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.

Display 1. Merging on baseline data incorrectly

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

Display 2. Merging on baseline data correctly

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

**Display 3. Merging mutually exclusive data together**

```
166 * Merge together mutually exclusive datasets *;
167 DATA ALLDOGS;
168 MERGE MALE;
169 FEMALE;
170 BY SUBJID;
171 RUN;
```

INFO: The variable NAME on data set WORK.MALE will be overwritten by data set WORK.FEMALE.
INFO: The variable BRED on data set WORK.MALE will be overwritten by data set WORK.FEMALE.
INFO: The variable BIRTH 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 W7_DL on data set WORK.MALE will be overwritten by data set WORK.FEMALE.
NOTE: There were 19 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**

```
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.

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.

**Display 5. LAB_NEW data that will be combined with LAB_VI**

```
LAB_NEW
Obs SUBJID WEIGHT LOCAL LPTEST LBSTRESN
1 1 19 CHEMISTRY CHOLESTEROL TOTAL 99
2 2 59 CHEMISTRY CHOLESTEROL TOTAL 200
3 3 35 CHEMISTRY CHOLESTEROL TOTAL 77
4 4 67 CHEMISTRY CHOLESTEROL TOTAL 79
5 5 30 CHEMISTRY CHOLESTEROL TOTAL 200
6 6 8 CHEMISTRY CHOLESTEROL TOTAL 100
7 7 170 CHEMISTRY CHOLESTEROL TOTAL 250
8 8 25 CHEMISTRY CHOLESTEROL TOTAL 77
9 9 10 CHEMISTRY CHOLESTEROL TOTAL 250
10 10 26 CHEMISTRY CHOLESTEROL TOTAL 77
```
Cleaning up your SAS® log: Overwritten Variable Info Messages, continued

### Display 6. LAB_VI before being combined with NEW_LAB

<table>
<thead>
<tr>
<th>Obs</th>
<th>SUBJID</th>
<th>NAME</th>
<th>BREED</th>
<th>WEIGHT</th>
<th>LOCAT</th>
<th>LBTEST</th>
<th>LBSTRESH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Wishbone</td>
<td>Jack Russell Terrier</td>
<td>20</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>109</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Elvis</td>
<td>Bloodhound</td>
<td>110</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>209</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Balto</td>
<td>Siberian Husky</td>
<td>89</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>209</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>Snoopy</td>
<td>Beagle</td>
<td>35</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>77</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Lassie</td>
<td>Collie</td>
<td>67</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>150</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>Brugger</td>
<td>Eagle</td>
<td>39</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>Snoop</td>
<td>German Shepherd</td>
<td>89</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>209</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>Moo</td>
<td>Pommeranian</td>
<td>8</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>100</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>Scooby</td>
<td>Great Dane</td>
<td>170</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>250</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>Lady</td>
<td>Cocker Spaniel</td>
<td>25</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>77</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Obs</th>
<th>SUBJID</th>
<th>NAME</th>
<th>BREED</th>
<th>WEIGHT</th>
<th>LOCAT</th>
<th>LBTEST</th>
<th>LBSTRESH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Wishbone</td>
<td>Jack Russell Terrier</td>
<td>19</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>99</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Elvis</td>
<td>Bloodhound</td>
<td></td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>.</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Balto</td>
<td>Siberian Husky</td>
<td>89</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>200</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>Snoopy</td>
<td>Beagle</td>
<td>35</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>77</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Lassie</td>
<td>Collie</td>
<td>67</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>.</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>Brugger</td>
<td>Eagle</td>
<td>38</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>79</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>Snoop</td>
<td>German Shepherd</td>
<td></td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>200</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>Boo</td>
<td>Pommeranian</td>
<td>8</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>100</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>Scooby</td>
<td>Great Dane</td>
<td>170</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>250</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>Lady</td>
<td>Cocker Spaniel</td>
<td>25</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>77</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Obs</th>
<th>SUBJID</th>
<th>NAME</th>
<th>BREED</th>
<th>WEIGHT</th>
<th>LOCAT</th>
<th>LBTEST</th>
<th>LBSTRESH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Wishbone</td>
<td>Jack Russell Terrier</td>
<td>19</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>99</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Elvis</td>
<td>Bloodhound</td>
<td>110</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>200</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Balto</td>
<td>Siberian Husky</td>
<td>59</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>200</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>Snoopy</td>
<td>Beagle</td>
<td>35</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>77</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Lassie</td>
<td>Collie</td>
<td>67</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>150</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>Brugger</td>
<td>Eagle</td>
<td>38</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>79</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>Snoop</td>
<td>German Shepherd</td>
<td>89</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>200</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>Boo</td>
<td>Pommeranian</td>
<td>8</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>100</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>Scooby</td>
<td>Great Dane</td>
<td>170</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>250</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>Lady</td>
<td>Cocker Spaniel</td>
<td>25</td>
<td>CHEMISTRY</td>
<td>CHOLESTEROL, TOTAL</td>
<td>77</td>
</tr>
</tbody>
</table>
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.

```
DATA DOG;
MERGE DOG BASELINE2;
BY SUBJID;
RUN;
```

NOTE: The variable NAME on data set WORK.DOG will be overwritten by data set WORK.BASLINE2.

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.

```
DATA DOG;
MERGE DOG BASELINE2;
BY SUBJID;
RUN;
```

INFO: The variable NAME on data set WORK.DOG will be overwritten by data set WORK.BASELINE2.

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

```
DATA DOG;
MERGE DOG BASELINE2;
BY SUBJID NAME;
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

```
DATA DOG;
MERGE DOG BASELINE2;
BY SUBJID;
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
```

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.