

The late publication of this issue has allowed me to include a report on the *VIEWS* 2003 Conference, which was held on May 20. The *VIEWS* Committee is now planning a 3-day conference for 2004, as well as Sub-Group Meetings in September and November this year, so now, after a long gap, we have plenty of events for you. More details can be found in the Diary section.

If you would like to contribute an article, or discuss the possibility of doing so, please send an email to me at newsletter@views-uk.org. A list of possible subjects for your articles can be found on the *VIEWS* web site, and anyone is very welcome to add to that list by sending emails with your own questions to newsletter@views-uk.org.

Philip R Holland (Newsletter Editor)

Membership of *VIEWS* is **free** and you can register for notification of all future *VIEWS* events by emailing us at membership@views-uk.org, making certain you include your name and postal address. If you would like to receive a free copy of the quarterly *VIEWS News* by email, please remember to also include your email address.

News - Stop Press!

Report on the *VIEWS* 2003 Conference

After missing 2002, the *VIEWS* Conference was reinstated to the conference calendar at the Commonwealth Conference & Events Centre in Kensington on May 20. The Exhibition Hall held the *VIEWS* Committee desk (which doubled as the Technical Question & Answer desk), the SAS Books-by-Users stand, and 7 sponsored stands from Aim Resourcing, Amadeus Software, Futrix, IMAJ Consulting, Mango Solutions, Oxford Pharmaceutical Sciences, and Statistical Tabling Services. The 3 seminar rooms allowed 245 delegates from 117 companies to watch 21 speakers present 35 technical SAS papers, and, apart from the Hunt Room initially living up to its name and the occasional showers, I only heard praise for the venue and conference as a whole on the day.

At the end of the day the Best Paper Award of £250 was won by Graham Murray. It was also announced that the *VIEWS* Committee were very happy with the support from delegates and sponsors, and the range and quality of the papers presented, and are now planning a 3-day conference for 2004 from 19-21 May at the same venue.

The *VIEWS* 2004 Conference is dependent on user papers being submitted. With the announcement of next year's provisional date now, you have almost 12 months to write your papers and presentations for 2004. It must also be emphasised that, while the conference will be held in London and *VIEWS* is a UK SAS User Group, we welcome presenters from Europe and the rest of the World.

2003 has been a good year, and the *VIEWS* Committee would like to make 2004 even better!

Philip R Holland (Newsletter Editor)

The Consultant

This part of *VIEWS News* is where you can get your technical questions answered. Send your questions to the Editor.

Reading CSV Files with Embedded New-Lines

Q: How do you read comma-separated files into SAS when the data contains embedded new-lines?

A: It is often necessary to use data in comma separated value (CSV) format either to import occasionally or regularly as a data source. CSV is a simple format, available to export from spreadsheets and other packages, yet it has named columns so that importing is often very simple.

However, the CSV file type supports one variation that causes normal SAS input processing serious trouble. That is the use of additional 'new-line' breaks in text fields. The simplest example is created in an Excel spreadsheet by using the 'Alt-Enter' key combination when typing a text cell.

	A	B	C
1	NumberCol	DateCol	LongTextCol
2	123	15/04/2003	Some descriptive text
			More descriptive text.
3	456	18/04/2003	This time with an extra new-line to aid readability, punctuated as we wish.
4	890	20/04/2003	If not removed our import is affected!
5			

When writing this data to CSV, Excel encloses the value in quotes ("...") whenever the cell contains such a value – according to the standard specification. Opening this CSV in Excel will display the cell with the value as originally entered, including the extra new-line.

Problems arise however when reading this into SAS using the standard features (Import wizard, **PROC IMPORT** and so on) all of which ultimately use the SAS Data Step to process the file. **INFILE** processing is designed to treat CR-LF as end-of-record, and it does this even when enclosed in quotes!

The main effect of this is to break to a new input line as soon as the CR-LF is encountered. The text following the CR-LF will be read as the start of the next line of input. Visible effects of this are:-

- Text read into the wrong column of the SAS table
- Text starting with a quote (") without a closing quote
- Text ending with a closing quote without a starting quote
- Missing numeric values (caused by reading text as numeric data)

Other effects may be seen if a cell containing this appears near the beginning of the file. **PROC IMPORT** and the Import

Wizard both use the first few rows to determine data type for the columns; therefore misreading of this text can cause:-

- Numeric columns to be read into character SAS columns
- Date values to be read as text into character SAS columns

Both of these issues occur with the example file shown:

VIEWTABLE: Work.Testfile1_data			
	NumberCol	DateCol	LongTextCol
1	123	15/04/2003	Some descriptive text
2	456	18/04/2003	"More descriptive text
3	This time with an extra new-line to aid readability	punctuated as we wish"	
4	890	20/04/2003	If not removed our import is affected!

If you are faced with such data, how can you assure that it is read successfully? Especially in a batch environment, where the filename may be parameterised?

One answer is "get the file created without the extra CR-LF", but that might not always be possible. A second potential answer is to use SAS/ACCESS for ODBC, having defined the required CSV file as a data source using Microsoft's CSV file driver; this uses the same 'engine' that Excel does, so it should handle the text correctly.

If neither of those is available then the pre-processing technique described here can solve the problem!

The basis of the technique is that SAS **DATA** step **INFILE** processing supports "byte-stream" reading of data, using **RECFM=N** to describe the file as 'unformatted binary'. You might think that all you need do is to recall the **DATA** step code generated by **PROC IMPORT** and change the **INFILE** options. However, when you do that you lose all end-of-record processing so it is even worse than before!

Thus the answer is to read the file with a **DATA** step using **RECFM=N**, changing the CR and LF to some harmless text, and writing out to a temporary file. You can then use **PROC IMPORT** or the wizard on the resulting file and have a successful import. The following code does this:

```
* This program removes additional CR-LF from inside
quoted strings. This is necessary so that the SAS
DATA step can correctly read the CSV file.;
%let infile=c:\testfile1.csv;
%let outfile=c:\testfile1_processed.csv;

data _null_;
  infile "&infile" recfm=N lrecl=32767 end=eof;
  file "&outfile" recfm=N lrecl=32767;
  do until (eof);
    * read bitwise and look for double-quote;
    input char $1.;
    * look for a " character (hex 22) used to enclose
      strings in CSV;
    if char='22'x then do;
      * inside quoted string, so write the quote and
        then look for CR or LF;
      put char $1.;
      char=' ';
      do until (char='22'x);
        * process until next " is found;
        input char $1.;
        if char='0D'x or char='0A'x then do;
          * change to a blank;
          char=' ';
        end;
        put char $1.;
      end;
    end;
    * for everything else just write it out;
    else put char $1.;
  end;
  stop;
run;
```

Key features of this utility program in addition to the use of **RECFM=N** for byte stream processing are:

- Using hexadecimal constants to match the ASCII symbols (quote, CR and LF).
- Replacing CR or LF with a blank.

When the resulting processed CSV file is read in the difference is clear to see!

VIEWTABLE: Work.Testfile1_processed_data			
	NumberCol	DateCol	LongTextCol
1	123	15/04/2003	Some descriptive text
2	456	18/04/2003	More descriptive text This time with an extra new-line to aid readability, punctuated as we wish
3	890	20/04/2003	If not removed our import is affected!

Now the text columns have everything – although, naturally, the new-line breaks have gone; and most importantly the data types and formats are correct for the numeric and date columns.

Steve Morton, Applied System Knowledge Ltd.

Which is better, the DATA step or PROC SQL?

Q: When should you use PROC SQL in preference to the DATA Step, and vice versa?

A: The simple answer is that, of course, it depends on who you are, what you are trying to do, and what your data looks like. But, in general, learn to use the power and flexibility of the SAS data step. Unless you fit the following profile:

- You are a statistician or analyst more interested in getting data into a SAS Procedure [PROC step] like PROC GLM.
- And you are more familiar with SQL methods and concepts via use of Oracle or MS Access.
- And your data is served up to you already in SAS datasets or tables.
- And your requirement to pre-process is limited to straightforward joins and selection criteria.
- And your tables are small, or you can tolerate the performance difference.

The situation reverses if you are using SAS/ACCESS to a RDBMS like Oracle: see the end of this discussion.

When is it really necessary or worthwhile to use PROC SQL?

Here are some situations where PROC SQL is a better choice to achieve an outcome than a DATA step in descending order of probability:

- Recoding data using a lookup table, where more than one column contributes to selecting the result (and so you can't simply use a format in a PUT function).
- Using the **DISTINCT** modifier with the **COUNT** function to count the number of discrete values in a table.
- Merging normalised data across several tables, especially if the tables are not large and the ease of writing and maintaining SQL statements (as against multiple sorts and merges) outweighs minor performance considerations.
- This one is worth remembering: you can use the **SEPARATED BY** clause in a **SELECT ... INTO** statement to create a single macro variable containing a list of values with a specified delimiter.

- Merges with more than one table having multiple rows with the same **BY** values, and you want to join every instance in each table with every instance in every other.
- Updating a table where some level of 'undo' is required.
- Use of the **DESCRIBE** statement:
 - a. with a table to list the characteristics of the columns, especially prior to transferring a table to a DBMS that is supported by SAS/ACCESS software.
 - b. with a **PROC SQL** view, to audit the statements used to construct the view (not available for dataset views).
 - c. with **TABLE CONSTRAINTS** to list the integrity constraints that are defined for a specific table.

You might also consider using **PROC SQL** as a replacement for **PROC PRINT** for displaying the values in a small table.

When is PROC SQL contra-indicated?

The SAS **DATA** Step is the best (if not the only) option for the following tasks:

- Reading data in from, or writing out to, external files (other than RDBMS tables).
- Reading data in from, or writing out to, SAS files on tape.
- Invoking a **CALL** routine.
- Iterative execution of commands in a **DO** loop.
- Creating and displaying windows for user input or information.
- Identifying the first or last row in a **BY** group.
- Creating more than one table in a single pass of the data.
- Processing arrays of variables.
- Conditionally halting execution with **STOP** (which terminates **DATA** step processing), or **ABORT** (which kills the whole job).

Accessing data in an external RDBMS with SAS/ACCESS

If your data is in a warehouse built on Oracle or some other RDBMS using SAS/ACCESS to import data to, or export from, SAS the balance changes. While in Version 8 it is possible to use the SAS **DATA** step in conjunction with a **LIBNAME** referencing the RDBMS data to write SAS **DATA** steps which treat the RDBMS similarly to an ordinary SAS library, it is only efficient to do so when all the SAS statements and functions can be translated in the interface to equivalent RDBMS statements and functions (or the tables being queried are suitably small). Typically, this precludes the use of formats to select or recast data.

The best and most robust solution is to write native RDBMS SQL code inside a query using the **CONNECTION TO [RDBMS]** functionality. In this way, multiple table merges can be managed, selection criteria can be applied, tables can be ordered using the RDBMS to sort the data, and special properties of the RDBMS such as Oracle hints can be exploited. **PROC SQL** also allows passing commands to the RDBMS system using the **EXECUTE** statement.

Richard Carson, Sysware Consulting

Did You Know?

If you have a useful hint or tip, please send it to the Editor and share it with the **VIEWS** membership.

A Flexible New Date Format

[SAS 9] In SAS version 9 there is a new date informat which interprets dates being read based on the value of a system option. The system option is called **DATESTYLE** and it is used to identify sequence of month, day and year when the **ANYDTDTE** informat data is ambiguous. When dates being read are not ambiguous, then the option is ignored and date is read correctly. This can be seen in the SAS Log below:

```
33  options datestyle=mdy;
34  data _null_;
35      date=input('01/02/03',anydtdte8.);
36      * ambiguous date ;
37      put date=date9.;
38  run;

date=02JAN2003

38  options datestyle=ydm;
39  data _null_;
40      date=input('01/02/03',anydtdte8.);
41      * ambiguous date ;
42      put date=date9.;
43  run;

date=02MAR2001

43  options datestyle=dmy;
44  data _null_;
45      date=input('01/31/2003',anydtdte10.);
46      * unambiguous date, so option ignored ;
47      put date=date9.;
48  run;

date=31JAN2003
```

Phil Mason, Wood Street Consultants

Any Room at the Inn?

[SAS 8] Recently at a client site it was requested that we establish how much space was available on the drive within which the **WORK** directories were stored. The attached program was run to establish just this scenario on an NT server.

```
options noxwait noxsync;

* Number of iterations ;
%let num_iter=2;
* Location of file to store the directory listing ;
%let loc=c:\testing.txt;
* Duration of time in which SAS goes to sleep before
  checking again (seconds) ;
%let delay=2;
* Data set to record all records for this series of
  checks ;
%let allrecords=work.all;
* Flag to indicate whether to delete the record of
  all checks if it currently exists
  0 indicates do not delete
  1 indicates delete ;
%let delete=1;

%macro check;
  data _null_;
    path=pathname('work');
    compress_path=compress(path,'\');
    num=length(path)-length(compress_path)+1;
    new_path=substr(path,1,(length(path)-num));
    call symput('dir',new_path);
  run;

  %if &delete=1 %then %do;
    proc datasets lib=%scan(&allrecords,1,'.')
      nolist nodetails;
      delete %scan(&allrecords,2,'.') ;
    quit;
  %end;
```

```

%do num=1 %to &num_iter;
  data _null_;
    x="cd &dir";
    call system(x);
    x="dir / s > &loc";
    call system(x);
  run;

  data work.thistime (keep=used free datetime);
  infile "&loc" lrecl=200 pad missover;
  input text $ 1-200;
  retain lines used free 0;
  if index(text,'Total Files Listed:') then
    lines=1;
  if lines=1 then do;
    if index(upcase(text),'FILE(S)') then
      used=input(scan(text,-2,' '),comma12.);
    if index(upcase(text),'DIR(S)') then
      free=input(scan(text,-3,' '),comma12.);
    end;
    if free ne 0 then do;
      format datetime datetime16.;
      datetime=datetime();
      output;
    end;
  run;

  proc append base=&allrecords
    data=work.thistime force;

  run;

  data _null_;
    rc=sleep(&delay);
  run;
%end;
%mend;

%check;

```

Amadeus Software Consultancy Team

Formats, Options, and Functions

This section is devoted to the description of useful, or unusual, SAS Formats, Options, and Functions.

[SAS 9] Phil Mason has described the new ANYDTDTew informat, but there are also similar informats for time values, ANYDTTMEw.d, and date-time values, ANYDTDTMw.d.

[SAS 6, 8 and 9] There has always been a problem in my mind about the RENAME= dataset option. I think it is because the syntax, to me, is backwards when compared to assignment statements:

```
RENAME=(oldname1=newname1 oldname2=newname2 ..)
```

It is useful to add RENAME= options to either input, or output, SAS tables if there is any likelihood of naming conflicts. Therefore, by including the syntax here, I will now be able to remind myself every time I reread *VIEWS News*!

[SAS 9] There are some new functions that work in a similar way to the INDEX() function, except they find the first position of a particular type of character, e.g. ANYALNUM() looks for alphanumeric characters, ANYALPHA() for alphabetic characters, ANYCNTRL() for control characters, ANYDIGIT() for numeric digits, ANYFIRST() for valid SAS name first characters, ANYGRAPH() for graphics, ANYLOWER() for lower-case alphabetic characters, ANYNAME() for valid SAS name characters, ANYPRINT() for printable characters, ANYPUNCT() for punctuation, ANYSPACE() for blanks, ANYUPPER() for upper-case alphabetic characters, and ANYXDIGIT() for hexadecimal characters that represent a numeric digit.

In Brief

- Back issues of *VIEWS News* are available from the *VIEWS* web site.
- Please note that all the papers presented at the *VIEWS* 2003 Conference are now accessible from a link on the *VIEWS* web site. Also, all conference delegates will be receiving hard-copy and CD-ROM Conference Proceedings by post after the papers have been made available on the web.
- Phil Mason now has a daily SAS tips newsletter, which can be requested by going to www.woodstreet.org.uk and then clicking on Tips. Previous tips can also be found on the web site.
- SAS users looking for contracts, jobs, services and products may find what they are looking for on the Sponsored Links web page at www.hollandnumerics.com/CSP.HTM.

Diary

Are you organising an event that would be of interest to the *VIEWS* readership? Let us know as we are interested in all non-profit making events related to SAS.

June 2003

12 Communicating with SAS (Web, IntraNet, SAS IT and VB), Amadeus Software - FREE!

See the Amadeus Software web site for further details.

17-19 SeUGi, Vienna, Austria

The European SAS User Group International (SeUGi) Conference is held each year and focuses more on the commercial aspects of SAS software.

19 Maximising Strategic Role of Credit Risk Management, Amadeus Software - FREE!

See the Amadeus Software web site for further details.

July 2003

3 Data Mining Techniques and Tools, Amadeus Software - FREE!

See the Amadeus Software web site for further details.

24 Tricks of the Trade, Amadeus Software - FREE!

See the Amadeus Software web site for further details.

August 2003

VIEWS News 23rd edition

5 An Independent Overview of SAS Products, Amadeus Software - FREE!

See the Amadeus Software web site for further details.

September 2003

?? VIEWS Sub-Group Meeting, London

We will be welcoming a Books-by-Users author from the USA to speak to users for the whole day.

11 Changes and Enhancements to SAS Version 9, Amadeus Software - FREE!

See the Amadeus Software web site for further details.

October 2003

23 Tricks of the Trade, Amadeus Software - FREE!

See the Amadeus Software web site for further details.

November 2003

VIEWS News 24th edition

?? VIEWS Sub-Group Meeting, London

We will be welcoming a Books-by-Users author from the USA to speak to users for the whole day.

25 Changes and Enhancements in SAS version 9 for the Applied Statistician, Amadeus Software - FREE!

See the Amadeus Software web site for further details.

December 2003

2 Communicating with SAS (Web, IntraNet, SAS IT and VB), Amadeus Software - FREE!

See the Amadeus Software web site for further details.

February 2004

IEWS News 25th edition

May 2004

IEWS News 26th edition

9-12 **SUGI**, Montreal, Canada

The SAS User Group International (SUGI) Conference is held each year and focuses on the technical aspects of SAS software.

19-21 **IEWS 2004 Conference**, Commonwealth Conference & Events Centre, Kensington, London

The *IEWS* Committee is looking for user papers to be written for the conference on technical SAS-related subjects.

All *IEWS* event information will be posted and/or emailed to registered members of *IEWS*, and will include an event application form. Please send your queries about any *IEWS* events to events@views-uk.org, and don't forget to look at the web site for the latest news.

Contacts

Amadeus Software

Email:

info@amadeus.co.uk

Tel: +44-(0)1993-878287

Web: www.amadeus.co.uk

Richard Carson

Sysware Consulting Group

Email:

richard@syswaregroup.com

Tel: +61-412-836-040

Fax: +64-4-915-1190

Web: www.syswaregroup.com

Phil Mason

Wood Street Consultants Ltd

Email:

Phil@woodstreet.org.uk

Tel: +44-(0)1491-201016

Web: www.woodstreet.org.uk

Steve Morton

Applied System Knowledge Ltd

Email:

steve.morton@appliedsystem.co.uk

Tel: +44-(0)1491-411977

Web: www.appliedsystem.co.uk

SAS UK

Tel: +44-(0)1628-486933

Web: www.sas.com/uk

IEWS

Web Site

www.views-uk.org

Daniel Morris

Chairman

Email:

chair@views-uk.org

Tel: +44-(0)1993-878287

Web: www.amadeus.co.uk

Philip R Holland

Newsletter Editor

Email:

newsletter@views-uk.org

Tel: +44-(0)7714-279085

Fax: +44-(0)1763-244497

Web: www.hollandnumerics.co.uk

***IEWS News** is published quarterly by IIEWS - the UK's independent SAS user group. IIEWS is a not-for-profit organisation. Our Mission Statement and Rules can be viewed on our web site together with a list of current committee members.*