Tips for Using a Data Warehouse

Curtis Alan Smith

In our last episode we explored how to conditionally execute code based on the number of observations in a SAS data set. This time we will explore how to document our SAS data set names and variables as we create them in our application job stream.

SAS© makes documenting our data very simple, whether we have one SAS data set, an entire data warehouse, or a bunch of loose SAS data sets in our application job stream. We can use the CONTENTS procedure to generate a detailed report of our SAS data libraries, SAS data sets, and SAS data views. But, maybe even more importantly, the label names can easily be used in our output and what we see when we look at our data interactively. We can use the data set LABEL option and the variable LABEL statement to better document our data. Documenting the data about our data, called "metadata", is critical to providing better assurance that our data is used properly. But, how and when do we do all of this documenting?

To Label, or Not to Label, that is the Question

There is an old saying in the information technology world that documentation happens last, if at all. However, we need to be diligent and build documentation into our data warehouses and data sets. Documenting our SAS data libraries, SAS data sets, and SAS data views is so very important. Without some type of documentation, our users will not know which SAS data libraries are which, or what is in them; nor will they know one SAS data set from another. They wouldn't even know a SAS data set from a SAS data view. Without documentation, we programmers, developers, and data warehouse architects will have a difficult time, too. SAS makes documentation very simple. We can use the data set LABEL option to attach a descriptive label to SAS data files. We can also use the variable LABEL statement to attach a descriptive label to each variable in our data sets That's right, there is a means to attach a descriptive label to both a data set and the variables within a data set. If we use them both on all the SAS data sets and SAS data views in our data warehouse and on each variable within each SAS data set and SAS data view, we will be able to produce better documentation.

As we create a new SAS data set or variable, we may think that the meaning or contents of some SAS data sets names and variable names are self-explanatory. Surely, everyone will know what is in the variable named 'amount', right? Well, maybe it represents a year-to-date amount, or a summarized amount, or an overtime amount. Because we can place more information in the label than we can in the data set or variable name, labels become very useful.

Labeling Made Easy

Adding descriptive labels is as easy as coding a SAS program. SAS provides the tools in the form of the data set LABEL= option and the variable LABEL statement. We will look at each.

DATA SET LABEL= OPTION

We can add a label to a SAS data set or data view when creating the data set or data view using the DATA step. We can also create or modify an existing label using the DATASETS© procedure. A label assigned to a file remains associated with that file when we update a data set in place, such as when we use the APPEND procedure or the MODIFY statement. However, a label will not be associated with a data set or data view that we create from a file that contains a label. For example, if we create a label on data set A and then create data set B using a DATA step with data set A in the SET statement, the label will not be added to data set B. This is not a problem, of course, because we can simply add a label to data set B as we create it.

So, how do we create a label on a SAS data set or data view? Quite simply, using the data set LABEL= option. The syntax is, as we would guess:

```
LABEL=’ ’
```

We can use the LABEL= option on both input and output data sets. When we use the LABEL= option on input data sets it assigns a label to the file for the duration of that DATA step or procedure. Creating a label on the input data set doesn’t do much good for documenting our data warehouse, because it will not be permanent.

We can use the data set LABEL= option with the DATA step as follows:

```
Data mylib.myfile(label=’General Ledger YTD 2004');
     set mylib.yourfile;
run;
```

VARIABLE LABEL STATEMENT

We can add a label to a SAS data file variable when creating a data set or data view using the DATA step. We can also create or modify an existing label using PROC DATASETS. When we specified a label for a variable in an output data set, the label becomes a permanent part of that file and can be printed using various procedures (PROC REPORT is one of my personal favorites). A label assigned to a variable remains associated with that variable when we update a data set in place, such as when we use the APPEND procedure or the MODIFY statement. It will also be associated with the variables of a new data file we create from the file containing the labels. For example, if we create labels on the variables in data set A and then create data set B using a DATA step with data set A in the SET statement, the variable labels will be added to data set B.

So, how do we create a label on variables within SAS data sets and data views? Quite simply, using the LABEL statement. The syntax is, as we would guess:

```
LABEL variable=<'label'>
```

We can assign a label to any number of variables in a single LABEL statement. Using a

LABEL statement in a DATA step or PROC DATASETS permanently associates labels with variables by affecting the descriptor information of the SAS data set or data view that contains the variables.

We can use the data set LABEL statement in a DATA step as follows:

```
Data mylib.myfile;
   set mylib.yourfile;
   Label
      Bu      =’Business Unit’
      Pool    =’Overhead Pool’
      Jv      =’Journal Voucher’
      Cdate   =’Business Cycle Date’
      Tdate   =’Transaction Date’
      Account =’Ledger Account’
      Hours   =’YTD Hours’
      Amount  =’YTD Amount’
   ;
run;
```

However, adding data set labels and variable labels may not be possible with some SAS procedures. Fortunately, SAS provides us with the DATASETS procedure that includes many wonderful features, including the LABEL option and LABEL statement. I have gotten into the habit of adding a DATASETS procedure after many of the procedures that I use to create a SAS data set. Here is an example of adding a label to the SAS data set and to each of the variables using PROC DATASETS.

```
Proc datasets library=mylib nolist;
   modify myfile(label=’General Ledger YTD 2004');
   label
      Bu      =’Business Unit’
      Pool    =’Overhead Pool’
      Jv      =’Journal Voucher’
      Cdate   =’Business Cycle Date’
      Tdate   =’Transaction Date’
      Account =’Ledger Account’
      Hours   =’YTD Hours’
      Amount  =’YTD Amount’
   ;
run;
quit;
```

Viola!

Next time we will explore...well, I’m not sure what we will explore, but it will be interesting.

Thanks for reading.

Curtis A. Smith
DoD Program Manager
ca.smith86@att.net

(2018) Base SAS® 9.4 Procedures Guide, Seventh Edition: DATASETS Procedure; Cary, NC: SAS Institute, Inc. Retrieved from https://documentation.sas.com/?docsetId=proc&docsetTarget=n1hmips60w5w3yn1hj9klna7aplw.htm&docsetVersion=9.4&locale=en