SESUG Paper 212-2023

Reading Comma Separated Files containing CR/LF embedded in quoted strings

Nat Wooding, J. Sargeant Reynolds Community College

ABSTRACT

Comma Separated Files (CSV) are commonly used for exchanging data between applications such as spreadsheets and other systems such as SAS®. Text strings sometimes contain embedded commas and the accepted practice for creating CSVs is to cloak the commas using quotation marks. Furthermore, some strings may contain carriage returns and or line feed characters used to format strings within cells in the original application. SAS does not recognize that the CR/LFs are intended to be cloaked and, unfortunately, breaks the input line at the control characters. Here, we look at a method of fixing this problem.

INTRODUCTION

Spreadsheets with cells containing text are often formatted so that the text is “wrapped” within the cell and appears as a series of short lines making the sheet more readable. Also, the text may contain embedded commas. In order to store the contents of the spreadsheet as a text file with interpretable column boundaries, a common solution is to enclose the contents of such cells in quotation marks and computer languages that adhere to industry standards ignore the commas within the quotes in these so-called Comma Separated Files (CSV). Furthermore, the text strings sometimes are sometimes internally formatted using hexadecimal characters that indicate a carriage return (CR, hex 0A) and or line feed (LF, hex 0D) and there is an industry standard which if followed by the import software, ignores the CR and LF when embedded in quotes. While SAS recognizes the masking of quotes when reading a CSV file, it treats the CR and LF as legitimate control characters and treats the import record as two or more records and this, of course, produces garbage results.

THE PROBLEM

My group recently encountered this situation when we attempted to download student data from the educational support tool Navigate®. The files are presented as CSVs and open in Excel ® with no issues. My colleague used Proc Import to read one of the files and it dutifully scanned the header lines and input and then bombed – the input variables were a mixture of character and numeric types and it was encountering text where it expected numbers to appear. Table 1 shows how the data would appear in a spreadsheet.

Name	Age	Occupation	Pet
Joe	19	Student	Python
Amy	35	Teacher Teaches Reading in first grade	Kitten

Table 1. Simple table with cells containing line breaks

The SAS log for the Proc Import step produces the note shown in Figure 1.

```
NOTE: Invalid data for Age in line 4 34-39.
RULE:     ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+--
--
4          Teaches Reading in first grade",Kitten 39
Name=Teaches Reading in first grade" Age=. Occupation=  Pet=  _ERROR_=1 _N_=3
NOTE: 3 records were read from the infile 'C:\Users\natha\Documents\SESUG
      2023\Sample_Sheet_CSV.csv'.
      The minimum record length was 15.
      The maximum record length was 39.
NOTE: The data set WORK.TEST_DATA has 3 observations and 4 variables
```

Figure 1. Log of Proc Import step showing invalid data values

I was asked to look into this and blundered about naively for a little while as I ignored critical information that was plainly displayed as shown in the log above. In particular, notice that while there were two input records, the new data set has three observations. Figure 2 shows the resultant SAS data set.

	Name	Age	Occupation	Pet
1	Joe	19	Student	Python
2	Amy	35	"Teacher	
3	Teaches Reading in first grade"	.		

Figure 2. SAS data set containing the results of the Proc Import step

These data are, of course, a lot simpler than those of the original file but in any case, it's obvious here that part of Occupation in the second input record has been read as the start of a third input record and Name in the third observation contains part of the Occupation in the second observation.

Eventually, it occurred to me that there appeared to be some sort of control characters in some of the text fields so I tried reading the lines with SAS and displaying the data using a hexadecimal format but I found no unwanted control characters. After posting my dilemma on SAS/L, I received a suggestion that I use something like UltraEdit ® to view the files contents and upon doing so, discovered the hidden character causing the problem. Figure 3 shows the UltraEdit in hex mode.

```
4E 61 6D 65 2C 41 67 65 20 2C 4F 63 63 75 70 61 ; Name,Age ,Occupa
74 69 6F 6E 20 2C 50 65 74 0D 0A 4A 6F 65 2C 31 ; tion ,Pet..Joe,1
39 2C 53 74 75 64 65 6E 74 20 2C 50 79 74 68 6F ; 9,Student ,Pytho
6E 0D 0A 41 6D 79 2C 33 35 2C 22 54 65 61 63 68 ; n..Amy,35,"Teach
65 72 0A 20 54 65 61 63 68 65 73 20 52 65 61 64 ; er. Teaches Read
69 6E 67 20 69 6E 20 66 69 72 73 74 20 67 72 61 ; ing in first gra
64 65 22 2C 4B 69 74 74 65 6E 0D 0A             ; de",Kitten..
```

Figure 3. The CSV displayed in hex mode using Ultraedit.

Note the 33 and 35 which tell us that we are looking at Amy's age and therefore, 41, 6D and 79 spell Amy which is followed by 2C, a comma. Immediately prior are 0D and 0A which are the CR and LF for the first record. Looking at the next line in the display, we see 0A which is the control character that causes the first line break in the original cell and is the culprit.

THE SOLUTION

At this point, I had a good idea of the nature of the problem but no programmatic solution. I could, of course, have edited the CSV manually but we were setting up an automated process that would download these files and create SAS data sets for use in dashboards so manual editing was not a good solution. My post to SAS/L brought in suggestions for using Python or other tools but our group doesn't use this language and we wanted a vanilla SAS solution. Fortunately, one of the gurus on SAS/L pointed me to the SAS Technical Support sample (I had forgotten to check for such)

Sample *26065:* Remove carriage return and linefeed characters within quoted strings

Whose link is https://support.sas.com/kb/26/065.html

The basis of the solution is to read the input data as a binary file using RECFM =N. This ignores the internal control characters but recognizes the end of file marker. Then, you read the input one character at a time and when the program finds a specified control character inside a quoted string, the character is set to a blank

```
%let dsnnme="path to file.csv";

data _null_;
  /* RECFM=N reads the file in binary format. The file consists    */
  /* of a stream of bytes with no RECORD boundaries.  SHAREBUFFERS */
  /* specifies that the FILE statement and the INFILE statement    */
  /* share the same buffer.                                         */

  infile &dsnnme recfm=n sharebuffers;
  file   &dsnnme recfm=n;
  * Note that infile and file refer to the same file so we are overwriting
our input file;
  /* OPEN is a flag variable used to determine if the CR/LF is within */
  /* double quotes or not.  Retain this value.                        */
  retain open 0;
  input a $char1.;
  /* If the character is a double quote, set OPEN to its opposite value. */
  if a = '"' then open = ^(open);
  /* If the CR or LF is after an open double quote, replace byte with */
  /* a blank.                                                         */
  if open then do;
    if a = '0D'x then put ' ';
    else if a = '0A'x then put ' ';
  end;
run;
```

This code gave us a clean CSV with no embedded line breaks.

CONCLUSION

Data from other software products may contain embedded carriage return or line feed hexadecimal characters enclosed in quotation marks but SAS does not recognize these marks as being masked and breaks the input record lines improperly. The solution is to read the input data as a single binary record one character at a time, convert the offending characters to blanks, and write a new file.

ACKNOWLEDGMENTS

Many thanks to the contributors of the SAS/L discussion group.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Nat Wooding
J Sargeant Reynolds Community College
804-523-5884
nwooding@reynolds.edu