Paper 194-25

Pruning the SASLOG – Digging into the Roots of NOTEs, WARNINGs, and ERRORs

Andrew T. Kuligowski – Nielsen Media Research

ABSTRACT

"Look at your SASLOG." You hear it from instructors of courses related to the SAS® System. You read it in papers that are published in the Proceedings of SESUG and other similar conferences. But they never seem to address the next obvious issue - "Then what?" What should you be looking for? And what should you do if you find it?

This presentation will look at one message that is often found in the SASLOG:

```
NOTE: MERGE statement has more than one data set with repeats of
      BY values.
```

We will discuss - and quickly discount - a "bandage" approach that will stop the message from being displayed. More importantly, we will discuss the creation of an ad-hoc routine that will help you understand why the message occurred, and how to revise your code to prevent its recurrence.

MERGE Statement: Repeats of BY Values

Most users of the SAS system have encountered the following message:

```
NOTE: MERGE statement has more than one data set with repeats of
      BY values.
```

There are many papers in the Proceedings from past SUGI, SESUG, and the other various regional SAS User Group conferences that describe how to programatically force such a merge to occur. This assumes that the user *wants* to merge datasets that have repeats of BY values. However, it is also possible that the user did not expect this condition. This implies an error in your SAS routine, caused by a misunderstanding of the input data. We want to isolate these cases and modify our assumptions, so that we can correct our MERGE and eliminate this condition.

The following will illustrate an example of "repeats of BY values". We are going to merge a dataset containing a list of dog breeds [see Table "1-A"] against a dataset containing the dogs owned by sample households [see Table "1-B"] using the variable **BREED**, keeping only those records from the Breed file that have a corresponding record in the Household file. As one would expect (knowing, of course, that this example was created to illustrate the problem at hand), the merge results in "more than one data set with repeats of BY values" [illustrated in Table "1-C"]. The problem is to determine which BY values had repeats, which records (on which files) are affected, and what additional information needs to be included to make the "BY values", also referred to as "merge variables", unique on at least one of the files.

BREED	VARIETY	OTHER INFO
Bulldog		17834
Dalmatian		49235
Dachshund	Mini	18435
Dachshund	MiniLonghair	75846
Dachshund	MiniWirehair	09431
Dachshund	Std	18098
Dachshund	Std Longhair	75324
Dachshund	Std Wirehair	09389
Ger Sheprd		09622
GoldRetrvr		38292
Husky,Sib		75555
Lab Retrvr		38192

Table "1-A" : Breed File

HHLD_ID	BREED	VARIETY	BIRTHDAY	GOAWAYDY
0005884	Dalmatian		07/31/87	
0005884	Dalmatian		12/23/89	
0005884	Bulldog	English	05/19/91	02/20/95
0005884	Dachshund	Std Longhair	09/17/94	
0005884	Dachshund	Std Longhair	10/29/95	
0008824	Ger Sheprd		11/24/89	12/07/92
0008824	Husky,Sib		05/26/93	
0008824	Lab Retrvr		02/28/94	05/06/95
0008824	GoldRetrvr		03/06/95	

Table "1-B" : Dogs per Household File

```
177   DATA DOGDTL;
178      MERGE  DOGOWNED (IN=IN_OWNED)
179             DOGBREED (IN=IN_BREED);
180      BY  BREED ;
181      IF  IN_OWNED ;
182   RUN;
NOTE: MERGE statement has more than one data set with repeats
      of BY values.
NOTE: The data set WORK.DOGDTL has 13 observations and
      6 variables.
```

Table "1-C" : SASLOG - MERGE with "repeats of BY values"

We can use basic elements of the SAS System to do most of the analysis for us; our most powerful tool will be the MEANS procedure. PROC MEANS is traditionally used to compute descriptive statistics for numeric variables in a SAS Data set. In this situation, we simply need a list of the unique values of our merge variable (or, in a more complex case, the unique combinations of values for our merge variables), along with a count of the number of occurrences of each in both of our datasets. The NWAY option on the PROC MEANS statement will limit the output dataset to only those observations with the "highest interaction among CLASS variables" -- that is, only those records containing the unique values or combinations of values for the BY variables will be written to the output dataset. NOPRINT is optional; it can be used or excluded based on personal preference. The variable used in the VAR statement can be any numeric variable in the dataset; the only condition is that it must be a *numeric* variable. Date variables and ID values should be considered, since many / most datasets contain them and they are normally stored as numeric values. (In the rare event your dataset contains exclusively character variables, you will need to add a numeric variable to either the original dataset or a copy of it prior to issuing PROC MEANS.) The OUTPUT statement must specify an OUT= dataset. It must also include the statistics keyword N=, so that a record count -- and only a record count -- for each unique combination of values for the BY variables will be written to each observation of the output dataset. [Table "1-D" illustrates this use of PROC.]

The next step is to take the outputs of our PROC MEANS for each affected dataset and merge *them* together. We *cannot* encounter the "... repeats of BY values" note, since we now only have one observation per unique combination of BY values! Each observation *does* have a count of the number of observations that contain the combination of BY values in their original dataset, stored as _FREQ_ by default. The RENAME= option the datasets in the MERGE statement can be used to give the _FREQ_ variable in each dataset a unique name in our output dataset. (Alternatively, a clearner approach would be to change the **N=** option on the OUTPUT statement in PROC MEANS to **N=*varname***, avoiding the need for the subsequent RENAME.) We will use a subsetting IF, so that we only keep those observations that have multiple occurrences in each input dataset. The output of this step will contain the unique combinations of values that are causing the "...repeats of BY values" note. [The routine is contained in Table "1-E", and the output depicted in Table "1-F".]

```
208  PROC MEANS DATA=DOGOWNED  NOPRINT  NWAY;
209     CLASS  BREED ;
210     VAR HHLD_ID ;
211     OUTPUT  OUT=SUMOWNED  N=CNTOWNED;
212  RUN;

NOTE: The data set WORK.SUMOWNED has 7 observations and
      4 variables.
NOTE: The PROCEDURE MEANS used 0.05 seconds.

213  PROC MEANS DATA=DOGBREED  NOPRINT  NWAY;
214     CLASS  BREED ;
215     VAR OTHRINFO ;
216     OUTPUT  OUT=SUMBREED  N=;
217  RUN;

NOTE: The data set WORK.SUMBREED has 7 observations and
      4 variables.
NOTE: The PROCEDURE MEANS used 0.05 seconds.
```

Table "1-D" : SASLOG - PROC MEANS example

```
585   DATA SUMMERGE (KEEP=BREED  CNTBREED  CNTOWNED);
586      MERGE  SUMBREED (RENAME=(_FREQ_=CNTBREED))
587             SUMOWNED ;
588      BY  BREED ;
589      IF  CNTBREED > 1  AND  CNTOWNED > 1;
590   RUN ;

NOTE: The data set WORK.SUMMERGE has 1 observations and
      3 variables.
```

Table "1-E" : SASLOG - Merging the PROC MEANS outputs

SAS Dataset WORK.SUMMERGE

OBS	BREED	CNTBREED	CNTOWNED
1	Dachshund	6	2

Table "1-F" : Results of Merging the PROC MEANS outputs

Up to this point, we have not discussed one important factor in this analysis - the human element. The process described in this section is meant to be used as a tool to guide the analyst through the unknown elements in their data - once these areas become *known* - there is no need to continue this analysis. The listing of unique combinations of values that occur multiple times in each dataset will often be that stopping point for an analyst. The information obtained will allow them to make modifications to their assumptions and corresponding changes to their routines. However, in the event that the oddities are still not clear, we can employ one or more additional MERGE steps, taking the merged outputs from the PROC MEANS, and merging *those* dataset against each of the original datasets. [Table "1-G" shows how this is done, while Table “1-H” displays the results of this analysis.] This final step should provide sufficient clarification for the analyst to determine which factor(s) are missing in their assumptions and adjust their routines accordingly.

```
599   DATA  CHKOWNED ;
600      MERGE  DOGOWNED (IN=IN_BREED )
601             SUMMERGE (IN=IN_MERGE );
602      BY  BREED ;
603      IF  IN_MERGE ;
604   RUN ;

NOTE: The data set WORK.CHKOWNED has 2 observations and
      7 variables.

605   DATA  CHKBREED ;
606      MERGE  DOGBREED (IN=IN_BREED )
607             SUMMERGE (IN=IN_MERGE );
608      BY  BREED ;
609      IF  IN_MERGE ;
610   RUN ;

NOTE: The data set WORK.CHKBREED has 6 observations and
      5 variables.
```

Table "1-G" : SASLOG - Merging the analysis back to the original input

SAS Dataset WORK.CHKOWNED

OBS	HHLD_ID	BREED	VARIETY	BIRTHDAY	GOTCHADY	CNTBREED	CNTOWNED
1	5884	Dachshund	Std Longhair	12678	12870	6	2
2	5884	Dachshund	Std Longhair	13085	13179	6	2

SAS Dataset WORK.CHKBREED

OBS	BREED	VARIETY	OTHRINFO	CNTBREED	CNTOWNED
1	Dachshund	Mini	1843	6	2
2	Dachshund	MiniLonghair	7584	6	2
3	Dachshund	MiniWirehair	943	6	2
4	Dachshund	Std	1809	6	2
5	Dachshund	Std Longhair	7532	6	2
6	Dachshund	Std Wirehair	938	6	2

Table "1-H" : Results of merging the analysis back to the original input

The end result of this analysis is the discovery that the BY statement in the routine does not contain the proper variables to uniquely identify each record. By adding the extra variable or variables to the original BY statement, the routine works without error – and without the offending NOTE. [Table "1-I" shows the corrected MERGE statement.]

```
682   DATA DOGDTAIL;
683       MERGE  DOGOWNED (IN=IN_OWNED)
684              DOGBREED (IN=IN_BREED);
685       BY  BREED VARIETY;
686       IF  IN_OWNED ;
687   RUN;

NOTE: The data set WORK.DOGDTAIL has 9 observations and
      6 variables.
```

Table "1-I" : SASLOG - MERGE without "repeats of BY values"

CONCLUSION

This paper addressed a message that is commonly found in a SASLOG. It discussed the use of ad hoc routines to explore WHY the message occurred, and covered how to correct a routine to prevent the recurrence of the message. It is hoped that the mechanisms discussed in this paper might be used by the readers in their daily jobs. However, this paper is a failure -- at least in part -- if the process stops there. It is hoped, even more strongly, that the *concepts* of developing and using ad hoc routines to fully understand data are the *true* lessons that the reader retains from this paper.

REFERENCES / FOR FURTHER INFORMATION

Kuligowski, Andrew T. (1996), "Software Validation and Testing". *Proceedings of the Twenty-First Annual SAS Users Group International Conference*. Cary, NC: SAS Institute, Inc.

SAS Institute, Inc. (1990), *SAS Language: Reference, Version 6, First Edition.* Cary, NC: SAS Institute, Inc.

SAS Institute, Inc. (1994), *SAS Software: Abridged Reference, Version 6, First Edition.* Cary, NC: SAS Institute, Inc.

The author can be contacted via e-mail at:
kuligowski@compaq.net or
kuligoat@tvratings.com