

Paper 047-2010

So Many Variables; Too Many Labels

Moving Labels from One Variable to Another

John Ladds, Statistics Canada, Ottawa, Ontario, Canada

ABSTRACT

So you have one hundred plus variables to copy and recode and you want their original variable labels to transfer with them. Good luck with that! SAS will let you do all kinds of things with the labels but it will not let you transfer them to a new variable as easily as you might think but with CALL EXECUTE and a few lines of code you can save yourself some time and a whole lot of typing.

INTRODUCTION

You may think that the key to transferring variable labels would be the VLABEL function or the CALL LABEL routine. Well they will only get you half way there. They will retrieve the variable's label alright but there are no tools to assign the old variable's label to the new variable. `label newVar = VLABEL(oldVar);` does not work! And there is no other assign label function or routine to finish the job.

This is where the CALL EXECUTE routine can be used. What CALL EXECUTE does is store up whatever string arguments you give it to be executed the moment that the current data step is finished. You could do the same thing by building your own temporary dataset and then %INCLUDING it as the next step. That is a lot of work. The following three lines (with some controlling IF statements) are all that a required.

```
call execute('data &syslast.; set &syslast.');
```

```
call execute("label R_&Q.=||' '||strip(vlabel(&Q.))||' '');
```

```
call execute ('run;');
```

LET'S GET STARTED

To transfer variable labels from one variable to another is a three stage process. The first stage is to set up the processing at the end of your current data step; the second stage is to harvest your variable labels; and the final stage is to close off the processing at the end of your current data step.

The CALL EXECUTE lines only need to be run once during the data step; at the beginning or end of the process. It is easiest to use the END= option of the SET statement to control the execution of the CALL EXECUTE statements.

```
set work.theData end=eof;
```

The use of **&SYSLAST** makes the code less error prone. With the use of **&SYSLAST**, the name of the dataset is changed automatically so the **CALL EXECUTE** is always referencing the most recently created SAS data file.

Then use the call routine **CALL EXECUTE** to assign the label text to the new variable at the end of the current data step. This statement is within a macro used to copy and recode each variable. (See the following macro ReverseValueOrder in the example code.) The argument passed to the **CALL EXECUTE** is a string built up from four components to form the LABEL statement that will be executed at the close of the current data step. The goal is to create something that looks like this:

```
label R_Q001="Have materials and equipment I need";
```

The string `label R_&Q. =` is concatenated (||) to a single double quote. The reason for the single and double quotes is to resolve macro variables while building the argument string and to ensure that a single quote in the variable's label does not cause problems at execution. The **VLABEL** function provides the variable's label and the **STRIP** function removes any leading or trailing blanks added by the **VLABEL** function. We end the string by closing the double quote and ending the statement with a semicolon (;).

```
if eof then call execute("label R_&Q.=||' '||strip(vlabel(&Q.))||' '');
```

The final order of business is to set up the last commands that the CALL EXECUTE will run at the close of the current data step. This is a RUN statement to end the data step.

```
if eof then call execute ('run;');
```

The following is snippet of the resulting log from the example program:

```
NOTE: There were 10 observations read from the data set WORK.THEDATA.
NOTE: The data set WORK.THEDATA has 10 observations and 21 variables.
NOTE: DATA statement used (Total process time):
      real time           0.01 seconds
      cpu time            0.00 seconds

NOTE: CALL EXECUTE generated line.
1  + data work.thedata           ; set work.thedata           ;
2  + label R_Q001="Have material's and equipment I need";
3  + label R_Q002="Material and tools avail in lang choice";
4  + label R_Q003="I use the language of my choice";
5  + label R_Q004="My jobs fits my interests.";
6  + label R_Q005="I have support at work.";
7  + label R_Q006="I am satisfied, current work arrangement";
8  + label R_Q007="I can claim overtime";
9  + label R_Q008="Overall, I like my job.";
10 + label R_Q009="I get satisfaction from my work.";
11 + label R_Q010="I know how my work contributes";
12 + run;

NOTE: There were 10 observations read from the data set WORK.THEDATA.
NOTE: The data set WORK.THEDATA has 10 observations and 21 variables.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds
```

CONCLUSION

With a little bit of innovative thinking you can do almost anything with SAS. As proven with this paper, even though SAS does not provide a direct function to transfer labels; you can create a simple call execute routine that solves the problem. CALL EXECUTE stores up whatever string arguments you give it to be executed the moment the current data step is finished.

FUNCTIONS, ROUTINES AND SPECIAL VARIABLES USED

VLABEL FUNCTION

This variable information function returns the label that is associated with the specified variable. It can only be used in a DATA step. By default the function returns a variable at a length of 200. It is necessary to clear any extra blank characters from the variable before it can be used in the new label.

The syntax is **VLABEL (variable)** the 'variable' argument specifies a variable reference.

CALL EXECUTE ROUTINE

This macro category routine resolves the string argument and queues them for execution at the end of the current data step. The statement(s) execute after the end of the **DATA** step that contains the **CALL EXECUTE** routine. **CALL EXECUTE** is fully documented in SAS Macro Language: Reference.

The syntax is **CALL EXECUTE (arguments)**; the 'arguments' specifies SAS statements in this example.

STRIP FUNCTION

This function returns a character string with all leading and trailing blanks removed.

The syntax is **STRIP (string)** the 'string' is a character constant, variable, or expression. The STRIP function is useful for concatenation because the concatenation operator does not remove leading or trailing blanks.

&SYSLAST AUTOMATIC MACRO VARIABLE

The **&SYSLAST** macro variable contains the name of the last SAS data file created. **&SYSLAST** can be directly entered into SAS code in place of a data set name because **&SYSLAST** contains the period between the libref and data set name. The name is in the libref.dataset format.

PROGRAM CODE

```

** ~~~~~;
** System:          SAS Global Forum 2010          ;
**                                                         ;
** Source:          Variable Label Transfer Example ;
**                                                         ;
** Purpose:         Demonstrate a technique to transfer the label from ;
**                 one variable to another        ;
**                                                         ;
** Date:            October 15, 2009              ;
**                                                         ;
** Programmer:      John Ladds,                   ;
**                 SAS Technology Centre,         ;
**                 Statistics Canada              ;
**                                                         ;
** ° º ¸ , , , , ° º ° ° ° º ¸ , , , , ° º ° ° ° º ¸ , , , , ° º ° ° ° º ¸ , , , , ° º ° ° ° º ¸ , , , , ° º ° ° ° º ¸ , , , , ° ;

```

```
proc format;
```

```

value OriginalOrderAgree
  1 = 'Strongly agree'
  2 = 'Somewhat agree'
  3 = 'Neither agree nor disagree'
  4 = 'Somewhat disagree'
  5 = 'Strongly disagree'
  6 = 'Not applicable'
  97 = "Don't know"
  99 = 'Not stated';
value ReverseOrderAgree
  5 = 'Strongly agree'
  4 = 'Somewhat agree'
  3 = 'Neither agree nor disagree'
  2 = 'Somewhat disagree'
  1 = 'Strongly disagree'
other='Missing';

```

```
run;
```

```
** Create some test data;
```

```
data work.theData;
```

```

input @01 id      $10.
      @11 Q001    2.
      @13 Q002    2.
      @15 Q003    2.
      @17 Q004    2.
      @19 Q005    2.
      @21 Q006    2.
      @23 Q007    2.
      @25 Q008    2.
      @27 Q009    2.
      @29 Q010    2.;

```

```

label Q001 = "Have material's and equipment I need";
label Q002 = "Material and tools avail in lang choice";
label Q003 = "I use the language of my choice";
label Q004 = "My jobs fits my interests.";
label Q005 = "I have support at work.";
label Q006 = "I am satisfied, current work arrangement";
label Q007 = "I can claim overtime";
label Q008 = "Overall, I like my job.";
label Q009 = "I get satisfaction from my work.";

```

```

label Q010 = "I know how my work contributes";

format Q001-Q010 OriginalOrderAgree.;

datalines4;
1234567890 1 2 3 4 5 6 1 2 3 4;
1234567892 2 3 4 5 1 1 2 3 4 5;
1234567893 3 4 5 1 2 2 2 3 5 1;
1234567894 4 5 1 2 3 3 3 3 1 2;
1234567895 5 1 2 3 4 4 4 4 2 3;
1234567896 1 2 3 4 5 5 5 5 3 4;
1234567897 2 3 4 5 1 1 1 1 4 5;
1234567898 3 4 5 1 2 2 2 2 5 1;
1234567899 4 598 2 3 3 3 3 1 2;
123456781099 199 3 4 4 4 5 2 3;
;;;
run;

** Recode response values so that higher scores are 'Positive' responses;
%macro reverseValueOrder (Q=,Qfmt=);

select(&Q.);
  when(1) R_&Q.=5;
  when(2) R_&Q.=4;
  when(3) R_&Q.=3;
  when(4) R_&Q.=2;
  when(5) R_&Q.=1;
  otherwise R_&Q.=.;
end;

** Transfer the questions label to the new variable;
if eof then call execute("label R_&Q.="||"'"||strip(vlabel(&Q.))||"'"");

** Add the new value labels to the new variable;
format R_&Q. &Qfmt..;

%mend reverseValueOrder;

data work.thedata;
set work.theData end=eof;

** Open the post DATA STEP Processing;
if eof then call execute('data &syslast.; set &syslast.');
```

```

**A. My Job World ;
%reverseValueOrder (Q=Q001, Qfmt=ReverseOrderAgree);
%reverseValueOrder (Q=Q002, Qfmt=ReverseOrderAgree);
%reverseValueOrder (Q=Q003, Qfmt=ReverseOrderAgree);
%reverseValueOrder (Q=Q004, Qfmt=ReverseOrderAgree);
%reverseValueOrder (Q=Q005, Qfmt=ReverseOrderAgree);
%reverseValueOrder (Q=Q006, Qfmt=ReverseOrderAgree);
%reverseValueOrder (Q=Q007, Qfmt=ReverseOrderAgree);
%reverseValueOrder (Q=Q008, Qfmt=ReverseOrderAgree);
%reverseValueOrder (Q=Q009, Qfmt=ReverseOrderAgree);
%reverseValueOrder (Q=Q010, Qfmt=ReverseOrderAgree);
** ... .. ;

** Close the post DATA STEP Processing;
if eof then call execute ('run;');
```

```

run;
```

```
proc contents data=work.theData;  
run;
```

PROGRAM LOG

NOTE: Copyright (c) 2002-2008 by SAS Institute Inc., Cary, NC, USA.

NOTE: SAS (r) Proprietary Software 9.2 (TS2M0)
 Licensed to STATISTICS CANADA, Site 88812367.

NOTE: This session is executing on the XP_PRO platform.

NOTE: SAS initialization used:

```
real time      1.96 seconds
cpu time       1.06 seconds
```

```
1  ****;
2  ** System:      SAS Global Forum 2010      ;
3  **              ;
4  ** Source:      Variable Label Transfer Example ;
5  **              ;
6  ** Purpose:     Demonstrate a technique to transfer the label from ;
7  **              one variable to another      ;
8  **              ;
9  ** Date:        October 15, 2009          ;
10 **              ;
11 ** Programmer:  John Ladds,                ;
12 **              SAS Technology Centre,      ;
13 **              Statistics Canada           ;
14 **              ;
15 ****;
16
17 proc format;
18   value OriginalOrderAgree
19     1 = 'Strongly agree'
20     2 = 'Somewhat agree'
21     3 = 'Neither agree nor disagree'
22     4 = 'Somewhat disagree'
23     5 = 'Strongly disagree'
24     6 = 'Not applicable'
25     97 = "Don't know"
26     99 = 'Not stated';
NOTE: Format ORIGINALORDERAGREE has been output.
27   value ReverseOrderAgree
28     5 = 'Strongly agree'
29     4 = 'Somewhat agree'
30     3 = 'Neither agree nor disagree'
31     2 = 'Somewhat disagree'
32     1 = 'Strongly disagree'
33     other='Missing';
NOTE: Format REVERSEORDERAGREE has been output.
34 run;
```

NOTE: PROCEDURE FORMAT used (Total process time):

```
real time      0.09 seconds
cpu time       0.00 seconds
```

```
35
36 ** Create some test data;
37 data work.theData;
38   input @01 id $10.
39     @11 Q001 2.
40     @13 Q002 2.
41     @15 Q003 2.
42     @17 Q004 2.
43     @19 Q005 2.
44     @21 Q006 2.
45     @23 Q007 2.
46     @25 Q008 2.
47     @27 Q009 2.
48     @29 Q010 2.;
49
50   label Q001 = "Have material's and equipment I need";
51   label Q002 = "Material and tools avail in lang choice";
52   label Q003 = "I use the language of my choice";
53   label Q004 = "My jobs fits my interests.";
```



```
3 + label R_Q002="Material and tools avail in lang choice";
4 + label R_Q003="I use the language of my choice";
5 + label R_Q004="My jobs fits my interests.";
6 + label R_Q005="I have support at work.";
7 + label R_Q006="I am satisfied, current work arrangement";
8 + label R_Q007="I can claim overtime";
9 + label R_Q008="Overall, I like my job.";
10 + label R_Q009="I get satisfaction from my work.";
11 + label R_Q010="I know how my work contributes";
12 + run;
```

NOTE: There were 10 observations read from the data set WORK.THEDATA.

NOTE: The data set WORK.THEDATA has 10 observations and 21 variables.

NOTE: DATA statement used (Total process time):

real time	0.01 seconds
cpu time	0.01 seconds

120

```
121 proc contents data=work.theData;
```

```
122 run;
```

NOTE: PROCEDURE CONTENTS used (Total process time):

real time	0.03 seconds
cpu time	0.01 seconds

PROGRAM OUTPUT

The SAS System 10:43 Thursday, October 15, 2009 1

The CONTENTS Procedure

Data Set Name	WORK.THEDATA	Observations	10
Member Type	DATA	Variables	21
Engine	V9	Indexes	0
Created	Thu, Oct 15, 2009 10:57:33 o'clock AM	Observation Length	176
Last Modified	Thu, Oct 15, 2009 10:57:33 o'clock AM	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS_32		
Encoding	wlatin1 Western (Windows)		

Engine/Host Dependent Information

Data Set Page Size	16384
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	92
Obs in First Data Page	10
Number of Data Set Repairs	0
Filename	C:\DOCUME~1\laddsj\LOCALS~1\Temp\SAS Temporary Files_TD5172\thedata.sas7bdat
Release Created	9.0202M0
Host Created	XP_PRO

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Label
2	Q001	Num	8	ORIGINALORDERAGREE.	Have material's and equipment I need
3	Q002	Num	8	ORIGINALORDERAGREE.	Material and tools avail in lang choice
4	Q003	Num	8	ORIGINALORDERAGREE.	I use the language of my choice
5	Q004	Num	8	ORIGINALORDERAGREE.	My jobs fits my interests.
6	Q005	Num	8	ORIGINALORDERAGREE.	I have support at work.
7	Q006	Num	8	ORIGINALORDERAGREE.	I am satisfied, current work arrangement
8	Q007	Num	8	ORIGINALORDERAGREE.	I can claim overtime
9	Q008	Num	8	ORIGINALORDERAGREE.	Overall, I like my job.
10	Q009	Num	8	ORIGINALORDERAGREE.	I get satisfaction from my work.
11	Q010	Num	8	ORIGINALORDERAGREE.	I know how my work contributes
12	R_Q001	Num	8	REVERSEORDERAGREE.	Have material's and equipment I need
13	R_Q002	Num	8	REVERSEORDERAGREE.	Material and tools avail in lang choice
14	R_Q003	Num	8	REVERSEORDERAGREE.	I use the language of my choice
15	R_Q004	Num	8	REVERSEORDERAGREE.	My jobs fits my interests.
16	R_Q005	Num	8	REVERSEORDERAGREE.	I have support at work.
17	R_Q006	Num	8	REVERSEORDERAGREE.	I am satisfied, current work arrangement
18	R_Q007	Num	8	REVERSEORDERAGREE.	I can claim overtime
19	R_Q008	Num	8	REVERSEORDERAGREE.	Overall, I like my job.
20	R_Q009	Num	8	REVERSEORDERAGREE.	I get satisfaction from my work.
21	R_Q010	Num	8	REVERSEORDERAGREE.	I know how my work contributes
1	id	Char	10		

REFERENCES

SAS OnlineDoc 9.2

ACKNOWLEDGMENTS

I would like to thank Tom Kari of Statistics Canada's SSD SAS Technology Centre for pointing me at the CALL EXECUTE routine in the first place.

RECOMMENDED READING

There are a few threads on SAS-L related to the issue of transferring variable label worth reading.

<http://www.listserv.uga.edu/archives/sas-l.html>

CONTACT INFORMATION

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

Name: John Ladds
Enterprise: Statistics Canada – SDD SAS Technology Centre
Address: 14-R-90 R.H. Coats Building,
100 Tunney's Pasture Driveway
Ottawa, Ontario, Canada
K1A 0T6
Phone: 613 951-1767
E-mail: John.Ladds@statcan.gc.ca
Web: www.statcan.ca

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.