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