
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 quocation 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

```
