ABSTRACT
In the CDISC dataset (Domain) LB we have lab test code, lab test name, lab units, etc. A simple macro can be set up to generate a list of macro variables with lab test code as the macro name and the lab test name as the value of each of the macro variables, or lab unit as the value, etc. These macro variables will then be used in the tables, listings or graphs. In this paper, a macro LABTEST will be introduced for this purpose. Three parameters are required in this macro. Some sample data and sample call will be presented here.

INTRODUCTION
In the pharmaceutical setting, lab data analyses are often required for the clinical study report including tables, listings and graphs. Lab test code, lab test name, and lab units are used in the tables, listings or figures. Currently the CDISC datasets (Domain) become a standard practice and accepted by FDA regulatory. This macro will provide a tool to generate a list of macro variables with lab test code as the macro name, and the lab test name as the value of the macro variable, or lab unit as the value, etc.

SAS MACRO CODE
The programming code presented in this paper is based on version 9.1 of SAS. It can also be used on version 8.2 of SAS. For demonstration purposes, lab units are used for the value of the macro variable in this paper. In the first part, PROC SORT with nodupkey will keep only unique lab test code in the output dataset _temp1. Then PROC SQL will generate a list of unique macro variables with lab test code as the macro name using the dataset _temp1. The following %global statement will assign all macro variables in the list as global macro variables for later use after the macro ends.

In the DATA NULL step, the value of lab units will manipulated so that some special characters will be converted into generally accepted format. The several statements CALL SYMPUT are used in the macro to assign the value of lab unit to each of macro variables correspondently. Lastly, a list of macro variables and their values are presented in the log file for reference in a %DO loop with %PUT statement.

Sample Data:
```sas
data lb;
  input SUBJID $ LBTESTCD $ LBSTRESU $;
datalines;
199-001   BCALK   U/L
199-001   BCALT   U/L
199-001   BCAST   U/L
199-033   BCBHCG  NONE
199-001   BCBUN   mmol/L
199-001   BCCL    mmol/L
199-001   BCCO2   mmol/L
199-001   BCCREAT  umol/L
199-001   BCGGTP  U/L
199-001   BCGLUC  mmol/L
199-001   BCK     mmol/L
```

Sample Macro Code:
%macro labtest(indat=, lbtest=, lbparm=);
    proc sort data=&indat out=_temp1(keep=&lbtest &lbparm) nodupkey;
        by &lbtest &lbparm;
    where &lbparm^in('NONE');
    run;

    proc sql noprint;
        select distinct &lbtest into: cdlist separated by ' ' from &indat;
    quit;

    %global &cdlist;
    data _null_; 
        set _temp1 end=eof;
            num=_n_; 
            length stdparm $25;
            stdparm=compbl(&lbparm);
            if substr(stdparm,1,1)='X' then substr(stdparm,1,1)='x';
            stdparm=tranwrd(stdparm,'**','^');
            stdtest=compbl(&lbtest);
            stdtest=tranwrd(stdtest,'%','_');
            stdtest=tranwrd(stdtest,'-','_');
            call symput(compress(stdtest),trim(stdparm));
            call symput('id'||compress(put(num,3.)), compbl(stdtest));
            call symput('u'||compress(put(num,3.)), compbl(stdparm));
        if eof then call symput('totid', compbl(put(num,3.)));
    run;

    %put *****************************************************;
    %put **** Macro variables and value of each macro created ***;
    %do i=1 %to &totid;
        %put &&id&i  &&u&i;
    %end;
    %put *****************************************************;
%mend labtest;

Sample Macro Call:
    %labtest(indat=lb, lbtest=lbtestcd, lbparm=lbstresu)

Sample Log:
**** Macro variables and value of each macro created ***
BCALK  U/L
BCALT  U/L
BCAST  U/L
BCBUN  mmol/L
BCCL   mmol/L
BCCO2  mmol/L
BCCREAT umol/L
BCGGTP U/L
CONCLUSION
The code presented in this paper can help programmers to produce a list of macro variables from lab test code while a list of another parameter such as lab units or lab test name will be the value of each of macro variables. These macro variables can be conveniently used in the tables, listings or graphs in the clinical study report.

REFERENCES

ACKNOWLEDGEMENTS
I would like to thank my colleague John Hotaling for reviewing this paper and giving valuable comments.

CONTACT INFORMATION
Huijuan Xu, MPH
Staff Analyst, Statistical Programming
biogen idec, Inc.
14 Cambridge Center, Cambridge, MA 02142
Tel: (617) 679-2373
Fax: (617) 679-3280
E-mail: huijuan.xu@biogenidec.com

DISCLAIMER
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.