

## Cleaning up your SAS log: Overwritten Variable Info Messages

Jennifer Srivastava, Quintiles Transnational Corporation, Durham, NC

### ABSTRACT

As a SAS programmer, you probably spend some of your time reading and possibly creating specifications. Your job also includes writing and testing SAS code to produce the final product, whether it is SDTM datasets, ADaM datasets or statistical outputs such as tables, listings or figures. You reach the point where you have completed the initial programming, removed all obvious errors and warnings from your SAS log and checked your outputs for accuracy. You are almost done with your programming task, but one important step remains.

It is considered best practice to check your SAS log for any questionable messages generated by the SAS system. In addition to messages that begin with the words WARNING or ERROR, there are also messages that begin with the words NOTE or INFO. This paper will focus on the overwritten variable INFO message that commonly appears in the SAS log, and will present different scenarios associated with this message and ways to remove the message from your log, if necessary.

### INTRODUCTION

The SAS log is a record of what happens when you run your SAS program and is an essential tool for debugging code. The log includes program statements, as well as messages generated by SAS. These messages can begin with the words WARNING, ERROR, NOTE or INFO. This paper will explore in detail some of the reasons that cause the overwritten variable INFO message to appear in the SAS log, and will give suggestions on how to remove this message from the log, if this is desired.

In order to get these messages to show up in your SAS log, you need to adjust the OPTIONS statement as follows. To display the INFO messages in your log, specify OPTIONS MSGLEVEL=I. Below is an example of one way to ensure that your log will display the INFO messages that will be discussed in this paper.

```
OPTIONS MSGLEVEL=I;
```

### OVERWRITTEN VARIABLE INFO MESSAGES

Below is an example of an overwritten variable info message.

```
INFO: The variable VAR on data set WORK.ONE will be overwritten by data set WORK.TWO.
```

This message occurs when you merge two datasets that contain the same variable. If the variable that occurs in both datasets is not in the BY statement, the value of the variable in the second dataset will overwrite the value of the variable in the first dataset. Referring to the code below and the INFO: message above, you can conclude that both datasets ONE and TWO contain the variable VAR. The dataset ALL will have the value of VAR that came from dataset TWO.

```
data all;
merge one
      two;
by subjid;
run;
```

Depending on your company's standards, the best practice may be to avoid having this message in your SAS log. Another viable option is to add a note to the log explaining that this message is acceptable and expected, once you determine that is indeed the case. Below are examples of five different scenarios where you may get an overwritten INFO: message in the SAS log and some suggestions on how to deal with them.

## SCENARIO 1: MERGING ON BASELINE FLAGS

The sample data used in this paper is based on a fictitious study testing whether a diet dog food is successful in helping canine subjects lose weight. As is often the case in clinical trial studies, baseline flags are necessary to determine change from baseline statistics. Below are two screen shots showing the SAS log where the baseline flags are being merged onto the main dataset, which is called DOG. In Display 1, you can see that all variables from the DOG file are being kept in the BASELINE dataset. When they are merged back onto the DOG file, the baseline values for the variables NAME, BREED, BIRTHDT, VISITDT, GENDER and VISIT write over the same variables on the DOG file, which is incorrect. Display 2 shows the correct way to do this, which is to keep only the necessary variables on BASELINE to avoid mistakenly overwriting variables on DOG.

```

139      * Get the baseline weight from visit 1 *;
140      DATA BASELINE (RENAME=(WEIGHT=WT_BL));
141          SET DOG;
142          IF VISIT=1;
143      RUN;

NOTE: There were 90 observations read from the data set WORK.DOG.
NOTE: The data set WORK.BASELINE has 30 observations and 8 variables.
NOTE: Compressing data set WORK.BASELINE increased size by 100.00 percent.
      Compressed is 2 pages; un-compressed would require 1 pages.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.01 seconds

144
145      * Merge the baseline data onto the main file *;
146      DATA DOG;
147          MERGE DOG
148              BASELINE;
149          BY SUBJID;
150      RUN;

INFO: The variable NAME on data set WORK.DOG will be overwritten by data set WORK.BASELINE.
INFO: The variable BREED on data set WORK.DOG will be overwritten by data set WORK.BASELINE.
INFO: The variable BIRTHDT on data set WORK.DOG will be overwritten by data set WORK.BASELINE.
INFO: The variable VISITDT on data set WORK.DOG will be overwritten by data set WORK.BASELINE.
INFO: The variable GENDER on data set WORK.DOG will be overwritten by data set WORK.BASELINE.
INFO: The variable VISIT on data set WORK.DOG will be overwritten by data set WORK.BASELINE.
NOTE: There were 90 observations read from the data set WORK.DOG.
NOTE: There were 30 observations read from the data set WORK.BASELINE.
NOTE: The data set WORK.DOG has 90 observations and 9 variables.

```

### Display 1. Merging on baseline data incorrectly

```

159      * Get the baseline data from visit 1 and only keep needed variables *;
160      DATA BASELINE (KEEP=SUBJID WEIGHT RENAME=(WEIGHT=WT_BL));
161          SET DOG;
162          IF VISIT=1;
163      RUN;

NOTE: There were 90 observations read from the data set WORK.DOG.
NOTE: The data set WORK.BASELINE has 30 observations and 2 variables.
NOTE: Compressing data set WORK.BASELINE increased size by 100.00 percent.
      Compressed is 2 pages; un-compressed would require 1 pages.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds

164
165      * Merge the baseline data onto the main file *;
166      DATA DOG;
167          MERGE DOG
168              BASELINE;
169          BY SUBJID;
170      RUN;

NOTE: There were 90 observations read from the data set WORK.DOG.
NOTE: There were 30 observations read from the data set WORK.BASELINE.
NOTE: The data set WORK.DOG has 90 observations and 9 variables.

```

### Display 2. Merging on baseline data correctly

## SCENARIO 2: MERGING OR SETTING DATA TOGETHER

There may be situations where you have mutually exclusive subsets of data. For example, usually subjects can be classified as either male or female. If we have one dataset for each gender and desire to combine them back into one dataset containing all subjects, we could merge or set the data together. In both cases, we will get the same ALLDOGS dataset. However setting the data together is preferable because there will be no overwritten messages showing up in the log.

```

186      * Merge together mutually exclusive datasets *;
187      DATA ALLDOGS;
188          MERGE MALE
189              FEMALE;
190          BY SUBJID;
191      RUN;

```

INFO: The variable NAME on data set WORK.MALE will be overwritten by data set WORK.FEMALE.  
INFO: The variable BREED on data set WORK.MALE will be overwritten by data set WORK.FEMALE.  
INFO: The variable BIRTHDT on data set WORK.MALE will be overwritten by data set WORK.FEMALE.  
INFO: The variable VISITDT on data set WORK.MALE will be overwritten by data set WORK.FEMALE.  
INFO: The variable GENDER on data set WORK.MALE will be overwritten by data set WORK.FEMALE.  
INFO: The variable WEIGHT on data set WORK.MALE will be overwritten by data set WORK.FEMALE.  
INFO: The variable VISIT on data set WORK.MALE will be overwritten by data set WORK.FEMALE.  
INFO: The variable WT\_BL on data set WORK.MALE will be overwritten by data set WORK.FEMALE.  
NOTE: There were 18 observations read from the data set WORK.MALE.  
NOTE: There were 12 observations read from the data set WORK.FEMALE.  
NOTE: The data set WORK.ALLDOGS has 30 observations and 9 variables.

Display 3. Merging mutually exclusive data together

```

193      * Set together mutually exclusive datasets *;
194      DATA ALLDOGS;
195          SET MALE
196              FEMALE;
197          BY SUBJID;
198      RUN;

```

NOTE: There were 18 observations read from the data set WORK.MALE.  
NOTE: There were 12 observations read from the data set WORK.FEMALE.  
NOTE: The data set WORK.ALLDOGS has 30 observations and 9 variables.

Display 4. Setting mutually exclusive data together

## SCENARIO 3: USING UPDATE TO COMBINE DATASETS

You may come across situations where you want to write over some of the data, but do not want to replace nonmissing data with missing data in the process. For instance, the programmer receives additional lab data for some of the subjects. She wants to use the most recent lab data, but doesn't want to write over the original lab data for dogs that do not have new data. She contemplates using the update statement to combine the two datasets instead of the merge statement. With the merge statement she gets the overwritten message in the log for two variables, WEIGHT and LBSTRESN. With the update statement, she does not get any overwritten messages. However, both WEIGHT and LBSTRESN are being overwritten unless those variables are missing on LAB\_NEW. Update can be used in place of merge to remove overwritten messages from the log. However, please be aware that some overwriting will occur anyway, and that update handles missing data differently from merge. Also, update can be used to combine only two datasets, while merge can combine as many datasets as memory will permit.

Obs	SUBJID	WEIGHT	LB CAT	LBTEST	LBSTRESN
1	1	19	CHEMISTRY	CHOLESTEROL, TOTAL	99
2	2	.	CHEMISTRY	CHOLESTEROL, TOTAL	.
3	3	59	CHEMISTRY	CHOLESTEROL, TOTAL	200
4	4	35	CHEMISTRY	CHOLESTEROL, TOTAL	77
5	5	67	CHEMISTRY	CHOLESTEROL, TOTAL	.
6	6	38	CHEMISTRY	CHOLESTEROL, TOTAL	79
7	7	.	CHEMISTRY	CHOLESTEROL, TOTAL	200
8	8	8	CHEMISTRY	CHOLESTEROL, TOTAL	100
9	9	170	CHEMISTRY	CHOLESTEROL, TOTAL	250
10	10	25	CHEMISTRY	CHOLESTEROL, TOTAL	77

Display 5. LAB\_NEW data that will be combined with LAB\_VI

Cleaning up your SAS® log: Overwritten Variable Info Messages, continued

LAB\_V1 (1st 10 obs), before combining with LAB\_NEW

Obs	SUBJID	NAME	BREED	WEIGHT	LBCAT	LBTEST	LBSTRESN
1	1	Wishbone	Jack Russell Terrier	20	CHEMISTRY	CHOLESTEROL, TOTAL	100
2	2	Elvis	Bloodhound	110	CHEMISTRY	CHOLESTEROL, TOTAL	200
3	3	Balto	Siberian Husky	59	CHEMISTRY	CHOLESTEROL, TOTAL	200
4	4	Snoopy	Beagle	35	CHEMISTRY	CHOLESTEROL, TOTAL	77
5	5	Lassie	Collie	67	CHEMISTRY	CHOLESTEROL, TOTAL	150
6	6	Brugger	Bagle	39	CHEMISTRY	CHOLESTEROL, TOTAL	80
7	7	Snoop	German Shepherd	89	CHEMISTRY	CHOLESTEROL, TOTAL	200
8	8	Boo	Pomeranian	8	CHEMISTRY	CHOLESTEROL, TOTAL	100
9	9	Scooby	Great Dane	170	CHEMISTRY	CHOLESTEROL, TOTAL	250
10	10	Lady	Cocker Spaniel	25	CHEMISTRY	CHOLESTEROL, TOTAL	77

Display 6. LAB\_VI before being combined with NEW\_LAB

LAB\_V1 (1st 10 obs), after merging with LAB\_NEW

Obs	SUBJID	NAME	BREED	WEIGHT	LBCAT	LBTEST	LBSTRESN
1	1	Wishbone	Jack Russell Terrier	19	CHEMISTRY	CHOLESTEROL, TOTAL	99
2	2	Elvis	Bloodhound	.	CHEMISTRY	CHOLESTEROL, TOTAL	.
3	3	Balto	Siberian Husky	59	CHEMISTRY	CHOLESTEROL, TOTAL	200
4	4	Snoopy	Beagle	35	CHEMISTRY	CHOLESTEROL, TOTAL	77
5	5	Lassie	Collie	67	CHEMISTRY	CHOLESTEROL, TOTAL	.
6	6	Brugger	Bagle	38	CHEMISTRY	CHOLESTEROL, TOTAL	79
7	7	Snoop	German Shepherd	.	CHEMISTRY	CHOLESTEROL, TOTAL	200
8	8	Boo	Pomeranian	8	CHEMISTRY	CHOLESTEROL, TOTAL	100
9	9	Scooby	Great Dane	170	CHEMISTRY	CHOLESTEROL, TOTAL	250
10	10	Lady	Cocker Spaniel	25	CHEMISTRY	CHOLESTEROL, TOTAL	77

Display 7. Using MERGE to combine datasets and replacing non missing data with missing data

LAB\_V1 (1st 10 obs), after updating with LAB\_NEW

Obs	SUBJID	NAME	BREED	WEIGHT	LBCAT	LBTEST	LBSTRESN
1	1	Wishbone	Jack Russell Terrier	19	CHEMISTRY	CHOLESTEROL, TOTAL	99
2	2	Elvis	Bloodhound	110	CHEMISTRY	CHOLESTEROL, TOTAL	200
3	3	Balto	Siberian Husky	59	CHEMISTRY	CHOLESTEROL, TOTAL	200
4	4	Snoopy	Beagle	35	CHEMISTRY	CHOLESTEROL, TOTAL	77
5	5	Lassie	Collie	67	CHEMISTRY	CHOLESTEROL, TOTAL	150
6	6	Brugger	Bagle	38	CHEMISTRY	CHOLESTEROL, TOTAL	79
7	7	Snoop	German Shepherd	89	CHEMISTRY	CHOLESTEROL, TOTAL	200
8	8	Boo	Pomeranian	8	CHEMISTRY	CHOLESTEROL, TOTAL	100
9	9	Scooby	Great Dane	170	CHEMISTRY	CHOLESTEROL, TOTAL	250
10	10	Lady	Cocker Spaniel	25	CHEMISTRY	CHOLESTEROL, TOTAL	77

Display 8. Using UPDATE to combine datasets and avoid replacing non missing data with missing data

## SCENARIO 4: ADDING MORE VARIABLES TO THE BY STATEMENT

One relatively easy way to avoid having overwritten messages appear in the SAS log is to include as many variables as needed in the by statement when you are merging multiple datasets. Displays 7 and 8 give examples of this scenario. Be sure that any variables you add to the by statement are at the appropriate level. For example, if you are merging by subject, be sure all variables in the by statement are subject-level variables.

```
255      DATA DOG;
256      MERGE DOG
257          BASELINE2;
258      BY SUBJID;
259      RUN;
```

INFO: The variable **NAME** on data set WORK.DOG will be overwritten by data set WORK.BASELINE2.  
NOTE: There were 90 observations read from the data set WORK.DOG.  
NOTE: There were 30 observations read from the data set WORK.BASELINE2.  
NOTE: The data set WORK.DOG has 90 observations and 9 variables.

### Display 9. Merging data together without using enough variables in the by statement

```
267      * Add another subject-level variable to the by statement *;
268      DATA DOG;
269      MERGE DOG
270          BASELINE2;
271      BY SUBJID NAME;
- 272      RUN;
```

NOTE: There were 90 observations read from the data set WORK.DOG.  
NOTE: There were 30 observations read from the data set WORK.BASELINE2.  
NOTE: The data set WORK.DOG has 90 observations and 9 variables.

### Display 10. Merging data together using enough variables in the by statement

## SCENARIO 5: THE MESSAGE IS ACCEPTABLE

There are times when the programmer decides that the overwritten message is expected and acceptable, and due to time constraints he/she may decide to keep them in the log, and will add a note to the log to document this. The additional documentation is helpful, so that there will not be doubts later on about whether the code is correct or not. Using the same example that was used for Scenario 4, the programmer decides that it is acceptable to merge the subject name from BASELINE2 onto DOG. She leaves the overwritten message in the log and adds a note to the log as shown below.

```
280      DATA DOG;
281      MERGE DOG
282          BASELINE2;
283      BY SUBJID;
- 284      RUN;
```

INFO: The variable **NAME** on data set WORK.DOG will be overwritten by data set WORK.BASELINE2.  
NOTE: There were 90 observations read from the data set WORK.DOG.  
NOTE: There were 30 observations read from the data set WORK.BASELINE2.  
NOTE: The data set WORK.DOG has 90 observations and 9 variables.  
NOTE: DATA statement used (Total process time):  
real time 0.00 seconds  
cpu time 0.00 seconds

---

```
285
286      %PUT %STR(PROGRAMMING NOTE:OK IF DOG(NAME) WILL BE OVERWRITTEN BY BASELINE2(NAME).);
PROGRAMMING NOTE:OK IF DOG(NAME) WILL BE OVERWRITTEN BY BASELINE2(NAME).
```

### Display 11. Adding a note to the log saying that the message is acceptable

## **CONCLUSION**

Hopefully the information provided in this paper will give you some insight into why Overwritten Variable Info Messages may be showing up in your SAS log and also give you some ideas how to remove them, or how to document in your log that they are acceptable if you decide that is the case. Having a SAS log that is clean and free of extra messages will help the programmer produce a higher quality final product, whether it is an SDTM dataset, ADaM dataset or a statistical output such as a table, listing or figure.

## **CONTACT INFORMATION**

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

Jennifer Srivastava  
Quintiles Transnational Corporation  
5927 S Miami Blvd  
Morrisville, NC 27560  
919-749-8567  
Jennifer.srivastava@quintiles.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.