Subject: multiple delimiters
Posted by nrh on Thu, 14 Sep 2000 03:58:00 GMT
View Forum Message <> Reply to Message

Has anyone nutted out a way to read ASCII files with multiple
delimiters? Our current solution involves some messy string operations
that are restricted to 5.3, and I/O operations we would like to avoid.
Am I asking the impossible?

Subject: Re: multiple delimiters
Posted by nrh on Thu, 14 Sep 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Well, its actually a whole heap of strings, most fields separated by
blanks, and some fields, where there is more than one word, are
encased by quotation marks. The fields inside the quotes have spaces as
well, but we want them to be all one field, if you know what I mean.
Right now we pull out the strings within the quotes, replace all the
spaces with '_', put it back in, remove the quotes and then we can use
the strsplit function to remove the extra white spaces created by
replacing the quotes.
so, in a nutshell, we have:
....PROJECTION-R OTYP DP EXTN img PROC "CM CARDIAC MIBI" .....
and make it to be(through many painful string ops - it is a huge
database file)
PROJECTION-R OTYP DP EXTN img PROC CM_CARDIAC_MIBI .....
and then we have to arrange it in a struct as every second field is the
info we actually need. Odd fields are the descriptors.
Clear as mud?

> Did you have data as
>
> 1 ,3 <TAB> 5<SPACE>6
> 1 ,3 5 6
>
> please give me an example.
>
> Reimar
>
> --
> R.Bauer
>

Sent via Deja.com http://www.deja.com/
Before you buy.

Subject: Re: multiple delimiters
Posted by R.Bauer on Thu, 14 Sep 2000 07:00:00 GMT
View Forum Message <> Reply to Message

nrh@imag.wsahs.nsw.gov.au wrote:
>
> Has anyone nutted out a way to read ASCII files with multiple
> delimiters? Our current solution involves some messy string operations
> that are restricted to 5.3, and I/O operations we would like to avoid.
> Am I asking the impossible?
>
> Sent via Deja.com http://www.deja.com/
> Before you buy.

Did you have data as

1 ,3 <TAB> 5<SPACE>6
1 ,3 5 6

please give me an example.

Reimar

--
R.Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de

Subject: Re: multiple delimiters
Posted by meron on Thu, 14 Sep 2000 07:00:00 GMT
View Forum Message <> Reply to Message

In article <39C09FE1.C9E0B54@dkrz.de>, Martin Schultz <martin.schultz@dkrz.de> writes:
> nrh@imag.wsahs.nsw.gov.au wrote:

>>
>> Has anyone nutted out a way to read ASCII files with multiple
>> delimiters? Our current solution involves some messy string operations
>> that are restricted to 5.3, and I/O operations we would like to avoid.
>> Am I asking the impossible?
>>
>> Sent via Deja.com http://www.deja.com/
>> Before you buy.
>
> 1. Read the file line by line as strings
> 2. use my StrRepl function to replace all delimiters with one value
>   e.g.
>   line = StrRepl(line, ';',' ')
>   line = StrRepl(line, ',',' ')
>   line = StrRepl(line, ':',' ')
> 3. Use ReadS, Str_Sep or StrSplit (5.3) to extract the numbers.
> Caution: With Str_Sep or StrSplit you should always add a
> StrTrim(StrCompress(line),2) before
>
> You can find StrRepl at
> http://www.mpimet.mpg.de/~schultz.martin/idl/html/libmartin_ schultz.html
>
Or, you can use my STRPARSE, which takes any number of delimiters.

Mati Meron | "When you argue with a fool,
meron@cars.uchicago.edu | chances are he is doing just the same"

Subject: Re: multiple delimiters
Posted by Martin Schultz on Thu, 14 Sep 2000 07:00:00 GMT
View Forum Message <> Reply to Message

... or you use some Unix tool like sed or awk to get rid of the
multiple delimiters before you even touch IDL. This will certainly
speed things up if you have a large file(s) and intend to access it
frequently.

Martin

--
[[[[[[[[[[
[[Dr. Martin Schultz   Max-Planck-Institut fuer Meteorologie   [[
[[Bundesstr. 55, 20146 Hamburg   [[
[[phone: +49 40 41173-308   [[
[[fax: +49 40 41173-298   [[
[[martin.schultz@dkrz.de   [[
[[[[[[[[[[

Subject: Re: multiple delimiters
Posted by Martin Schultz on Thu, 14 Sep 2000 07:00:00 GMT
View Forum Message <> Reply to Message

nrh@imag.wsahs.nsw.gov.au wrote:
>
> Has anyone nutted out a way to read ASCII files with multiple
> delimiters? Our current solution involves some messy string operations
> that are restricted to 5.3, and I/O operations we would like to avoid.
> Am I asking the impossible?
>
> Sent via Deja.com http://www.deja.com/
> Before you buy.

1. Read the file line by line as strings
2. use my StrRepl function to replace all delimiters with one value
   e.g.
```
    line = StrRepl(line, ';',' ')
    line = StrRepl(line, ',',' ')
    line = StrRepl(line, ':',' ')
```
3. Use ReadS, Str_Sep or StrSplit (5.3) to extract the numbers.
Caution: With Str_Sep or StrSplit you should always add a
StrTrim(StrCompress(line),2) before

You can find StrRepl at
http://www.mpimet.mpg.de/~schultz.martin/idl/html/libmartin_ schultz.html

Cheers,
Martin

--
[[[[[[[[[
[[Dr. Martin Schultz   Max-Planck-Institut fuer Meteorologie   [[
[[Bundesstr. 55, 20146 Hamburg   [[
[[phone: +49 40 41173-308   [[
[[fax: +49 40 41173-298   [[
[[martin.schultz@dkrz.de   [[
[[[[[[[[[

Subject: Re: multiple delimiters
Posted by Chris Rennie on Fri, 15 Sep 2000 04:58:43 GMT
View Forum Message <> Reply to Message

nrh@imag.wsahs.nsw.gov.au wrote:
>
> Well, its actually a whole heap of strings, most fields separated by
> blanks, and some fields, where there is more than one word, are

> encased by quotation marks. The fields inside the quotes have spaces as
> well, but we want them to be all one field, if you know what I mean.
> Right now we pull out the strings within the quotes, replace all the
> spaces with '_', put it back in, remove the quotes and then we can use
> the strsplit function to remove the extra white spaces created by
> replacing the quotes.
> so, in a nutshell, we have:
> ....PROJECTION-R OTYP DP EXTN img PROC "CM CARDIAC MIBI" .....
> and make it to be(through many painful string ops - it is a huge
> database file)
> PROJECTION-R OTYP DP EXTN img PROC CM_CARDIAC_MIBI .....
> and then we have to arrange it in a struct as every second field is the
> info we actually need. Odd fields are the descriptors.
> Clear as mud?

This is my suggestion:

```
PRO ParseLine, line, structure
   ; This routine first separates the line into 'coarse' chunks, based
on
   ; using quoation marks as delimiters.  This intermediate result is
   ; a set of strings.  Every 0th, 2nd, 4th,... string is then
separated
   ; further by using spaces as delimiters, and every 1st, 3rd, 5th....
   ; string has its spaces translated to underscores.

   CoarseChunks=str_sep(line,"")
   if (n_elements(CoarseChunks) mod 2) ne 1 then stop, 'ParseLine
error'

   ; Process 0th coarse chunk
   FineChunks=str_sep(CoarseChunks[0],' ')
   structure.field0=FineChunks[0]
   structure.field1=FineChunks[1]
   structure.field2=FineChunks[2]
   structure.field3=FineChunks[3]
   structure.field4=FineChunks[4]
   structure.field5=FineChunks[5]

   ; Process 1st coarse chunk
   Bytearray=byte(CoarseChunks[1])
   spaces=where(ByteArray eq 32, NSpaces)
   if NSpaces gt 0 then ByteArray[spaces]=byte('_')
   structure.field6=string(ByteArray)

   ; Process 2nd coarse chunk
   CoarseChunks[2]=strtrim(CoarseChunks[2],2)
```

```
   FineChunks=str_sep(CoarseChunks[2],' ')
   structure.field7=FineChunks[0]
   structure.field8=FineChunks[1]
end   ; ParseLine

;;;;;;;;;;;;;;;;;;;;;;;; main ;;;;;;;;;;;;;;;;;;;;;;;;
TestLine='PROJECTION-R OTYP DP EXTN img PROC "CM CARDIAC MIBI" etc etc'
TestStruct={field0:'', field1:'', field2:'', field3:'', field4:'', $
          field5:'', field6:'', field7:'', field8:''}
ParseLine, TestLine, TestStruct

print, TestStruct
end
```

This is the result:

```
IDL> help, /struct, TestStruct
** Structure <8192ab4>, 9 tags, length=72, refs=1:
   FIELD0          STRING    'PROJECTION-R'
   FIELD1          STRING    'OTYP'
   FIELD2          STRING    'DP'
   FIELD3          STRING    'EXTN'
   FIELD4          STRING    'img'
   FIELD5          STRING    'PROC'
   FIELD6          STRING    'CM_CARDIAC_MIBI'
   FIELD7          STRING    'etc'
   FIELD8          STRING    'etc'
```

--
Chris Rennie rennie@physics.usyd.edu.au
Rm 466, School of Physics
Building A29 Tel: +61 (2) 9351 5799
University of Sydney
NSW 2006, Australia Fax: +61 (2) 9351 7726

Subject: Re: multiple delimiters
Posted by Martin Schultz on Fri, 15 Sep 2000 07:00:00 GMT
View Forum Message <> Reply to Message

nrh@imag.wsahs.nsw.gov.au wrote:
>
> Well, its actually a whole heap of strings, most fields separated by
> blanks, and some fields, where there is more than one word, are
> encased by quotation marks. The fields inside the quotes have spaces as
> well, but we want them to be all one field, if you know what I mean.
> Right now we pull out the strings within the quotes, replace all the
> spaces with '_', put it back in, remove the quotes and then we can use

> the strsplit function to remove the extra white spaces created by
> replacing the quotes.
> so, in a nutshell, we have:
> ....PROJECTION-R OTYP DP EXTN img PROC "CM CARDIAC MIBI" .....
> and make it to be(through many painful string ops - it is a huge
> database file)
> PROJECTION-R OTYP DP EXTN img PROC CM_CARDIAC_MIBI ....
> and then we have to arrange it in a struct as every second field is the
> info we actually need. Odd fields are the descriptors.
> Clear as mud?

Your life could be a LOT easier if you had a formatted output, i.e. if all columns are aligned (I am sure the database that you are using should be able to produce this). Then you could simply use a formatted read statement

```
   readf, lun, proj, otyp, dp, extn, img, proc, label, ...,
format='(i4,i6,...,A20,...)'
```

or (even more elegantly) read into a structure

```
   temp = { proj:1.0, otyp:0L, ..., label:'', ... }
   readf, lun, temp, format='...'
```

It probably boils down to workflow optimization: If you have to do it once, don't bother and just use a simple code. If you have to do it several times - always with the same data set - convert the data set once into a better format (something binary to speed up reading, optimally a scientific data format for they allow better usability and are self-describing). If you need to do this several times with changing data sets, make sure the original data set producer change their format ;-)

Cheers,
Martin

--
[[[[[[[[[[
[[Dr. Martin Schultz   Max-Planck-Institut fuer Meteorologie   [[
[[Bundesstr. 55, 20146 Hamburg   [[
[[phone: +49 40 41173-308   [[
[[fax: +49 40 41173-298   [[
[[martin.schultz@dkrz.de   [[
[[[[[[[[[