Subject: Re: Substituting multiple strings into a program
Posted by Paul Van Delst[1] on Wed, 08 Aug 2012 21:49:02 GMT
View Forum Message <> Reply to Message

On 08/08/12 17:04, MightyMrFish wrote:
> I want something that will read one of the filenames and insert it into my program. How I have it set up right now,
> I'm using this command line: name ='' read, 'What is the filename?', name
>
> Then I use a READCOL command to plot the data:
>
> READCOL, name, ï¿½
>
> I'd like it if I could have the "name" part automatically filled in with the names on the list. One by one of course.
> Once the name is inserted, the rest of the command does its thing and I get a graph of the data.
>
> If it helps, I'll post my program.
>
>
> On Wednesday, August 8, 2012 1:16:20 PM UTC-7, Paul van Delst wrote:
>> So you want to read a "control" file containing a list of filenames (or portions of filenames), and then loop over
>> that
>>
>> filename list reading and displaying the data in each file?

O.k., still not sure I understand. But, here's what I came up with:

1) First create (what I call) the control file:

```
$ cat > dognames.txt
Rex
Cherry
Sniffles
CaptainKevin
^D
```

2) O.k., now the following procedure reads that file

```
PRO ReadControlFile, control_file
```

```
  ; Open the control file
  OPENR, fid, control_file, /GET_LUN

  ; Loop over the entries in the control file
  entry = ''
  WHILE ~EOF(fid) DO BEGIN
    READF, fid, entry
    PRINT, "The current entry is: ", entry
  ENDWHILE

  ; Close the control file
  FREE_LUN, fid

END

IDL> .run readcontrolfile
% Compiled module: READCONTROLFILE.
IDL> readcontrolfile, 'dognames.txt'
The current entry is: Rex
The current entry is: Cherry
The current entry is: Sniffles
The current entry is: CaptainKevin
```

3) Now let's assume there is a separate datafile associated with each entry in the control file, e.g.

```
$ cat Rex.dat
# time woof-factor
   1     2.4
   2     3.6
   3     4.8
   4     15.6
   5     6.7
   6     5.2
```

Let's just copy the one file for the other doggies,

```
$ cp Rex.dat Cherry.dat
$ cp Rex.dat Sniffles.dat
$ cp Rex.dat CaptainKevin.dat
```

and edit them so they have different numbers of lines in them:

```
$ wc -l *.dat
  4 CaptainKevin.dat
  6 Cherry.dat
```

```
  7 Rex.dat
  5 Sniffles.dat
 22 total
```

4) Let's write a script to read that sort of datafile and return its data (doing this bit simply is probably the hardest
part),

```
PRO ReadDataFile, data_file, x, y

  ; How many lines?
  n_lines = FILE_LINES(data_file)

  ; Assume first line is comment
  n_points = n_lines - 1

  ; And now create the output x and y arrays
  x = FLTARR(n_points)
  y = FLTARR(n_points)

  ; Open the data file
  OPENR, fid, data_file, /GET_LUN

  ; Loop over the lines in the datafile
  entry = ''
  a = 0.0
  b = 0.0
  FOR i = -1, n_points - 1 DO BEGIN
    READF, fid, entry            ; Read the line into a string
    IF ( i EQ -1 ) THEN CONTINUE  ; If it's the comment line, skip it
    READS, entry, a, b           ; Read the data from the string
    x[i] = a                     ; Assign the data to...
    y[i] = b                     ; ...the output arrays
  ENDFOR

  ; Close the data file
  FREE_LUN, fid

END

IDL> .run readcontrolfile
% Compiled module: READDATAFILE.
% Compiled module: READCONTROLFILE.
IDL> readdatafile, 'Rex.dat', x, y
IDL> help, x, y
X               FLOAT     = Array[6]
```

```
Y               FLOAT     = Array[6]
IDL> print, x, y
      1.00000      2.00000      3.00000      4.00000      5.00000      6.00000
      2.40000      3.60000      4.80000      15.6000      6.70000      5.20000
```

5) Now modify the control file reader to also read the individual data files:

```
PRO ReadControlFile, control_file

  ; Open the control file
  OPENR, fid, control_file, /GET_LUN

  ; Loop over the entries in the control file
  entry = ''
  WHILE ~EOF(fid) DO BEGIN
    READF, fid, entry
    PRINT, "The current entry is: ", entry,". Reading the associated datafile..."
    ReadDataFile, entry+'.dat', x, y
    HELP, x, y
  ENDWHILE

  ; Close the control file
  FREE_LUN, fid

END

IDL> .run readcontrolfile
% Compiled module: READDATAFILE.
% Compiled module: READCONTROLFILE.
IDL> readcontrolfile,'dognames.txt'
The current entry is: Rex. Reading the associated datafile...
X               FLOAT     = Array[6]
Y               FLOAT     = Array[6]
The current entry is: Cherry. Reading the associated datafile...
X               FLOAT     = Array[5]
Y               FLOAT     = Array[5]
The current entry is: Sniffles. Reading the associated datafile...
X               FLOAT     = Array[4]
Y               FLOAT     = Array[4]
The current entry is: CaptainKevin. Reading the associated datafile...
X               FLOAT     = Array[3]
Y               FLOAT     = Array[3]
```

Of course, rather than "HELP, x, y" you can do "PLOT, x, y" or whatever.

cheers,

paulv