

---

Subject: Re: IDL input files.

Posted by [David Fanning](#) on Mon, 18 Jun 2007 17:38:59 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

ryanselk@gmail.com writes:

> ive been reading some manuals and books for IDL but im not sure how to  
> solve this likely simple problem.

Oh, dear. Which books have you been reading!?

You need to explain your problem a little bit better for us. Your code doesn't work for many reasons, but one big reason is that you are trying to read from a file you opened for writing. And I can't tell from your example, what it is you are trying to do.

Do you wish to *create* a file (OPENW and PRINTF), or do you wish to *read* from a file (OPENR and READF)?

Or, do you wish to ask the user a question (maybe you need [http://www.dfanning.com/widget\\_tips/popup.html](http://www.dfanning.com/widget_tips/popup.html)) and then write the answers in a file?

I'm very confused by your question. Do you think you could clarify for us?

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. ("Perhaps thou speakest truth.")

---

---

Subject: Re: IDL input files.

Posted by [ryanselk](#) on Mon, 18 Jun 2007 18:35:36 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On Jun 18, 11:38 am, David Fanning <n...@dfanning.com> wrote:

> ryans...@gmail.com writes:

>> ive been reading some manuals and books for IDL but im not sure how to  
>> solve this likely simple problem.

>

> Oh, dear. Which books have you been reading!?

>

> You need to explain your problem a little bit better  
> for us. Your code doesn't work for many reasons, but  
> one big reason is that you are trying to read from  
> a file you opened for writing. And I can't tell from  
> your example, what it is you are trying to do.  
>  
> Do you wish to \*create\* a file (OPENW and PRINTF), or  
> do you wish to \*read\* from a file (OPENR and READF)?  
>  
> Or, do you wish to ask the user a question (maybe you  
> need [http://www.dfanning.com/widget\\_tips/popup.html](http://www.dfanning.com/widget_tips/popup.html))  
> and then write the answers in a file?  
>  
> I'm very confused by your question. Do you think you  
> could clarify for us?  
>  
> 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. ("Perhaps thou speakest truth.")

Whoops, it's just reading from a file, I messed with it earlier and forgot to change it back. It should be:

```
PRO inputer
  print, 'starting input'

  getfile = DIALOG_PICKFILE (/READ)

  print, getfile

  OPENR, 1, getfile
  POINT_LUN, 1, 0

  IDL_Name = '          ' ; want extendable length if possible.

  READU, 1, IDL_Name

  print, IDL_Name

  POINT_LUN, 1, 0
  CLOSE, 1
```

END

Ideally,

This idl file just reads whatever file is selected and stores the name variable to IDL\_Name .

But what I want to do is 'search' the file opened for certain words, when it finds them I want to be able to store that value to an IDL variable.

Im going to need to declare more variables then IDL\_Name, but im unsure how to do it.

ie: if the user selects this file:

```
Name = {Joe}
Department = {CS}
Age = {25}
```

I want in IDL to have those 3 variables stored ie:

```
IDL_Name = Joe
IDL_DEPARTMENT = CS
IDL_AGE = 25
```

But, if the file reads:

```
Age = {30}
Department = {Science}
```

I would want IDL to return;

```
IDL_DEPARTMENT = Science
IDL_Age = 30
```

I think i need to use some sort of 'search file' command. I hope this is clearer, sorry for being unclear before.

Thank you! and David your website has been of help to me frequently in the past.

---

Subject: Re: IDL input files.

Posted by [ryanselk](#) on Mon, 18 Jun 2007 18:40:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On Jun 18, 12:35 pm, ryans...@gmail.com wrote:

> On Jun 18, 11:38 am, David Fanning <n...@dfanning.com> wrote:

```

>
>
>
>> ryans...@gmail.com writes:
>>> ive been reading some manuals and books for IDL but im not sure how to
>>> solve this likely simple problem.
>
>> Oh, dear. Which books have you been reading!?!
>
>> You need to explain your problem a little bit better
>> for us. Your code doesn't work for many reasons, but
>> one big reason is that you are trying to read from
>> a file you opened for writing. And I can't tell from
>> your example, what it is you are trying to do.
>
>> Do you wish to *create* a file (OPENW and PRINTF), or
>> do you wish to *read* from a file (OPENR and READF)?
>
>> Or, do you wish to ask the user a question (maybe you
>> need http://www.dfanning.com/widget\_tips/popup.html)
>> and then write the answers in a file?
>
>> I'm very confused by your question. Do you think you
>> could clarify for us?
>
>> 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. ("Perhaps thou speakest truth.")
>
> Whoops, its just reading from a file, I messed with it earlier and
> forgot to change it back, It should be:
>
> PRO inputer
>     print, 'starting input'
>
>     getfile = DIALOG_PICKFILE (/READ)
>
>         print, getfile
>
>         OPENR, 1, getfile
>         POINT_LUN, 1, 0
>
>         IDL_Name = '          ' ; want extendable length if possible.

```

```
>
>     READU, 1, IDL_Name
>
>     print, IDL_Name
>
>     POINT_LUN, 1, 0
>     CLOSE, 1
>
> END
>
> Ideally,
> This idl file just reads whatever file is selected and stores the name
> variable to IDL_Name .
> But what I want to do is 'search' the file opened for certain words,
> when it finds them I want to be able to store that value to an IDL
> variable.
> Im going to need to declare more variables then IDL_Name, but im
> unsure how to do it.
>
> ie: if the user selects this file:
>
> Name = {Joe}
> Department = {CS}
> Age = {25}
>
> I want in IDL to have those 3 variables stored ie:
>
> IDL_Name = Joe
> IDL_DEPARTMENT = CS
> IDL_AGE = 25
>
> But, if the file reads:
>
> Age = {30}
> Department = {Science}
>
> I would want IDL to return;
>
> IDL_DEPARTMENT = Science
> IDL_Age = 30
>
> I think i need to use some sort of 'search file' command. I hope this
> is clearer, sorry for being unclear before.
>
> Thank you! and David your website has been of help to me frequently in
> the past.
```

Crap, again, i messed it up. It should be a readf not a readu.

---

---

Subject: Re: IDL input files.

Posted by [Paul Van Delst\[1\]](#) on Mon, 18 Jun 2007 18:41:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

ryanselk@gmail.com wrote:

> ive been reading some manuals and books for IDL but im not sure how to  
> solve this likely simple problem.

>

> I have this I need to input to idl (.txt file):

>

> Name = {Joe}

> Department = {CS}

> Age = {25}

>

>

> and in IDL i need to save each attribute to its own variable. But if,

> lets say, department is omitted I want it to not input age as

> department. With c/c++ this isnt so bad as you can search the file for

> a 'keyword' and print the value from there, but I cant find how to do

> this in IDL.

Well, I don't think you can search the file in IDL, but you can read it in line by line and search the lines. Why not create the file as IDL commands? e.g.:

pro test

```
; Create file to read
openw, lun, 'test.input', /get_lun
printf, lun, 'Name = "Joe"'
printf, lun, 'Department = "CS"'
printf, lun, 'Age = 25'
free_lun, lun
```

```
; Now read the file
openr, lun, 'test.input', /get_lun
buffer = ''
while not eof(lun) do begin
  readf, lun, buffer
  result = execute(buffer)
endwhile
free_lun, lun
help, name, department, age
```

end

??

IDL> .run test

```
% Compiled module: TEST.
IDL> test
NAME      STRING  = 'Joe'
DEPARTMENT STRING  = 'CS'
AGE       INT    =   25
```

```
IDL> $more test.input
Name = "Joe"
Department = "CS"
Age = 25
```

--

Paul van Delst            Ride lots.  
CIMSS @ NOAA/NCEP/EMC            Eddy Merckx

---

---

Subject: Re: IDL input files.

Posted by [David Fanning](#) on Mon, 18 Jun 2007 19:15:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Paul van Delst writes:

```
> ryanselk@gmail.com wrote:
>> ive been reading some manuals and books for IDL but im not sure how to
>> solve this likely simple problem.
>>
>> I have this I need to input to idl (.txt file):
>>
>> Name = {Joe}
>> Department = {CS}
>> Age = {25}
>>
>>
>> and in IDL i need to save each attribute to its own variable. But if,
>> lets day, department is ommited I want it to not input age as
>> department. With c/c++ this isnt so bad as you can search the file for
>> a 'keyword' and print the value from there, but I cant find how to do
>> this in IDL.
>
> Well, I don't think you can search the file in IDL, but you can read it in line by line
> and search the lines. Why not create the file as IDL commands? e.g.:
>
>
> ??
>
>
> IDL> .run test
```

```

> % Compiled module: TEST.
> IDL> test
> NAME      STRING  = 'Joe'
> DEPARTMENT STRING  = 'CS'
> AGE       INT     =   25
>
> IDL> $more test.input
> Name = "Joe"
> Department = "CS"
> Age = 25

```

Well, I'd do this a little differently:

```

pro test, name, dept, age

; Create file to read
openw, lun, 'test.input', /get_lun
if n_elements(name) NE 0 then $
  printf, lun, 'Name = ' + name else $
  printf, lun, 'Name = '
if n_elements(dept) NE 0 then $
  printf, lun, 'Department = ' + dept else $
  printf, lun, 'Department = '
if n_elements(age) NE 0 then $
  printf, lun, 'Age = ' + StrTrim(age,2) else $
  printf, lun, 'Age = '

free_lun, lun

; Now read the file
openr, lun, 'test.input', /get_lun
buffer = ''
while not eof(lun) do begin
  readf, lun, buffer
  parts = StrSplit(buffer, " ", /Extract)
  case n_elements(parts) of
    2: print, 'No value for ' + parts[0]
    3: print, parts[0] + '= {' + parts[2] + '}'
    else: print, 'Whoa, I am like totally confused!!'
  endcase
endwhile
free_lun, lun
end

```

Then try it like this:

```

IDL> test, 'coyote', 'PE', 43
Name= {coyote}

```

Department= {PE}  
Age= {43}

```
IDL> test
No value for Name
No value for Department
No value for Age
```

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. ("Perhaps thou speakest truth.")

---

---

Subject: Re: IDL input files.  
Posted by [ryanselk](#) on Mon, 18 Jun 2007 19:59:14 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Jun 18, 1:15 pm, David Fanning <n...@dfanning.com> wrote:  
> Paul van Delst writes:  
>> ryans...@gmail.com wrote:  
>>> ive been reading some manuals and books for IDL but im not sure how to  
>>> solve this likely simple problem.  
>  
>>> I have this I need to input to idl (.txt file):  
>  
>>> Name = {Joe}  
>>> Department = {CS}  
>>> Age = {25}  
>  
>>> and in IDL i need to save each attribute to its own variable. But if,  
>>> lets say, department is omitted I want it to not input age as  
>>> department. With c/c++ this isnt so bad as you can search the file for  
>>> a 'keyword' and print the value from there, but I cant find how to do  
>>> this in IDL.  
>  
>> Well, I don't think you can search the file in IDL, but you can read it in line by line  
>> and search the lines. Why not create the file as IDL commands? e.g.:  
>  
>> ??  
>  
>> IDL> .run test  
>> % Compiled module: TEST.  
>> IDL> test

```

>> NAME      STRING  = 'Joe'
>> DEPARTMENT STRING  = 'CS'
>> AGE       INT     =   25
>
>> IDL> $more test.input
>> Name = "Joe"
>> Department = "CS"
>> Age = 25
>
> Well, I'd do this a little differently:
>
>   pro test, name, dept, age
>
>       ; Create file to read
>       openw, lun, 'test.input', /get_lun
>       if n_elements(name) NE 0 then $
>         printf, lun, 'Name = ' + name else $
>         printf, lun, 'Name = '
>       if n_elements(dept) NE 0 then $
>         printf, lun, 'Department = ' + dept else $
>         printf, lun, 'Department = '
>       if n_elements(age) NE 0 then $
>         printf, lun, 'Age = ' + StrTrim(age,2) else $
>         printf, lun, 'Age = '
>
>       free_lun, lun
>
>       ; Now read the file
>       openr, lun, 'test.input', /get_lun
>       buffer = ''
>       while not eof(lun) do begin
>         readf, lun, buffer
>         parts = StrSplit(buffer, " ", /Extract)
>         case n_elements(parts) of
>           2: print, 'No value for ' + parts[0]
>           3: print, parts[0] + '= {' + parts[2] + '}'
>           else: print, 'Whoa, I am like totally confused!!!'
>         endcase
>       endwhile
>       free_lun, lun
>     end
>
> Then try it like this:
>
>   IDL> test, 'coyote', 'PE', 43
>     Name= {coyote}
>     Department= {PE}
>     Age= {43}

```

```
>
> IDL> test
>   No value for Name
>   No value for Department
>   No value for Age
>
> 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. ("Perhaps thou speakest truth.")
```

Thank you guys! these should help, although I cant have IDL writing the program unfortunately. Ill have to figure out how to get the pre-made files read.

---

---

Subject: Re: IDL input files.  
Posted by [Paul Van Delst\[1\]](#) on Mon, 18 Jun 2007 21:23:43 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

```
ryanselk@gmail.com wrote:
> On Jun 18, 1:15 pm, David Fanning <n...@dfanning.com> wrote:
>> Paul van Delst writes:
>>> ryans...@gmail.com wrote:
>>>> ive been reading some manuals and books for IDL but im not sure how to
>>>> solve this likely simple problem.
>>>> I have this I need to input to idl (.txt file):
>>>> Name = {Joe}
>>>> Department = {CS}
>>>> Age = {25}
>>>> and in IDL i need to save each attribute to its own variable. But if,
>>>> lets day, department is ommited I want it to not input age as
>>>> department. With c/c++ this isnt so bad as you can search the file for
>>>> a 'keyword' and print the value from there, but I cant find how to do
>>>> this in IDL.
>>> Well, I don't think you can search the file in IDL, but you can read it in line by line
>>> and search the lines. Why not create the file as IDL commands? e.g.:
>>> ??
>>> IDL> .run test
>>> % Compiled module: TEST.
>>> IDL> test
>>> NAME          STRING   = 'Joe'
>>> DEPARTMENT    STRING   = 'CS'
>>> AGE           INT      =   25
```

```

>>> IDL> $more test.input
>>> Name = "Joe"
>>> Department = "CS"
>>> Age = 25
>> Well, I'd do this a little differently:
>>
>>   pro test, name, dept, age
>>
>>   ; Create file to read
>>   openw, lun, 'test.input', /get_lun
>>   if n_elements(name) NE 0 then $
>>     printf, lun, 'Name = ' + name else $
>>     printf, lun, 'Name = '
>>   if n_elements(dept) NE 0 then $
>>     printf, lun, 'Department = ' + dept else $
>>     printf, lun, 'Department = '
>>   if n_elements(age) NE 0 then $
>>     printf, lun, 'Age = ' + StrTrim(age,2) else $
>>     printf, lun, 'Age = '
>>
>>   free_lun, lun
>>
>>   ; Now read the file
>>   openr, lun, 'test.input', /get_lun
>>   buffer = ''
>>   while not eof(lun) do begin
>>     readf, lun, buffer
>>     parts = StrSplit(buffer, " ", /Extract)
>>     case n_elements(parts) of
>>       2: print, 'No value for ' + parts[0]
>>       3: print, parts[0] + '= {' + parts[2] + '}'
>>       else: print, 'Whoa, I am like totally confused!!'
>>     endcase
>>   endwhile
>>   free_lun, lun
>> end
>>
>> Then try it like this:
>>
>> IDL> test, 'coyote', 'PE', 43
>>   Name= {coyote}
>>   Department= {PE}
>>   Age= {43}
>>
>> IDL> test
>>   No value for Name
>>   No value for Department
>>   No value for Age

```

>>  
>> 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. ("Perhaps thou speakest truth.")  
>  
> Thank you guys! these should help, although I cant have IDL writing  
> the program unfortunately. Ill have to figure out how to get the pre-  
> made files read.

Um, that's what these examples do. The writing of the test input files was simply to provide an entirely self-contained example. The various details that aren;t addressed are the formats of the files. If you can't create them (i.e. change their formats to make them more "IDL friendly" as it were), then all that needs to be modified in both David's and my examples is how you extract the relevant information from the string you read from file (i.e. the STRSPLIT.)

On a side note, I initially started using regular expressions with the STRSPLIT function only to discover IDL seems to use a hamstrung version that doesn't understand stuff like \s, \w, \n, etc. For the format you list, e.g.

```
Name = {Joe}
Department = {CS}
Age = {25}
```

regexps would be the go (IMO) - although I don't see any mention of how you would capture certain parts of a match and not others (but I didn't look too hard :o)

cheers,

paulv

--

Paul van Delst            Ride lots.  
CIMSS @ NOAA/NCEP/EMC            Eddy Merckx

---

Subject: Re: IDL input files.  
Posted by [ryanselk](#) on Mon, 18 Jun 2007 22:19:56 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Jun 18, 3:23 pm, Paul van Delst <[Paul.vanDe...@noaa.gov](mailto:Paul.vanDe...@noaa.gov)> wrote:  
> [ryans...@gmail.com](mailto:ryans...@gmail.com) wrote:  
>> On Jun 18, 1:15 pm, David Fanning <[n...@dfanning.com](mailto:n...@dfanning.com)> wrote:

```

>>> Paul van Delst writes:
>>>> ryans...@gmail.com wrote:
>>>> > ive been reading some manuals and books for IDL but im not sure how to
>>>> > solve this likely simple problem.
>>>> > I have this I need to input to idl (.txt file):
>>>> > Name = {Joe}
>>>> > Department = {CS}
>>>> > Age = {25}
>>>> > and in IDL i need to save each attribute to its own variable. But if,
>>>> > lets day, department is ommited I want it to not input age as
>>>> > department. With c/c++ this isnt so bad as you can search the file for
>>>> > a 'keyword' and print the value from there, but I cant find how to do
>>>> > this in IDL.
>>>> Well, I don't think you can search the file in IDL, but you can read it in line by line
>>>> and search the lines. Why not create the file as IDL commands? e.g.:
>>>> ??
>>>> IDL> .run test
>>>> % Compiled module: TEST.
>>>> IDL> test
>>>> NAME      STRING  = 'Joe'
>>>> DEPARTMENT STRING  = 'CS'
>>>> AGE       INT     =  25
>>>> IDL> $more test.input
>>>> Name = "Joe"
>>>> Department = "CS"
>>>> Age = 25
>>> Well, I'd do this a little differently:
>
>>> pro test, name, dept, age
>
>>> ; Create file to read
>>> openw, lun, 'test.input', /get_lun
>>> if n_elements(name) NE 0 then $
>>>   printf, lun, 'Name = ' + name else $
>>>   printf, lun, 'Name = '
>>> if n_elements(dept) NE 0 then $
>>>   printf, lun, 'Department = ' + dept else $
>>>   printf, lun, 'Department = '
>>> if n_elements(age) NE 0 then $
>>>   printf, lun, 'Age = ' + StrTrim(age,2) else $
>>>   printf, lun, 'Age = '
>
>>> free_lun, lun
>
>>> ; Now read the file
>>> openr, lun, 'test.input', /get_lun
>>> buffer = ' '
>>> while not eof(lun) do begin

```

```

>>> readf, lun, buffer
>>> parts = StrSplit(buffer, " ", /Extract)
>>> case n_elements(parts) of
>>>     2: print, 'No value for ' + parts[0]
>>>     3: print, parts[0] + '= {' + parts[2] + '}'
>>>     else: print, 'Whoa, I am like totally confused!!'
>>> endcase
>>> endwhile
>>> free_lun, lun
>>> end

```

>  
>>> Then try it like this:

```

>
>>> IDL> test, 'coyote', 'PE', 43
>>>     Name= {coyote}
>>>     Department= {PE}
>>>     Age= {43}

```

```

>
>>> IDL> test
>>>     No value for Name
>>>     No value for Department
>>>     No value for Age

```

>  
>>> 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. ("Perhaps thou speakest truth.")

```

>  
>> Thank you guys! these should help, although I cant have IDL writing  
>> the program unfortunately. Ill have to figure out how to get the pre-  
>> made files read.

>  
> Um, that's what these examples do. The writing of the test input files was simply to  
> provide an entirely self-contained example. The various details that aren;t addressed are  
> the formats of the files. If you can't create them (i.e. change their formats to make them  
> more "IDL friendly" as it were), then all that needs to be modified in both David's and my  
> examples is how you extract the relevant information from the string you read from file  
> (i.e. the STRSPLIT.)

>  
> On a side note, I initially started using regular expressions with the STRSPLIT function  
> only to discover IDL seems to use a hamstrung version that doesn't understand stuff like  
> \s, \w, \n, etc. For the format you list, e.g.

>  
> Name = {Joe}

```
> Department = {CS}
> Age = {25}
>
> regexps would be the go (IMO) - although I don't see any mention of how you would capture
> certain parts of a match and not others (but I didn't look too hard :o)
>
> cheers,
>
> paulv
>
> --
> Paul van Delst           Ride lots.
> CIMSS @ NOAA/NCEP/EMC   Eddy Merckx
```

haha, I realized this as soon as I looked at the code but didnt at the time of my last post.

I got things working excellent. The strsplit I have working very well for my application. Somehow I didnt know about this command, now I got it figured out, very useful tool that I have missed.

Thanks again guys!

---

Subject: Re: IDL input files.

Posted by [Michael Galloy](#) on Tue, 19 Jun 2007 00:27:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On Jun 18, 4:19 pm, ryans...@gmail.com wrote:

```
> On Jun 18, 3:23 pm, Paul van Delst <Paul.vanDe...@noaa.gov>
>> On a side note, I initially started using regular expressions with the STRSPLIT function
>> only to discover IDL seems to use a hamstrung version that doesn't understand stuff like
>> \s, \w, \n, etc. For the format you list, e.g.
>
>> Name = {Joe}
>> Department = {CS}
>> Age = {25}
>
>> regexps would be the go (IMO) - although I don't see any mention of how you would capture
>> certain parts of a match and not others (but I didn't look too hard :o)
```

I'm just catching this at the end, but I think this would be useful for you:

```
IDL> results = stregex(line, '([[:alnum:]]*) = \{([[:alnum:]]*)\}', /
extract, /subexpr)
IDL> print, results[0]
```

```
Name = {Joe}
IDL> print,
results[1]
Name
IDL> print, results[2]
Joe
```

IDL regular expressions have character classes as well, but a different notation than Perl, i.e. \s is `[:blank:]`, \w is `[:alnum:]`, etc. See

<http://www.itvis.com/codebank/search.asp?FID=311>

for a more detailed account.

Mike

--

[www.michaelgalloy.com](http://www.michaelgalloy.com)

---