

The Fundamentals of MERGE

MERGE is a very effective and powerful tool to use when combining data sets in the DATA step. Even though MERGE is powerful, it is not complicated. There are two important rules to remember when merging. Once these rules are understood, deciding how and when to use MERGE will be a much simpler process.

The first rule to remember when merging data sets is that the MERGE statement has an implied RETAIN statement associated with it. This means when there is a match on a value of the BY variable(s), all values in the program data vector are retained until they are overwritten by a new value. Once all the observations for a particular BY-group have been processed, the variable values in the PDV are re-initialized back to missing.

The second rule is that SAS will only read an observation once. This is an important rule to remember. For example, someone is doing a one-to-many merge and wants to update a common variable value that is not a BY variable. One data set contains duplicate observations for the BY variable, the other data set contains unique observations for the BY variable. Due to the rule above, the common variable will contain the value from the unique data set on the first observation, but the subsequent observations for the same BY-group will have the original values from the duplicate data set. The unique record is not read again when the rest of the duplicate records are read.

The two rules above will be illustrated in some of the merge examples below.

***** ONE-TO-ONE MERGE *****

The first example illustrates a one-to-one MERGE. Neither of the 2 rules above pertain to this example, but a one-to-one MERGE is the simplest type of MERGE and is a good starting point in understanding how MERGE works.

To do a one-to-one merge, use a MERGE statement without a BY statement. SAS combines the first observation from all data sets listed on the MERGE statement into the first observation of the new data set, the second observation from all data sets into the second observation of the new data set, and so on.

File 1:

ID	NAME
111	Tom Smith
112	Jenny Jones
113	Mary Brown

File 2:

ID	AGE
111	31
112	25
113	49

Let's step through the MERGE data step as if we were SAS executing the code:

```
data new;
  merge file1 file2;
run;
```

Read in the first observation from FILE1, filling in values of the variables:

ID	NAME	AGE	_ERROR_	_N_
111	Tom Smith	.	0	1

Read in the first observation from FILE2. The common variable value of ID from FILE1 is overwritten by the value in FILE2. Output the observation.

ID	NAME	AGE	_ERROR_	_N_
111	Tom Smith	31	0	1

→ output

Since all the variables in the PDV were already defined and not created in the DATA step, the values are not re-initialized to missing when the DATA step iterates. Read in the second observation from FILE1.

ID	NAME	AGE	_ERROR_	_N_
112	Jenny Jones	31	0	2

Read in the second observation from FILE2. The common variable value of ID from FILE1 is overwritten by the value in FILE2, and the previous value of AGE is overwritten by the new value. Output the observation.

ID	NAME	AGE	_ERROR_	_N_
112	Jenny Jones	25	0	2

→ output

Read in the last observation from FILE1.

ID	NAME	AGE	_ERROR_	_N_
113	Mary Brown	25	0	3

Read in the last observation from FILE2 and output the observation.

ID	NAME	AGE	_ERROR_	_N_
113	Mary Brown	49	0	3

→ output

There are no more observations to be read, so the final data set looks like:

ID	NAME	AGE
111	Tom Smith	31
112	Jenny Jones	25
113	Mary Brown	49

***** MATCH MERGE EXAMPLES *****

In a match-merge, observations from two or more data sets are combined into a single observation in a new data set, according to the values of a common variable. You must use a *MERGE* statement with a *BY* statement to perform a match-merge, and all data sets being merged must either be in sorted order or have an index.

The next example illustrates how SAS performs a one-to-many merge:

File1:

LNAME	COURSEID
garner	eng600
garner	eng700
garner	eng800

File2:

LNAME	COURSEID	CLASSNO
garner	eng400	96511

LNAME is the BY variable, COURSEID is common to both data sets, and CLASSNO is unique to FILE2. There are 3 observations for the BY-group (LNAME=garner) in FILE 1, and one observation for the BY-group in FILE 2.

Let's step through the MERGE data step as if we were SAS executing the code:

```
data final;
  merge file1 file2 ;
  by lname;
run;
```

Read in the first observation from FILE1, filling in values of the variables:

LNAME	COURSEID	CLASSNO	_ERROR_	_N_
garner	eng600	.	0	1

Read in the first observation of the same BY-group from FILE2. The value of COURSEID from FILE2 will overwrite the value of COURSEID from FILE1. Output the observation.

LNAME	COURSEID	CLASSNO	_ERROR_	_N_
garner	eng400	96511	0	1

→ output

At this point, SAS does a 'look-ahead' to see if there are any more observations in either data set for the current BY-group, and sees that there are more in FILE 1.

Read in the second observation of same BY-group from FILE 1. Remember there are no other observations coming from FILE 2. The current value of COURSEID is overwritten by the new value from FILE1. Since CLASSNO is unique and values in the PDV are not re-initialized to missing until the BY group changes, the value of CLASSNO is retained down the BY group. Output the observation.

LNAME	COURSEID	CLASSNO	_ERROR_	_N_
garner	eng700	96511	0	2

→ output

Read in the last observation of the same BY-group from FILE 1. Once again, there are no other observations coming from FILE 2. The current value of COURSEID is

overwritten by the new values from FILE1, and the value of CLASSNO is retained down the BY group. Output the observation.

LNAME	COURSEID	CLASSNO	_ERROR_	_N_
garner	eng800	96511	0	3

→ output

There are no more observations to be read, so the final data set looks like:

LNAME	COURSEID	CLASSNO
garner	eng400	96511
garner	eng700	96511
garner	eng800	96511

The last example illustrates what happens when merging data sets that contain duplicate observations for a BY-group in both data sets. This is referred to as a many-to-many merge. The MERGE statement combines observations within BY-groups, but the end product is not every combination of observations as you might think. SAS does a **one-to-one merge** while there are duplicate observations in more than one data set for a particular BY group. If a BY group is exhausted in one of the data sets, the values from the last observation are then retained and merged as a one-to-many merge until all observations in the BY group have been read.

FILE 1:

ID	FRUIT
1	apple
2	banana
2	lime
2	mango

FILE 2:

ID	COLOR
1	blue
1	green
2	purple
2	yellow

Let's step through the MERGE data step as if we were SAS executing the code:

```

data c;
  merge file1 file2;
  by id;
run;

```

Using the example above, MERGE combines observations in the following manner:

Read in the first observation from FILE1, filling in values of the variables:

ID	FRUIT	COLOR	_ERROR_	_N_
1	apple		0	1

Read in the first observation of the same BY-group from FILE2. Output the observation.

ID	FRUIT	COLOR	_ERROR_	_N_
1	apple	blue	0	1

→ output

At this point, SAS does a 'look-ahead' to see if there are any more observations in either data set for the same BY-group. Since there are no more in FILE1, but there is one more in FILE2, the unique variable values from FILE1 are retained and merged with the second observation in FILE2. The new value of COLOR from FILE2 will overwrite the previous value.

ID	FRUIT	COLOR	_ERROR_	_N_
1	apple	green	0	2

→ output

Since there are no more observations for the BY-group (ID=1), SAS resets all values in the PDV to missing and starts with the second BY-group. Read in the first observation of the second BY-group (ID=2) from FILE1.

ID	FRUIT	COLOR	_ERROR_	_N_
2	banana		0	3

Read in the first observation of the second BY-group from FILE2. Output the observation.

ID	FRUIT	COLOR	_ERROR_	_N_
2	banana	purple	0	3

→ output

When SAS does the look-ahead this time, it sees there are more observations for the same BY-group in BOTH data sets, and reads in the new observations. It does NOT combine each observation from FILE1 with every observation in FILE2 for a particular BY-group (cartesian product). For an example of performing a cartesian product, please see the references at the end of this document.

Read in the next observation from FILE1. The previous value of FRUIT is overwritten by the new value.

ID	FRUIT	COLOR	_ERROR_	_N_
2	lime	purple	0	4

Read in the next observation from FILE2. The previous value of COLOR is overwritten by the new value. Output the new observation.

ID	FRUIT	COLOR	_ERROR_	_N_
2	lime	yellow	0	4

→ output

Now when SAS does the look-ahead, it sees there are no more observations in FILE2 for the BY-group (ID=2), but there is one more observation in FILE1. This time, SAS retains the values from FILE2, merges it with the fourth observation in FILE1, and outputs the observation.

ID	FRUIT	COLOR	_ERROR_	_N_
2	mango	yellow	0	5

→ output

There are no more observations to be read, so the final data set looks like:

ID	FRUIT	COLOR
1	apple	blue
1	apple	green
2	banana	purple
2	lime	yellow
2	mango	yellow

For further information regarding MERGE, please see the following references:

- TS-DOC 644: Everything You Wanted to Know About MERGE But Were Afraid to Ask.
<http://support.sas.com/techsup/technote/ts644.html>
- SAS Language Reference: Concepts, Version 8, Chapter 23: BY-Group Processing in The DATA Step.
- SAS Language Reference: Concepts, Version 8, Chapter 24: Reading, Combining, and Modifying SAS Data Sets.

For an example illustrating how to perform a cartesian product, please look at example 4.4 on the following link:

- <http://ftp.sas.com/samples/A55219>