Rename and Modify Attributes of Variables Across All SAS® Datasets in the Data Build Catalog

Jay Garacani, AstraZeneca, Wilmington, DE

ABSTRACT
This paper illustrates step by step how to rename and modify attributes of variables using SAS® 9.1. The main objectives are: 1) renaming variables; 2) modifying the attribute of variables (i.e. length, format and label); 3) assuming attribute of a variable in specific dataset and applying that to all other datasets. The result is new datasets with unique variable attribute for the same variable across all SAS datasets, all variables have the same value as before except when the length of them are modified (variables are truncated in length reduction).

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 UNIX OS. The inputs for program are:
• Path name in
• Path name out - Default is Path name in, if it is missing
• Upper case or Lower case - Converts all variable 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 include dataset(s)
• Rename variable(s) – List of all variable(s) to be renamed, Scope is global
• Change variables’ attribute – If any field is missing it fills with blank
• Modify variables’ attribute - Modify at least one field
• Key Variable(s) – Choose attribute of variable for all variable in all other datasets

Above parameters are entered in the program with sets of macros, and process according to their functions. The output is a catalog of datasets, where each dataset’s variable has unique attributes across all datasets. The following illustrates the steps to get the output catalog.

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

```
%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 coverts 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.

```
%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.

**RENAME**

Macro \%REN 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.

```plaintext
%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\_ATR 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 \%C\_ATR is for changing variable’s attributes and has the same parameters as \%M\_ATR 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 \%C\_ATR are the same as \%M\_TR except they are started with letter C instead of M.

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

\%M\_ATR modifies BIRTHDTN format but \%C\_ATR 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 \%BVAR and generates 3 macro variables BV\_DSN for dataset name, BV\_N for the name and BV\_T for the type.

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

\%BVAR 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.

```plaintext
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.

``` Sas
%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.

``` Sas
f_name=0;  ** Initialize flags to 0 ;
f_len=0;
fLbl=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;

```

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.

``` Sas
** Hash table declation ;
```

``` Sas
Posters
NESUG 2007
```
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 will help to unify all data sets in a given catalog. 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.

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.