

## Paper CC03

# Merging: Avoid the Cones

Jessica Bonow, Arkansas Foundation for Medical Care, Little Rock, AR

### ABSTRACT

Travelers around the United States drive through many construction zones. Informational road signs tell drivers what to expect as they approach the work area. One of the signs may read, "Right Lane Closed, Merge Now". In this illustration, the two lanes of the road become one. It is the same with data sets that SAS® users work with. They want to combine observations from two or more data sets to form one single observation in a new data set. This can be accomplished by using the match-merge method, one of four ways to combine data sets. The match-merge method may have some cones that the SAS user has to avoid in order to produce the desired results that are needed. This paper describes the basics of the match-merge method, examines some of the cones users may drive into, and gives solutions to avoid the cones.

The SAS tools used throughout the paper are in BASE SAS. The operating system used is PC SAS, but could be used with any operating system to achieve similar results. A beginner's skill level should be considered for this paper.

### COMBINING DATA SETS

Combining data sets may be a common practice for BASE SAS users, but you must know the essentials before jumping into it. Some of the essentials include knowing how you want your output data set to look like, determining how the input data sets are alike, performing intermediate steps before combining, selecting the appropriate method for combining, and finally selecting the SAS tools to complete the task.

Relationships between data sets exist and must be identified in order to receive the desired results when combining data sets. All data falls into one of the following categories:

**One to one** – a single observation in one data set is related to a single observation from another data set.

**1 to Many or Many to 1** – one data set has at most one observation with a specific value of the selected variable while the another data set has more than one observation with a specific value of the selected variable or vice versa

**Many to Many** – multiple observations from each input data set may be related based on values of one or more common variables.

Performing some intermediate steps may involve sorting the input data sets or creating indexes for the variables. Once you have determined what your output data set is suppose to look like, choosing which method you use for combining the data sets is one of six methods: concatenating, interleaving, one-to-one reading, one-to-one merging, match merging, or updating. The focus of this paper is on the match merging method. Performing a match merge means to combine observations from two or more data sets into a single observation in a new data set based on the values of one or more common variables. For example, merge DATA1 data set with DATA2 data set to get a new data set.

| Data1 |   |      | Data2 |   |      |
|-------|---|------|-------|---|------|
| Obs   | i | varx | Obs   | i | vary |
| 1     | a | x1   | 1     | a | y1   |
| 2     | b | x2   | 2     | b | y2   |
| 3     | c | x3   | 3     | c | y3   |
| 4     | d | x4   | 4     | d | y4   |

Several key things to remember when using the match merging method are the number of observations in the new data set is the sum of the largest number of observations of the input in each BY group in all data sets and all input data sets must be sorted by the BY variable or they must have an index.

Using the right tools, the MERGE statement along with the BY statement, you will be on your way to achieving the desired results. Below is the syntax for the match merge:

```
DATA data-set;
  MERGE data-set(s);
  BY variable(s);
RUN;
```

The data sets in the MERGE statement are names of at least two data sets from which observations are read and the variable in the BY statement is the name of each variable by which the data sets are to be sorted or indexeded, reflecting how the data sets are related.

### WHAT IS GOING ON IN THE DATA STEP?

The DATA step does processes in order to complete the match merging of data sets. If you want to merge DATA1 and DATA2 from above by the common variable ID and create a new data set COMBINE, the SAS code would look like

```
DATA COMBINE;
  MERGE DATA1 DATA2;
  BY ID;
RUN;
```

The first step that SAS does is reads the information from each of the data sets that is in the merge statement creating a data vector containing all the variables from all of the data sets mentioned. The first.variable and last.variable are also created for the BY variable.

Step two of the process is SAS looks at the first BY variable in each of the data sets to determine which of them should appear first in the output data set, COMBINE. In the meantime, the DATA step reads into the data vector the first observation of the BY variable from each data set in the MERGE statement in the order that they appear in the statement. If one of the data sets does not contain observations in that BY variable, the data vector contains missing values for the variable unique to that data set. Therefore, the data vector would contain this observation.

| Data Vector |    |      |      |
|-------------|----|------|------|
| Obs         | id | varx | vary |
| 1           | a  | x1   | y1   |

Step three is that after processing the first observation for the last data set in the MERGE statement, SAS writes the contents of the data vector into the output data set and continues to merge all observations until it writes all observations from the first BY group to the output data set. Once this is complete for the first BY group, SAS put all variables in the data vector to missing.

| Data Vector |    |      |      | COMBINE |    |      |      |
|-------------|----|------|------|---------|----|------|------|
| Obs         | id | varx | vary | Obs     | id | varx | vary |
| 1           | .  | .    | .    | 1       | a  | x1   | y1   |

Then it looks at the next BY group in each data set to determine which BY group is next to appear in the output data set. Finally, SAS repeats steps two and three until all observations are read from all BY groups in all the data sets, which results in an output data set.

| COMBINE |    |      |      |
|---------|----|------|------|
| Obs     | id | varx | vary |
| 1       | a  | x1   | y1   |
| 2       | b  | x2   | y2   |
| 3       | c  | x3   | y3   |

NOTE: When duplicate values of the BY group in a data set exist, SAS reads the last observation for a BY group in one data set and retains the values in the data vector for all variables that are unique to that data set until all observations for that BY group have been read from all data sets. With mismatched observations, SAS retains the values of all variables in the data vector even if the value is missing. Missing values happen when variables are in one data set but not in the other.

## CONES AND SOLUTIONS

As you drive along in programming and decide you would like to merge, you must know your surroundings and take the necessary precautions to avoid the cones in the way. These cones may be fairly noticeable or they may go unrecognized. Let us examine five cones and their solutions before you run them over in other programming you do.



### CONE ONE

The first cone that we come upon is a missing BY statement. If you use the following data sets, the code would like

```
DATA COMBINE;
  MERGE ONE TWO;
RUN;
```

| ONE |    |       |           | TWO |    |        |     | COMBINE |    |       |           |        |     |
|-----|----|-------|-----------|-----|----|--------|-----|---------|----|-------|-----------|--------|-----|
| Obs | id | Name  | County    | Obs | id | Gender | Age | Obs     | id | Name  | County    | Gender | Age |
| 1   | 1  | Joe   | Jefferson | 1   | 1  | M      | 5   | 1       | 1  | Joe   | Jefferson | M      | 5   |
| 2   | 2  | Susie | Winona    | 2   | 2  | F      | 17  | 2       | 2  | Susie | Winona    | F      | 17  |
| 3   | 5  | Bill  | Gilmore   | 3   | 6  | M      | 24  | 3       | 6  | Bill  | Gilmore   | M      | 24  |
| 4   | 9  | Jane  | Huff      | 4   | 9  | F      | 9   | 4       | 9  | Jane  | Huff      | F      | 9   |

This cone results in a one-to-one merge. ID 6 overwrites ID 5 because it came second in the order of merging. No warning message is given in the log file so you will think that it is doing what you want it to do. The solution is to add a BY statement after the MERGE statement in order to avoid the cone.



### CONE TWO

The second cone you are driving by is called the BY variable is missing. In this case, you have a BY statement so you avoided cone #1, but one or more of the BY variables in the data set you want to merge is missing. So when you combine the data sets the MERGE statement just combines them and they may be the same person or they may be totally different. To avoid this, make sure all observations of the BY variable have a valid value. Writing some SAS code that checks for missing values may be a way to handle this if your data sets contain numerous observations.

```
DATA COMBINE;
  MERGE ONE TWO;
  BY ID;
  IF ID= ' ' then put "Missing Value of BY Variable"; *for Character type BY Variable;
  IF ID=. then put "Missing Value of BY Variable"; *for Numeric type BY Variable;
RUN;
```



### CONE THREE

Using the two data sets ONE and TWO above, the BY variable ID in ONE is defined as a character type and the BY variable ID in TWO is defined as a numeric type. Can this merge take place? NO, it will result in an error message in the log file.

**ERROR: Variable id has been defined as both character and numeric.**

One solution to avoid cone 3 is to know the defined formats of the BY variable before trying to merge data sets. Another solution is to convert one of the BY variables so that it is consistent with the type of the other prior to merging data sets. Below is some straightforward code to convert types when dealing with character and numeric variable types.

Character to Numeric: NewID = ID +0; or NewID = input(ID, format);

Numeric to Character: NewID = put(ID, format);

#### CONE FOUR



Multiple lengths of the BY variable is cone number 4. In this case, the length of the BY variable in the first data set is shorter than the length of the BY variable in the second data set mentioned in the MERGE statement. As the DATA step begins the merging process, SAS assigns the length of the BY variable in the output data set according to the length specified in the first data set. Any other length size mentioned could not alter the existing one that was assigned. SAS could produce a truncation problem as well as others. Additional information about this cone can be found in the paper entitled "Danger: MERGE Ahead! Warning: BY Variable with Multiple Lengths" by Bob Virgile. Two quick solutions to avoid this cone are adding a length statement before the MERGE statement or changing the order of the data sets in the MERGE statement so the data set with the longer BY variable is first.

#### CONE FIVE



Complex merges exist so proceed with caution. A couple of them are many-to-many match merges and few-to-many match merges. When doing a many-to-many match merge, you want to combine data sets that contain multiple observations with the same BY variable. When you run the data step, a note will appear in the log file. It reads

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

The same thing happens when you run a few-to-many match merge. In one of the data sets there is less observations with the same BY variable when compared to the other data set.

For example,

| DATA1 |    |      | DATA2 |    |      | COMBINED |    |      |      |
|-------|----|------|-------|----|------|----------|----|------|------|
| Obs   | id | varx | Obs   | id | vary | Obs      | id | varx | vary |
| 1     | 1  | 2    | 1     | 1  | 1    | 1        | 1  | 2    | 1    |
| 2     | 1  | 4    | 2     | 1  | 3    | 2        | 1  | 4    | 3    |
|       |    |      | 3     | 1  | 5    | 3        | 1  | 4    | 5    |



The note in the log file means that you may not be getting the desired results and be merging haphazardly. The solution to get the expected results is to pick another option depending on your data or possibly pick a BY variable that is unique to an observation.

#### CONCLUSION

I hope this paper helps you understand what processes SAS performs in a match merge and gives you basic information to carefully program them in your work without any error or warning messages in the log file or undesired results in your output data set. Many different merge scenarios exist, so remember avoid the big orange cones by keeping the merge short and simple.

#### REFERENCES

SAS Institute Inc. (1999), *SAS OnlineDoc®*, Version 8, Cary, NC: SAS Institute Inc.

Virgile, Bob. "Danger: MERGE Ahead! Warning: BY Variable with Multiple Lengths!" *Proceeding of the Twenty-Eighth SAS® Users Group International Conference*. March 2003. <http://www2.sas.com/proceedings/sugi28/098-28.pdf>

## **CONTACT INFORMATION**

Your comments and questions are valued and encouraged.

Contact the author at:

Jessica Bonow  
Arkansas Foundation for Medical Care  
401 W Capitol Suite 508  
Little Rock, AR 72211  
Work Phone: (501) 375-5700 ext 660  
Fax: (501) 375-5705  
Email: [jbonow@afmc.org](mailto:jbonow@afmc.org)  
Web: [www.afmc.org](http://www.afmc.org)

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.