How to Keep Multiple Formats in One Variable after Transpose
Mindy Wang

Abstract
In clinical trials and many other research fields, proc transpose are used very often. When many variables with their individual formats are transposed into one variable, we lose the formats. We can do a series of if then statements to put the formats back in. However, when the variables involved are too many, the above method can be very tedious. This paper illustrates how to extract formats from dictionary.columns or sashelp.vcolumn, and then use PUTN function to assign the formats at run time and make the task much easier. In addition, it is much easier to apply the same method to other projects without a lot of hard coding in the SAS program. Efficiency is largely increased with this method

Introduction
Problems arise when transposing the different observations with different formats into one variable. It is impossible to have different formats for different observations for the same variable. Therefore the original formats do not show up at all after transpose. I have the following formats before transpose.

```
proc format;
  value sexcode
    1 = 'Male'
    2 = 'Female';
  value racecode
    1 = 'White'
    2 = 'Black'
    3 = 'Hispanic'
    4 = 'Asian'
    5 = 'Other'
    6 = 'Multi-racial';
  .......
run;

data demo2;
set desk.demo;
  format
datestmp datetime17.
birthdt mmdyy10.
sexcd sexcode.
racecd racecode.

run;
```

Following is the snapshot before the transpose. Please note that the sexcd and racecd variables display text formats.
Following is the snapshot after the transpose. Please note the sexcd and racecd now are numeric, instead of text.
**If Then and Else Method**

There is some work to do to fix the problem. One way of doing it is to have a series of if then statements to accommodate value labels and hard code the value labels into a new variable. We can do it individually to each variable with format. Following is the codes that I used with this method.

```plaintext
if upcase(compress(_name_))="DATESTMP" then do;
   n_value=put(Col1, datetime17.);
end;
else if upcase(compress(_name_))="BIRTHDT" then do;
   n_value=put(Col1, mmddyy10.);
end;
else if upcase(compress(_name_))="SEXCD" then do;
   if Col1=1 then n_value= 'Male  ';
   else if Col1=2 then n_value= 'Female' ;
end;
else if upcase(compress(_name_))="RACECD" then do;
   if Col1=1 then n_value= 'White      ' ;
   else if Col1=2 then n_value= 'Black' ;
   else if Col1=3 then n_value= 'Hipanic' ;
   else if Col1=4 then n_value= 'Asian' ;
   else if Col1=5 then n_value= 'Other' ;
   else if Col1=6 then n_value= 'Multi-racial' ;
end;
else n_value=Col1;
```

Following is the snapshot after I used if then and else method.
While the method is working well with only a few variables with formats to be transposed, it becomes very tedious when the observations are in the hundreds or thousands.

**Extracting formats from Dictionary.Columns or Sashelp.Vcolumn and then applying PUTN function**

An easier way of doing it is to pull out the format information from dictionary.columns using sql, and then apply putn function to assign the individual format for each observation at run time. If you are not a sql person, you can do the same by pulling out format information from Sashelp.vcolumn. Following is the demonstration using dictionary.columns or sashelp.vcolumn then applying the formats using putn function. This method is much easier to adapt to other situations without a lot of hard coding.

Following is where we can find the Sashelp.Vcolumn file.
You can use the following code to print the variable names and formats from Sashelp.Vcolumn.

```sas
proc print data=sashelp.vcolumn noobs;
where libname = 'WORK' and memname = 'SAMPLE';
```
var name format;
run;

Following is the snapshot of the output window after running the above code.

You can also export the Sashelp.Vcolumn file to excel if you are more familiar with Excel.
Following is the snapshot of the Excel file with all other data from Sashelp.Vcolumn except the sample file (that we are interested) filtered out.
Following is the code using proc sql. Then print the output file to Excel using ods tagsets.excelxp.

```sas
proc transpose data=sample name=name out=t89;
run;

data 289;
set t89;
seq+1;
run;

/*PULL OUT INFORMATION FROM DICTIONARY.COLUMNS*/
proc sql noprint;
create table temp as select name, format from dictionary.columns
   where libname='WORK' and memname="SAMPLE";
quit;

/*PULL OUT INFORMATION FROM SASHELP.VCOLUMN*/
/*IT DOSE THE SAME AS THE ABOVE METHOD*/
/*UDE THIS ONE OR THE ABOVE SQL METHOD*/
data temp2;
set sashelp.vcolumn;
where libname = 'WORK' and memname='SAMPLE';
keep name format;
run;

proc sort data=temp nodupkey;
by name;
run;

proc sort data=289;
by name;
run;

data f89;
length name $32.;
merge 289(in=a) temp;
by name;
if a;
if format gt ' ' then value=putn(Col1,format);
if value=' ' then value=col1;
run;

proc sort data=f89;
by seq;
run;

ods listing close;
ods tagsets.excelxp file="desktop\data_for_visual_check.xls";
ods tagsets.excelxp options(sheet_name="id 89");
proc print data=f89 noobs;
where _label_ ^= ' ';
var _label_/style=[cellwidth=750];
var value /style=[cellwidth=300];
title1;
```
run;
ods tagsets.excelxp close;
ods listing;

Following is the snapshot of the file after applying formats using PUTN function.

Conclusions
The latter method certainly makes your life easier. Not only it has minimized typing when the variables involved are too many, but is also more adaptable to other projects. It is always beneficial to write the programs that are easier to adapt to new situations, even though at first it takes time to develop, in the long run, it really saves time.

Acknowledgement
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.
Many thanks to SAS Technical Support for providing valuable suggestions in the developing of this paper.

Contact Information
Your comments and questions are valued and encouraged. Please contact the author at:
Mindy Wang
14412 Stonebridge View Dr.
North Potomac, MD 20878
Phone: (240) 855-3479
Email: ymindywang@yahoo.com