

VIEWS

International **SAS** Programmer Community



News, Hints, Tips and Information for SAS® Users

Issue 30, 2nd Quarter 2005

A new European SAS conference is announced in this issue. It is hoped that PhUSE will provide a platform for the distribution of technical SAS software knowledge, and *VIEWS* is supporting the conference. More details of the organisation can be found in the News and Diary sections below.

While supporting this new European organisation, I would like to welcome back 2 US-based *VIEWS* consultants, LeRoy Bessler and John Adams. I have also contributed an extract from a recent ODS presentation, along with my regular Formats, Options, and Functions section. I hope you will find them interesting.

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 email notification whenever a new issue of the free quarterly VIEWS News is published on the web site, please remember to also include your email address.*

News

PhUSE Conference 2005

Dear *VIEWS* Member,

We'd like to draw your attention to a new SAS-related conference happening this October.

PhUSE is a new organisation with the purpose of creating a European Forum for Programmers in the Pharmaceutical Industry. SAS is sponsoring the event and we expect a significant number of papers on SAS at the conference. The information below provides more detail on this exciting new event, and the latest news can be found on the PhUSE web site, www.phuse.info.

Thank you for your continued support for *VIEWS*.

The VIEWS Committee.

What is PhUSE?

PhUSE stands for "Pharmaceutical Users Software Exchange" and is a non-profit organisation run by volunteers. Its purpose is to create a European forum for programmers in the pharmaceutical industry with an annual multiple day conference.

PhUSE 2005 will be held from October 10-12 2005 in the beautiful historic town of Heidelberg, Germany. This first PhUSE conference will be chaired by Stephen Bamford (Pfizer), who has been the driving force behind organising this event. After an opening session on Monday morning, PhUSE will continue for 2½ days with three parallel streams of presentations. Between the presentations you will have the opportunity to wander around the demo and exhibition area where pharmaceutical companies, CROs, solution providers and SAS will feature their products and solutions.

A special presentation stream "SAS Starters" is for those who are new to working with SAS in the pharmaceutical environment. Audience and presenters have a maximum of 3 years experience working with SAS. On Monday night we will have a big party to celebrate the inception of PhUSE at the famous HAVANA restaurant. Before the closing session on Wednesday, PhUSE will host a panel discussion on "ODS as a reporting tool" with experts from the industry and SAS.

Ask The Consultant

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

Right-Sizing Data Imported in EG from Excel

Q: When I use the Import Data function in Enterprise Guide to get data from Excel, all character columns are defaulted to 255 bytes. How can I easily re-size them to their actual lengths?

A: [SAS 8] To demonstrate the problem, export (Save As) **sashelp.class** to **c:\temp\class.xls**. Then use the Enterprise Guide Import Data function to get the data back, saving it as **work.FromExcel**. During the Import process, you are given the option to override the default length of 255 bytes for character columns, but in a real situation you may encounter numerous character columns for a potentially huge number of rows, with no practical way of guessing their actual length requirements. The solution here was validated in SAS 8, but a demo of Import Data in SAS 9 did not show any relief for this problem.

The most efficient and effective solution is to first determine which columns are character type, then read through the entire data set to determine the maximum length of each, and finally resize the data with that knowledge. This is best handled through a macro. Here is its definition:

```
%MACRO RightSizeColumnsInTable(
    MatchingFormat = YES,
    MatchingInformat = YES,
    DataLib =, Table =,
    RightSizedDataLibTable =);

%LET DataLib = %UPCASE(&DataLib);
%LET Table = %UPCASE(&Table);
```

Capture all the character column names in a SAS data set:

```
PROC SQL;
    CREATE TABLE work.CharacterColumns AS
    SELECT name AS Column FORMAT = $32.
    FROM dictionary.columns
    WHERE libname EQ "&DataLib"
    AND memname EQ "&Table"
    AND type EQ 'char'
    ;
QUIT;
```

Store them as an indexed list of macro variables in the symbol table, along with a count of how many there are:

```
DATA _NULL_;
    SET work.CharacterColumns END=Last;
    CALL SYMPUT('Col' || TRIM(LEFT(_N_)),
        TRIM(LEFT(Column)));
    IF Last;
    CALL SYMPUT('ColCount', _N_);
RUN;
```

Read through the entire data set, retaining the maximum length of each character column. After reading the last observation, store the maximum length of each (and a correspondingly sized format and informat) as an indexed list of macro variables in the symbol table:

```
DATA _NULL_;
%DO i = 1 %TO &ColCount;
    RETAIN CollLength&i 0;
%END;
    SET &DataLib..&Table (KEEP=
%DO i = 1 %TO &ColCount;
        &&Col&i
%END;
    ) END=Last;
%DO i = 1 %TO &ColCount;
    CollLength&i = MAX(CollLength&i,
        LENGTH(&&Col&i));
%END;
    IF Last;
%DO i = 1 %TO &ColCount;
    CALL SYMPUT("CollLength&i", CollLength&i);
    CALL SYMPUT("ColFormat&i",
        '$' || TRIM(LEFT(CollLength&i)) || '.');
    CALL SYMPUT("ColInformat&i",
        '$' || TRIM(LEFT(CollLength&i)) || '.');
%END;
RUN;
```

Create a right-sized data set, by reading the original data set and applying the correct length (and optionally format and informat) to each character column:

```
DATA &RightSizedDataLibTable;
%DO i = 1 %TO &ColCount;
    LENGTH &&Col&i $ &&CollLength&i;
    %IF %UPCASE(&MatchingFormat) EQ YES
    %THEN %DO;
```

```
    FORMAT &&Col&i &&ColFormat&i ;
%END;
    %IF %UPCASE(&MatchingInformat) EQ YES
    %THEN %DO;
    INFORMAT &&Col&i &&ColInformat&i;
%END;
%END;
    SET &DataLib..&Table;
RUN;
%MEND RightSizeColumnsInTable;
```

To see what this complicated-looking code does, invoke the macro so that the resolved code appears in the SAS log:

```
OPTIONS MPRINT; /* show resolved code */
%RightSizeColumnsInTable(DataLib = work,
    Table = FromExcel,
    RightSizedDataLibTable = work.RightSized);
```

From the SAS log, the `PROC SQL` step and the first `DATA _NULL_` step are omitted here since they are not different from the macro definition code, except for resolution of `&DataLib` and `&Table`. Here are the interesting parts of the SAS log:

```
DATA _NULL_;
    RETAIN CollLength1 CollLength2 0;
    SET work.FromExcel (KEEP=Name Sex) END=Last;
    CollLength1 = MAX(CollLength1, LENGTH(Name));
    CollLength2 = MAX(CollLength2, LENGTH(Sex));
    IF Last;
    CALL SYMPUT("CollLength1", CollLength1);
    CALL SYMPUT("ColFormat1",
        '$' || TRIM(LEFT(CollLength1)) || '.');
    CALL SYMPUT("ColInformat1",
        '$' || TRIM(LEFT(CollLength1)) || '.');
    CALL SYMPUT("CollLength2", CollLength2);
    CALL SYMPUT("ColFormat2",
        '$' || TRIM(LEFT(CollLength2)) || '.');
    CALL SYMPUT("ColInformat2",
        '$' || TRIM(LEFT(CollLength2)) || '.');
RUN;

DATA work.RightSized;
    LENGTH Name $ 7;
    FORMAT Name $7. ;
    INFORMAT FORMAT Name $7.;
    LENGTH Sex $ 1;
    FORMAT Sex $1. ;
    INFORMAT Sex $1.;
    SET work.FromExcel;
RUN;
```

The second `DATA _NULL_` step delivers the column names and the lengths, formats, and informats needed to right-size the data in the final `DATA` step.

LeRoy Bessler, Assurant Health

Did You Know?

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

Saving Your Reports for Later

[SAS 8.2, 9] The `ODS DOCUMENT` statement can be used to store ODS output, so that it can be replayed later using the `DOCUMENT` procedure, rather than re-run(!), with any user-specified ODS

style, to any user-specified ODS destination. The code below is flagged as “Experimental” in the SAS 8.2 Log, but will still run. It has “Production” status in SAS 9.1.3.

Code:

Saving your reports to a SAS Item Store.

```
ODS DOCUMENT NAME = work.doc1;

PROC MEANS DATA = sashelp.class;
  CLASS age;
  VAR height weight;
RUN;

PROC TABULATE DATA = sashelp.class;
  CLASS age;
  VAR height weight;
  TABLE (age ALL)
    , (height * MEAN weight * MEAN);
RUN;

ODS DOCUMENT CLOSE;
```

Copying your Item Store file.

```
PROC COPY IN = work OUT = sasuser;
  SELECT doc1 (MEMTYPE = ITEMSTOR);
RUN;
```

Producing reports from an Item Store without having to rerun your report processing.

```
ODS HTML FILE = "c:\temp\document.htm"
  STYLE = Default;

PROC DOCUMENT;
  DOC;
  DOC NAME = work.doc1;
  LIST / LEVELS = ALL;
  REPLAY;
RUN;
QUIT;

ODS HTML CLOSE;
```

Uses:

- All the reports generated by a program run can be stored, prior to printing, in a single file. This file can then be used to generate reports selectively to one or more destinations without the need to re-run the original data processing.
- SAS Item Store files have the “**sas7bitm**” suffix in Windows Explorer and UNIX File Manager.

Drawbacks:

- By default Item Store members cannot be seen in the SAS Explorer window. To allow them to be seen in SAS Explorer in SAS 8.2, carry out the following changes to the Explorer options:
 - (a) Start the SAS Explorer with the **EXPLORER** command, or select **Tools => Options => Explorer**.
 - (b) Select the **Members** tab.
 - (c) Select **ITEMSTOR** in the **Type** list. You may have to check the **View Undefined Types** box first.

(d) Click the **Edit** button.

(e) Click the **Select Icon** button, select the **itemstor** icon in the **Files, Folders, and Reports** group, and click the **OK** button.

(f) Click the **OK** button.

(g) Click the **Close** button.

- To allow them to be seen in SAS Explorer in SAS 9.1.3, carry out the following changes to the Explorer options:
 - (a) Start the SAS Explorer with the **EXPLORER** command, or select **Tools => Options => Explorer**.
 - (b) Select the **Members** tab.
 - (c) Select **ITEMSTOR** in the **Type** list.
 - (d) Click the **Unhide** button, so that it changes to **Hide**. A **Select Icon** windows will appear, select the **itemstor** icon in the **Files, Folders, and Reports** group, and click the **OK** button.
 - (e) Click the **OK** button.
- Currently certain SAS procedures, e.g. **PROC PRINT** and **PROC REPORT**, do not support the DOCUMENT destination. Other SAS procedures, e.g. **PROC FREQ**, can write to the DOCUMENT destination, but reports generated from that Item Store do not look the same as when the reports are sent directly to the final ODS destination. **PROC FREQ** reports from Item Stores look similar to **PROC TABULATE** reports.

Philip R Holland, Holland Numerics

Extracted from the paper "Everyday Uses of ODS" presented by the author at the ODS Meeting of the PSI Statistical Computing SIG in Slough, UK, on 23 March 2005.

Undocumented Logical Operators

[SAS 8] There is a mostly unknown set of logical operators for comparing strings in **PROC SQL**, like the “:” string modifier in the **DATA** step. When two strings equal strings are compared, some unpredictable results may occur. The best way, of course, is to make sure they are of equal lengths. But there is another way.

If I want to check for a prefix of a string, I could use a **SUBSTR()** function to grab the first **n** characters of a string and compare it with the desired prefix. However, it is much easier, and faster, to use one of the “truncated string logical functions”. For example:

```
WHERE variable1 EQT 'sdy'
```

checks if the value of variable1 starts with the character string “sdy”. The “T” (in **EQT**) means to truncate both strengths to that of the shortest one first and then perform an **EQ** comparison. There are a number of other similar functions that truncate first and then compare, e.g. **GET** (**>=**), **GTT** (**>**), **LTT** (**<**), **LET** (**<=**), etc.

John Adams, Boehringer Ingelheim

Formats, Options, and Functions

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

[SAS 8] The **YYQw.** format has been around since before SAS 8, but there are now some new formats for displaying SAS dates as year and quarter, e.g.:

YYQ6. ==> "2005Q2"
YYQC6. ==> "2005:2"
YYQD6. ==> "2005-2"
YYQN6. ==> "20052 "
YYQP6. ==> "2005.2"
YYQS6. ==> "2005/2"

If you want to share a SASUSER library between multiple SAS sessions, then you can use the **RSASUSER** system option, which opens the SASUSER library in read-only mode. The default option is **NORSASUSER**, which opens the SASUSER library in update mode. The **RSASUSER** system option is useful for sites that use a single SASUSER library for all users and want to prevent users from modifying it.

[SAS 9] The **SYMGET()** function returns the character value of a macro variable during DATA step execution, but **SYMGETN()** returns that value as a numeric value.

In Brief

- Back issues of *VEWS News* are available from the *VEWS* web site.
- 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/careers.

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 2005

15 **Special Interest Groups**, SAS UK

See www.sas.com/uk/events for further details.

16 **Changes and Enhancements in SAS Version 9**, Amadeus Software -FREE!

See the Amadeus Software web site, www.amadeus.co.uk, for further details.

21-23 **SAS Forum International**, Lisbon, Portugal

The SAS Forum International is held each year and focuses on the more commercial aspects of SAS software.

September 2005

20 **Data Analytics and Reporting**, Amadeus Software - FREE!

See the Amadeus Software web site, www.amadeus.co.uk, for further details.

October 2005

10-12 **PhUSE Conference 2005**, Heidelberg, Germany

The Pharmaceutical Users Software Exchange (PhUSE) presents the European Forum for programmers in the pharmaceutical industry. See www.phuse.info for the latest news.

November 2005

tba **SAS Forum UK**

The SAS Forum UK is held each year and presents the latest SAS developments to SAS users in the UK. See www.sas.com/uk/events for the latest news.

15 **Tricks of the Trade**, Amadeus Software - FREE!

See the Amadeus Software web site, www.amadeus.co.uk, for further details.

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

Contacts

John Adams
Boehringer Ingelheim,
Ridgefield, Ct, USA

Email:
jadams@boehringer-ingelheim.com

LeRoy Bessler
Assurant Health,
Milwaukee, USA

Email:
bessler@execpc.com

SAS UK

Tel: +44-(0)1628-486933
Web: www.sas.com/uk

VEWS

Web Site

www.views-uk.org

Chairman

Email:
chair@views-uk.org

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

Membership secretary

Email:
membership@views-uk.org

Events secretary

Email:
event@views-uk.org

***VEWS News** is published quarterly by VIEWS International SAS Programmer Community. VIEWS 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.*