ABSTRACT
PROC TRANSPOSE is a useful procedure used almost daily in SAS programming. One powerful feature is the variable created by PROC TRANSPOSE to identify the source of the values in each observation in the output data set. But this feature limits the transposition of formatted values, particularly when the variables in the input data set have mixed and/or complicated formats. This paper will demonstrate the use of a macro to create a new format to transpose differently formatted values.

INTRODUCTION
Take a clinical data set with over 200 questioned variables and corresponding (or relevant) responses which includes more than 30 different styled formats. The request is transposing the questions and response data set vertically for each subject.

But PROC TRANSPOSE is limited to transposing formatted values. We considered three possible options:

One option would be to use an array to group similar formatted variables together then use the PUT function to change all the variables to character format. This solution can be quite time consuming and could involve much programming.

A second option would be to use SQL or data steps to create a label table or data set then merge back to the original data set with a long list of client specified formats. Using this method you have to transpose data twice and again can be quite involved from a time and programming point of view.

A third option uses the following macro. The macro uses an output data set from PROC CONTENTS and macro variables to dynamically specify the PUT function with the new variable name and new formatted character variables.

```sas
%macro putfmt (ds=);
  proc contents data=&ds noprint out=fmt(keep=name format format1 label);
  run;

  data fmt;
  length format $15;
  set fmt;
  if format1 ^=0 and format=''
    then format=trim(left(put(format1, 2.)))||'.';
  else if format1 ^=0 then do;
    if format1=200 then format='$200.';
    else format=trim(left(format))||trim(left(put(format1,2.)))||'.';
  end;
  else if format1=0 and format='' then format='';
  else format=trim(left(format))||'.';
  if substr(name, 1,3)='BEF' and format='' then format='$20.';
  if substr(name, 1,3)='BEF';
  run;

  proc sort;
  by name;
  run;

  proc sql noprint;
  select name, format into :vname1-:vname500, :vfmt1-:vfmt500
  from fmt ;
  quit;
  %let nobs=&sqlobs;

  data &ds;
  set &ds;
```


CONCLUSION
There are many ways to list the clinical data for formatted variables. Compared to other methods, using this macro to create a new variable and format requires the data set to be transposed only once. It is a very simple data step and is particularly useful where a data set has many variables and formats.

The author of this paper hopes that other SAS users can benefit from this little tip.

ACKNOWLEDGMENTS
Many thanks to my manager, Danny Cerro, and my co-worker, principle programmer Peter Lin, Gordon Cummingham, for their suggestions and support for this paper.

CONTACT INFORMATION
Shulin Yuan
AstraZeneca
C3C-613
1800 Concord Pike
PO Box 15437
Wilmington DE 19850-5437
Work Phone: (302) 885 3160
Fax: (302) 885 5343
E-mail: shulin.yuan@astrazeneca.com