Wayne

> Here are my results when I run the program:
>
>
>
> This is the result structure:
>
> { 2006/1 0.000000 0.000000 0.000000 6.00000}
>
> { 2006/1 0.000000 0.000000 0.000000 12.2500}
>
> { 2006/2A 0.000000 0.000000 0.000000 2.00000}
>
> { 2006/2A 0.000000 0.000000 0.000000 5.90000}
>
> { 2006/2A 0.000000 0.000000 0.000000 8.70000}
>
>
>
> What am I doing wrong?
>
>
>
> 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: Writing a modified .txt file issue
Posted by David Fanning on Tue, 30 Jul 2013 02:31:35 GMT
View Forum Message <> Reply to Message

Wayne Landsman writes:

> Hmm, I just cut and pasted your code and got the expected output structure corresponding to the input file. The only change I had to make was to define Ncolumns which isn't defined in your code snippet. Perhaps you had an incorrect value of Ncolumns stored somewhere?

Really!? I added in ncolumns=5 (which was defined from before, but I put it in the code anyway). I get exactly the same results as before. This is really very weird. In your data file, are your columns LEFT justified?

OK, now I'm pissed. Here is the data file I used, and the program I used.

http://www.idlcoyote.com/misc/read_bad_data.zip

And, again, here are my results:

```
IDL> .go
This is the result structure:
{ 2006/1      0.000000     0.000000     0.000000      6.00000}
{ 2006/1      0.000000     0.000000     0.000000      12.2500}
{ 2006/2A     0.000000     0.000000     0.000000      2.00000}
{ 2006/2A     0.000000     0.000000     0.000000      5.90000}
{ 2006/2A     0.000000     0.000000     0.000000      8.70000}
```

This is IDL 8.2.3 on Windows.

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: Writing a modified .txt file issue
Posted by Heinz Stege on Tue, 30 Jul 2013 04:42:58 GMT
View Forum Message <> Reply to Message

Good Morning David,

the reason is, that there are tabs (9b) instead of blanks (32b) in
your text file. Only the first two columns are separated by a tab
which is followed by a blank.

That makes pos=strpos(word,' ') to be -1 after extracting the first
column. That, what your eyes see as column 2 to 5 is not splitted
within the loop. It remains as one column in the word array.

When IDL converts this strings to floating point, it quietly ignores
all the stuff right of the first tab. For example
```
  help,float('12.25 14 39.16 12.06')
```
prints
```
  <Expression>    FLOAT     =       12.2500
```
on the screen.

Do you have a string-replace-routine within your coyote library to
replace all tabs within a string array by blanks? If yes, use it.

If not, you can replace
```
  pos=strpos(word,' ')
```
in line 21 by
```
  pos=ulong(strpos(word,' ')) < ulong(strpos(word,string(9b)))
```
and you will get the expected result. But please, don't blame me to
make cryptic code. ;-)

The ulong function above is used to change -1 to a very big number.
The replacement line returns the position of the next blank or the
next tab, depending on what is more left.

Cheers, Heinz

Subject: Re: Writing a modified .txt file issue
Posted by Ahmet Hakan BAYRAK on Tue, 30 Jul 2013 06:49:58 GMT
View Forum Message <> Reply to Message

Thank you all for the answers. My aim is to plot histograms for differences of height of drilling data and parameters data each other. This is I can get so far. I remove the header part to not confuse you because it makes dirty all code. :)

```
pro hist

infile='drilling.txt'
n=FILE_LINES(infile)
a=fltarr(n)
b=fltarr(n)
c=fltarr(n)
d=fltarr(n)
a1=0.0
b1=0.0
c1=0.0
d1=0.0
openr, iunit, infile, /GET_LUN
FOR i=0, n-1 DO BEGIN
readf, iunit, a1, b1, c1, d1
a[i]=a1
b[i]=b1
c[i]=c1
d[i]=d1
ENDFOR
FREE_LUN, iunit
FOR i=0, n-1 DO BEGIN
openw, lun, 'output.txt', /GET_LUN
a2=TRANSPOSE(a)
b2=TRANSPOSE(b)
c2=TRANSPOSE(c)
d2=TRANSPOSE(d)
int=b2-a2
g=[a2, b2, c2, d2, int]
FORMAT="(2F12.5)"
PRINTF, lun, g
FREE_LUN, lun
END
END
```

This is my input file: drilling.txt. It has 770 lines with header:

```
hole_id from to Pr_1 Pr_2
2006/1 6 8.2 39.23 0.09
2006/1 12.25 14 39.16 12.06
2006/2A 2 5.9 59.17 0.42
```

2006/2A 5.9 8.7 51.02 0.61
2006/2A 8.7 11.7 49.8 0.97
2006/2A 13.05 14.5 38.18 0.05
2006/2A 14.9 18 38.31 0.47
2006/2A 21.75 23.2 40.89 0.3
2006/3A 5.75 7.8 39.89 0.63
2006/3A 8.7 11.5 44.87 0.16
2006/3A 34 36.2 29.76 0.27
2006/3A 36.2 38.9 56.35 0.67
2006/3A 53.25 54.7 31.61 4.16
2006/3A 54.95 55.9 55.51 2.56
2006/4 38.2 39.6 29.38 5.5

And this is my output file: output.txt: In the output file I didn't remove header to show how it doesn't match.

from to pr_1 pr_2 int

6.00000	8.20000	39.2300	0.0900000	2.20000
12.2500	14.0000	39.1600	12.0600	1.75000
2.00000	5.90000	59.1700	0.420000	3.90000
5.90000	8.70000	51.0200	0.610000	2.80000
8.70000	11.7000	49.8000	0.970000	3.00000
13.0500	14.5000	38.1800	0.0500000	1.45000
14.9000	18.0000	38.3100	0.470000	3.10000
21.7500	23.2000	40.8900	0.300000	1.45000
5.75000	7.80000	39.8900	0.630000	2.05000
8.70000	11.5000	44.8700	0.160000	2.80000
34.0000	36.2000	29.7600	0.270000	2.20000
36.2000	38.9000	56.3500	0.670000	2.70000
53.2500	54.7000	31.6100	4.16000	1.45000
54.9500	55.9000	55.5100	2.56000	0.950001
38.2000	39.6000	29.3800	5.50000	1.40000
39.6000	40.4000	32.1900	3.96000	0.800003

Thank you for now,

Hakan

Subject: Re: Writing a modified .txt file issue
Posted by wlandsman on Tue, 30 Jul 2013 10:38:35 GMT
View Forum Message <> Reply to Message

On Tuesday, July 30, 2013 2:49:58 AM UTC-4, Ahmet Hakan BAYRAK wrote:

> And this is my output file: output.txt: In the output file I didn't remove header to show how it doesn't match.

Why don't you think that they match? Is it because the output file has numbers like 8.2000 while the input file has 8.2? You gave an output format of F12.5 which is why the output has 5 digits after the decimal point. As Phillip wrote earlier

****
For the output, Hakan, I would say those numbers *do* match, although they might be formatted differently. You'll want to use the FORMAT keyword, however you're writing the file back out if you want to match the numbers/decimal places/etc.

A simple example:
```
IDL> a = 13.2
IDL> PRINT, a
      13.2000
IDL> PRINT, a, format='(F5.2)'
13.20
IDL> PRINT, a, format='(F4.1)'
13.2
```

You would do something similar using PRINTF.
****

Or is there some other reason why you think that the output doesn't match the input?

Subject: Re: Writing a modified .txt file issue
Posted by Ahmet Hakan BAYRAK on Tue, 30 Jul 2013 11:02:51 GMT
View Forum Message <> Reply to Message

> Or is there some other reason why you think that the output doesn't match the input?

There is no problem matching between input file and output file for me. I tried to say in my output file my header and my data are not one under the other. I mean "from" is not top of the first column, "to" is not top of the second column and goes on like that.

Subject: Re: Writing a modified .txt file issue
Posted by David Fanning on Tue, 30 Jul 2013 12:43:09 GMT
View Forum Message <> Reply to Message

Heinz Stege writes:

> the reason is, that there are tabs (9b) instead of blanks (32b) in
> your text file. Only the first two columns are separated by a tab
> which is followed by a blank.

Well, I took the file from the person asking the questions.
I've already pointed out that the person giving him the file should have

all his fingers broken. Here is another reason to think so. :-)

> That makes pos=strpos(word,' ') to be -1 after extracting the first
> column. That, what your eyes see as column 2 to 5 is not splitted
> within the loop. It remains as one column in the word array.

Ah, this is probably also why I had to modify Wayne's code to get it to work. I suppose this is just another way of endorsing READCOL, since that reads the file correctly. It probably works because it has been around long enough to have seen all the perversion in the world. ;-)

> But please, don't blame me to make cryptic code. ;-)

No, no. You weren't the one to create lists and hashes and the _OVERLOADBRACKETSLEFTSIDE operator. I'm sure you had nothing to do with cryptic code. :-)

Thanks for your help!

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: Writing a modified .txt file issue
Posted by Andy Sayer on Tue, 30 Jul 2013 15:17:39 GMT
View Forum Message <> Reply to Message

So are you saying you want the words in the first column to align with the numbers underneath them? If that's it, you may need to use a different format code for the first line (i=0 in your loop). For example:

```
IDL> print,'b',format='(A15)'
           b
IDL> print,'b',format='(A10)'
       b
IDL>
```

Here 'A' is for formatting characters, and 15 and 10 are the 'width' to set the output to.

Based on a glance at your output it looks like yours are 13 characters wide, so give format='(A13)'

a shot.

Hope this helps,

Andy

On Tuesday, July 30, 2013 7:02:51 AM UTC-4, Ahmet Hakan BAYRAK wrote:
>
> There is no problem matching between input file and output file for me. I tried to say in my output file my header and my data are not one under the other. I mean "from" is not top of the first column, "to" is not top of the second column and goes on like that.
