

This code will take as input a comma seperated text data file (a '.csv'), and will create an array with the following dimensions:

First dimension: The number of lines in the .csv file

Second dimension: The number of fields per line in the .csv

The zero position of the array will be filled with the contents of the first line of the .csv, which can be the field names for the rest of the data lines.

The rest of the array will contain all the values from the file.

The code includes an array inspector window to aid in debugging.

This code worked ok with my .csv which originally had 294 lines, each line with six fields; the last field being a 'comment' field which contained text within double quotes.

The subroutines were hard-coded to ignore commas found within the text inside the sixth field.

Modify to your use at will!

```
global readingsarray$  
global datalinecounter  
dim readingsarray$(11,11)  
  
call LoadGraphData  
  
call ArrayInspect  
  
wait  
  
  
sub LoadGraphData  
  
'find out how big the file is before dimming  
datalinecounter = -1 'offset the final wend below  
open "blood_pressure_readings.csv" for input as #datafile  
while eof(#datafile) = 0  
line input #datafile, datastring$  
datalinecounter = datalinecounter + 1  
wend  
close #datafile  
  
'create the correct-sized array
```

```
'add 1 because array starts at zero
redim readingsarray$(datalinecounter + 1,5)

rowcounter = 0
open "blood_pressure_readings.csv" for input as #datafile

for dataloadloop = 0 to datalinecounter
line input #datafile, datastring$

itemToFill = 0
ignore = 0

for scanChar = 1 to len(datastring$)

if mid$(datastring$,scanChar,1) = chr$(34) then ignore = 1

if (mid$(datastring$,scanChar,1) = "," and ignore = 0) then
itemToFill = itemToFill + 1
else
readingsarray$(rowcounter,itemToFill) = _
readingsarray$(rowcounter,itemToFill) + _
mid$(datastring$,scanChar,1)
end if

next scanChar

rowcounter = rowcounter + 1
next dataloadloop

close #datafile
end sub

sub ArrayInspect

WindowWidth = 550
WindowHeight = 302

TexteditorColor$ = "white"
texteditor #arrayinsp.text, 1, 5, 540, 250

open "Array Inspector" for window as #arrayinsp
print #arrayinsp, "font ms_sans_serif 10"
print #arrayinsp, "trapclose [quit.arrayinsp]"
```

```
for printout = 0 to datalinecounter
print #arrayinsp.text, "Line " + str$(printout) + " of "
+ str$(datalinecounter) + " : ";
for subloop = 0 to 5
print #arrayinsp.text, readingsarray$(printout,subloop);
if subloop < 5 then
print #arrayinsp.text, ",";
else
print #arrayinsp.text, ""
end if
next subloop

next printout
end sub

[quit.arrayinsp] 'close inspector
close #arrayinsp
wait
```

Here is some example .csv data which should work for testing the above code. Save it to a text file then change the file extension to '.csv'. The actual filename referenced in the code above is 'blood\_pressure\_readings.csv'. Note that there are two instances of the filename to edit.

```
Date, Time, Systolic, Diastolic, Pulse, Notes
28/07/2005,08:45,116,92,75, ""
28/07/2005,10:30,124,96,68, ""
28/07/2005,18:30,134,94,65, ""
30/07/2005,12:00,107,76,70, "Whatever you want, "
31/07/2005,12:00,110,75,85, ""
31/07/2005,15:00,108,78,85, "as notes, can go "
31/07/2005,18:00,116,88,68, ""
02/08/2005,12:00,107,78,78, "between the quotes"
02/08/2005,20:00,115,84,85, ""
03/08/2005,20:00,102,73,92, "in"
04/08/2005,08:30,116,86,66, ""
04/08/2005,12:00,121,100,68, "the sixth"
04/08/2005,12:30,116,92,78, ""
04/08/2005,14:10,110,75,76, "field"
05/08/2005,11:45,115,76,88, ""
05/08/2005,19:00,120,80,89, "including, commas"
05/08/2005,20:20,112,81,84, ""
05/08/2005,23:00,121,89,86, ""
05/08/2005,23:59,113,75,82, ""
```

-  
[scapegoat](#)