Interactively Make Uniform and Reorganize Variables in a SAS® Library

Jay Garacani, AstraZeneca, Wilmington, DE

ABSTRACT
Datasets in a SAS® library generally have some common variables. For instance a variable SEX in data set DEMOG may have a different name (like GENDER) in data set AE, or a common variable in a different dataset may not have the same length, format, informat or label. With the help of SAS® 9.1 and Microsoft Visual Basic, we can reorganize all variables which do not have the same name, format, informat or length and make them uniform; also we can uppercase or lowercase all variables across the library. All operations are done interactively.

INTRODUCTION
This paper is designed to rename or to modify attributes of variables across a given library, the output may be generated in-place or in a different library. The program takes advantage of hash table using SAS® 9.1 under Microsoft Windows.
The Microsoft Visual Basic application reads contents of a SAS library and outputs a SAS program that represents the changes.

The output program creates the followings:

- Path name out - Path name for the new library that contains modified datasets
- Upper case or Lower case - Converts all variables in either Lower case or Upper case, default is no change
- Exclude or include Dataset(s) – List of all excluded dataset(s) or list of included dataset(s)
- Rename variable(s) – List of all variable(s) to be renamed, Scope is global
- Change variables’ attributes – If any field is missing it will be filled with blank
- Modify variables’ attribute - Modifies at least one field
- Key Variable(s) – Chooses variable’s attribute to be used in all other datasets

The above parameters are entered into the SAS program with help of a Microsoft Visual Basic application and sets of macros process according to their functions.
The output is a library of datasets, where each dataset’s variable has unique attributes across all datasets. The following illustrates the steps to get the output catalog.

MICROSOFT VISUAL BASIC APPLICATION
The application developed with VB6 due to compatibility with all other Microsoft Windows, the following is the screenshot at the initial run:

It has a Text Box named Log Window for displaying the log and 3 Drop-Boxes File, Operations and Set up.
LOG WINDOW
This window displays all activities, at the initial run shows the default path for SAS®.exe file and the library name and the path to be modified.

FILE DROP BOX
It contains Read Datasets, Write Datasets and Exit.
- Read Datasets - Reads the library's content such as Dataset name, Variable names, variable's type, variable's length, label, format and informat.
- Write Datasets – Writes a short SAS® code from Operation Drop-Box
- Exit – Exits the application

OPERATIONS DROP BOX
It contains Brows Dataset, Upper case, Lower case, Exclude Dataset(s), include Dataset(s), Rename variable(s), Change variables' attribute, Modify variables' attribute and Key Variable(s).
- Brows Dataset – Displays valuable's names with one selected attribute (i.e. Length, Label, … )
- Upper case – Upper cases all variable names
- Lower case – Lower cases all variable names
- Exclude Dataset(s) – Excludes dataset(s) from the operations
- include Dataset(s) – includes dataset(s) from the operations
- Rename variable(s) - List of all variable(s) to be renamed, Scope is global
- Change variables' attribute - If any field is missing, it will be filled with blank
- Modify variables' attribute – Modifies at least one field
- Key Variable(s) - Chooses variable's attribute to be used in all other datasets

SET UP DROP BOX
It contains SAS® bin Path and Name, SAS® Library Path and Log View (Visible)
- SAS® (.exe) Path and Name- Reassigns the Path and the name of the SAS®.exe
- SAS® Library Path-
- Log Window (Visible toggle)- Disable or enable the Log Window

PATH NAME
Macro %PATH_NAME is for setting up IN_CAT and OUT_CAT lib names, which are input and output datasets catalogs. IN and OUT are the macro parameters, OUT default value is equal to IN parameters if it is missing. PATH_IN and PATH_OUT declared %global for further use.

```sas
%path(in=/users/p1/data/final,out=/users/p1/data/tmp );
%path(in=/users/p1/data/final); 
```
First %PATH assigns IN_CAT to /users/p1/data/final and OUT_CAT to /users/p1/data/tmp, Second %PATH assigns both IN_CAT and OUT_CAT to /users/p1/data/final. We have to make sure that OUT_CAT has write access.

UPPER CASE OR LOWER CASE
We can convert the variable names to upper case or lower case in all datasets. %VAR_CASE macro with parameter CASE assigns a desire value to the global macro variable V_CASE. V_CASE may have U for uppercase or L for lowercase. %var_case(vc=L) will convert all variables to lowercase, if we do not call this macro no change will be made to the variables.

INCLUSION OR EXCLUSION OF DATASETS
Macro %INC_EXCL. is for either including or excluding datasets with parameters INC and EXCL. Only one parameter can be presented (i.e. if we want to exclude two datasets, we assign them to EXCL, or if we want to select only 4 datasets we assign them to INC and then will call the macro). For this case program will accept only single quote (not double quote), default is including all datasets.

```sas
%inc_excl(excl='PLASMA2' 'PLASMA_FINAL' );
%inc_excl(inc= 'DEMOG' 'PLASMA'); 
```
First %INC_EXCL excludes PLASMA2 and PLASMA_FINAL datasets from the library and second %INC_EXCL takes only DEMOG and PLASMA datasets.
RENAMENOREN is for renaming the variable across all datasets; it has three parameters NAME, TYPE and NEW_NAME. TYPE can have value of 1 for numeric and 2 for character, default value for TYPE is numeric. If we have four variables to rename, we have to call %REN for each one with the proper parameters. %REN increments macro variable R_CNT with initial value of 0 for every call and generates three new macro variables R_N for the NAME, R_T for the TYPE and R_NN for the NEW_NAME with concatenation of &R_CNT at the end of them.

```
%ren(name=AGE,new_name=YEAR);
%ren(name=SEX,new_name=GENDER,type=2);
```

First %REN renames variable AGE to YEAR and second %REN renames SEX to GENDER. Default for TYPE is 1 (numeric) and we don’t have to include it in %REN.

MODIFY OR CHANGE ATTRIBUTES
Macro %M_ATTR is for modifying variable’s attributes such as length, label, format and informat; it has 6 parameters NAME, TYPE, LEN, LBL, FMT and INF. NAME and TYPE identify the variable uniqueness and LEN is for length, LBL is for label, FMT is for format and INF is for informat of the variable; there is no default value for any LEN, LBL, FMT and INF. The same as %REN it generates a counter M_CNT with the initial value of 0 which will increment with each call to the %M_ATTR and 6 new macros corresponding to each macro variable namely M_N&M_CNT for the name, M_T&M_CNT for the type, M_LEN&M_CNT for the length M_LBL&M_CNT for the label, M_FMT&M_CNT for the format and M_INF&M_CNT for informat of the variable.

Macro %CATTR is for changing variable’s attributes and has the same parameters as %M_ATTR and a default value will be assign to a missing parameter; the default value for LEN is 8 and default for LBL, FMT and INF are %str(). The macro variables that generated by %CATTR are the same as %M_TR except they are started with letter C instead of M.

```
%m_atr(name=BIRTHDTN,fmt=YYMMDD10.);
%c_atr(name=BIRTHDTN,fmt=YYMMDD10.);
```

%M_ATTR modifies BIRTHDTN format but %C_ATTR modifies BIRTHDTN format and also, clears LABEL, FORMAT and INFORMAT and assigns default length of 8 to the LENGTH.

BASE VARIABLES
%BVARS macro is for selecting a variable within a dataset as a base, that means all variables with same name and type in other datasets will have the same attributes as the base variable. It has 3 macro variables DSN, NAME and TYPE, global macro variable BV_CNT with initial value of 0 will increment with each call to %BVARS and generates 3 macro variables BV_DSN for dataset name, BV_N for the name and BV_T for the type.

```
%bvar(dsn=DEMOG,name=BIRTHDTN);
```

%BVARS will copy LENGTH, LABEL, FORMAT and INFORMAT of BIRTHDTN in DEMOG into LENGTH, LABEL, FORMAT and INFORMAT of BIRTHDTN in all other datasets.

DATASETS CONTENTS
%CONTENTS loads all informations about datasets in the library to MS_DS1 dataset, then we use EXLD_DS and &INC_DS to exclude or include datasets.

```
proc contents data=in_cat._all_ noprint
out=ms_ds1(
 %if "&exld_ds" ne "" %then %do; where=(not (dsn in (&exld_ds))) %end;
 %if "&inc_ds" ne "" %then %do; where=( dsn in (&inc_ds)) %end;
 keep=  MEMNAME    NAME   TYPE  LENGTH LABEL FORMAT  INFORMAT VARNUM
rename=( NAME=varn MEMNAME=dsn) );run;
```
CHANGING VARIABLES NAME

In order to change the variable name, we are creating arrays of dataset name, variable name and variable type.

```%
if &r_cnt ne 0 %then
  %do;array ren_n{&r_cnt} $32 _temporary_
    (%do j=1 %to &r_cnt;"&&r_n&j" %end;);
  array ren_t{&r_cnt} _temporary_
    (%do j=1 %to &r_cnt; &&r_t&j %end;);
  array ren_nn{&r_cnt} $32 _temporary_
    (%do j=1 %to &r_cnt;"&&r_nn&j" %end;);
  %end;
else
  %do;array ren_n{1} $1 _temporary_  (' ' );
  array ren_t{1} _temporary_     (0);
  array ren_nn{1} $1 _temporary_  (' ' );
  %end;

dsn_k=dsn;    ** Assign hash-key variable dataset-name ;
type_k=TYPE;  ** and type                              

&R_CNT is the size of variables, since we have to have an array, if &R_CNT is equal to 0, then just one 
dimension   array is created. This pattern of creation of arrays will be repeated for all other cases. We 
create a set of flags for NAME, LENGTH, LABEL, FORMAT and INFORMAT; they will be set to 1 when they 
are modified. For HASH table we assign a copy of dataset name, variable name and the type as hash key 
variables.

f_name=0;  ** Initilize flags to 0 ;
f_len=0;
f_lbl=0;
f_fmt=0;
f_inf=0;
do i=1 to &r_cnt;
  if upcase(varn)=upcase(ren_n{i}) and TYPE=ren_t{i}
    then
      do;varn_k=ren_nn{i};
      f_name=1;
      i=&r_cnt;  ** Forced to exit the loop ;
    end;
  else varn_k=varn;
end;

select ("&v_case");
  when ('U')
    do;  varn_fin = upcase(varn_k);
      if varn_k ne varn_fin then f_name=1;
    end;
  when ('L')
    do;  varn_fin = lowcase(varn_k);
      if varn_k ne varn_fin then f_name=1;
    end;
  otherwise
    varn_fin = upcase(varn_k);
end;

VARN_FIN is the final variable.

HASH TABLE

Hash table has 3 key variables dataset name, variable name and type and has 13 data variables including 
dataset name, variable name, type, length, label, format, informat and flags.

** Hash table declaration

;
length dsn_k varn_k dsn varn varn_fin $32
FORMAT INFORMAT $32 LABEL $256;

declare hash h( dataset: 'hash1');
h.definekey('dsn_k','varn_k','type_k');
h.definedata( 'dsn', 'varn', 'varn_fin', 'TYPE', 'LENGTH', 'LABEL', 'FORMAT', 'INFORMAT',
            'f_name', 'f_len', 'f_lbl', 'f_fmt', 'f_inf' );

h.definedone();
call missing(type_k, dsn, varn, varn_fin, TYPE,
            LENGTH, LABEL, FORMAT, INFORMAT, f_name,
            f_len, f_lbl, f_fmt, f_inf);

end;

First we construct arrays of variables for %M_ATR, %C_ATR, and %BVAR, then with hash function, check and modify the array for each case and at the end, create set of macros from elements of array which are modified.

CALL EXECUTE
%EMULATE generates a sequence of SAS® data steps and then run them with call execute function, for each dataset, one data step will be generated:
- Rename - old name=new name
- LENGTH - RETAIN (list of all dataset variable) followed by LENGTH statement before SET
- Label - LABEL var-name='New label'
- Format - FORMAT varname FMT.
- INFORMAT - INFORMAT varname FMT.

If we have any illegal SAS® syntax, call execute will generate an error for each one, such as illegal format or rename when the new variable exist.

CONCLUSION
This program interactively make uniform and reorganize variables in a SAS® Library. If we change a variable’s label in one data set we don’t have to go to all data sets to replace it. Uppercase, lowercase, rename change and modify variable's attributes, will increase productivity and homogenize all datasets.

REFERENCES

Jay Garacani, Rename and Modify Attributes of Variables Across All SAS® Datasets in the Data Build Catalog, Proceedings of NESUG 2007 Conference,

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

Jay Garacani
AstraZeneca
1800 Concord Pike
Wilmington, DE 19850
(302) 885-1024
E-mail: Jay.Garacani@AstraZeneca.com

SAS® is a registered trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.