```
%macro mu_var_attributes
(datasets=
, variables=
, debug=
, help=)
/store source des='V1.0.0.5' ;
%*************************************************************************
FILENAME: MU_VAR_ATTRIBUTES.SAS
DEVELOPER: (b) (6)
PLATFORM: SAS 9.1.2, 9.2, 9.3 on PC SAS
DATE: 06JUN2012
PURPOSE: This macro provieds the attributes of supplied variables. Below is the
list of attributes which are
                provided.
                1.     Type of a variable. Macro Variable :-
library_dsname_variablename_vartyp (In the case of WORK library :-
dsname_variablename_vartyp)
                2. Format of a variable. Macro Variable :-
library_dsname_variablename_fmt (In the case of WORK library :-
dsname_variablename_fmt)
                3. Label of a variable. Macro Variable :-
library_dsname_variablename_lbl (In the case of WORK library :-
dsname_variablename_lbl)
                4. Length of a variable. Macro Variable :-
library_dsname_variablename_len (In the case of WORK library :-
dsname_variablename_len)
                5. Length of the label of a variable. Macro Variable :-
library_dsname_variablename_lbllen (In the case of WORK library :-
dsname_variablename_lbllen)
                6. Length of the name of a variable. Macro Variable :-
library_dsname_variablename_namelen (In the case of WORK library :-
dsname_variablename_namelen)
NOTES:
MACROS USED: mu_help_debug mu_check_req_parameters mu_wordscan
mu_check_data_and_var_exist

PARAMETERS:
DATASETS: Dataset names from which the attributes of supplied variables will be
found. Every dataset will need
                 to be separated by SPACE delimiter.
VARIABLES: Variables for which the attributes are needed. Different variables from
same dataset will need to be
                  separated by SPACE deliemiter whereas variables from different
datasets will need to be separated by
                  ! delimiter.
DEBUG: If the value of DEBUG macro variable is set to Y then mprint merror mlogic
symbolgen system options will
           be turned on.
*************************************************************************;
```

```
%*EXAMPLES:

%mu_var_attributes(datasets=adb.adsl adb.adsf adeg,
                                    variables=usubjid age !
                                    dsdt scrnum !
                                    trtsdt trt01a,
                                    debug=Y, help=Y);
%*In above macro call the program will provide attributes for variables USUBJID and
AGE from ADB.ADSL dataset,
variables DSDT and SCRNUM from ADB.ADSF dataset, and also for the variables TRTSDT
and TRT01A from ADB.ADEG
dataset. There will be 6 global macro variables created for each variable. For
example for variable AGE 6 macro
variables will be ADB_ADSL_AGE_VARTYP, ADB_ADSL_AGE_FMT, ADB_ADSL_AGE_LBL,
ADB_ADSL_AGE_LEN, ADB_ADSL_AGE_LBLLEN,
ADB_ADSL_AGE_NAMELEN and for Variable TRT01A 6 macro variables will be
ADEG_TRT01A_VARTYP, ADEG_TRT01A_FMT,
ADEG_TRT01A_LBL, ADEG_TRT01A_LEN, ADEG_TRT01A_LBLLEN and ADEG_TRT01A_NAMELEN.

%*************************************************************************
MODIFIED BY:            DATE:
*************************************************************************;

%PUT ------------------------------------------------ ;
%PUT INFO: (&SYSMACRONAME);
%PUT INFO: Version 1.0;
%PUT -START----------------------------------------- ;

%mu_help_debug

%put help=&help;
%put debug =&debug;

%***** get current setting of mprint and mlogic;
%let mprint_setting=%sysfunc(getoption(mprint));
%let mlogic_setting=%sysfunc(getoption(mlogic));
%if &debug = Y %then %do;
        option mprint ;
%end;
%if &help = Y %then %do;
        option mlogic ;
%end;

%global MU_VAR_ATTRIBUTES_RC;
%let MU_VAR_ATTRIBUTES_RC=0;

%mu_check_req_parameters(parameters_to_check=datasets variables, help=no);
%let MU_VAR_ATTRIBUTES_RC=&MU_CHECK_REQ_PARAMETERS_RC.;

%if &MU_VAR_ATTRIBUTES_RC.>0 %then %do;
```

```
        %put ALERT_P: Parameters are not passed correctly. Program will ABORT.;
        %let MU_VAR_ATTRIBUTES_RC=1;
        %goto exit;
%end;

%* parse list of data sets;
%mu_wordscan(string=&datasets., root=data, numw=totaldata, delim=%str( ));
%* parse list of variables for each data set;
%let cnt_dset=%sysfunc(countw(&datasets," "));
%if &cnt_dset gt 1 %then %do;
        %mu_wordscan(string=&variables., root=vars_data, numw=totalvars,
delim=%str(!));
%end;
%else %do;
        %let vars_data1=&variables;
        %let totalvars=&cnt_dset;
%end;

%* ifthe number of data sets to does notmatch the number of groups of variables,
then abort;
%if &totaldata.^=&totalvars. %then %do;
        %put ALERT_P: Every Dataset does not have corresponding Variables. Program
will ABORT.;
        %let MU_VAR_ATTRIBUTES_RC=2;
        %goto exit;
%end;
%* otherwise, proceed to get attributes;
%else %do;

        %*cycle through each data set;
        %do _dt=1 %to &totaldata;

                %if %sysfunc(exist(&&data&_dt.))=1 %then %do;

                        %*cycle through each group of variables for current data
set;
                        %mu_wordscan(string=&&vars_data&_dt., root=va,
numw=totalvar, delim=%str( ));
                        %do _vars=1 %to &totalvar.;

                                %* first check that the variable exits;
                                %mu_check_data_and_var_exist
                                ( data_to_check=&&data&_dt.
                                , vars_to_check_in_all_data=&&va&_vars.
                                , vars_to_check_in_respective_data=
                                , abort_if_does_not_exist=No
                                , help=no
                                , debug=
                                )
```

```
                                %*set up global macvars for each attribute for each
variable;
                                data _null_;
                                length lib_ds $100;
                                lib_ds=tranwrd(strip("&&data&_dt."),".","_");
                                call symput ("lib_ds",strip(lib_ds));
                                run;
                                %global &lib_ds._&&&va&_vars.._vartyp
                                        &lib_ds._&&&va&_vars.._len
                                        &lib_ds._&&&va&_vars.._fmt
                                        &lib_ds._&&&va&_vars.._lbl
                                        &lib_ds._&&&va&_vars.._lbllen
                                        &lib_ds._&&&va&_vars.._namelen
                                        ;

                                %* open the current data set and get the attributes
for the current variable;
                                %let dsid=%sysfunc(open(&&data&_dt.));
                                %let varnum=%sysfunc(varnum(&dsid.,&&va&_vars.));
                                %if &varnum.>0 %then %do;
                                        %let
&lib_ds._&&&va&_vars.._vartyp=%sysfunc(vartype(&dsid.,&varnum.));
                                        %let
&lib_ds._&&&va&_vars.._len=%sysfunc(varlen(&dsid.,&varnum.));
                                        %let
&lib_ds._&&&va&_vars.._fmt=%sysfunc(varfmt(&dsid.,&varnum.));
                                        %let
&lib_ds._&&&va&_vars.._lbl=%bquote(%sysfunc(varlabel(&dsid.,&varnum.)));
                                        %*if "&&&&&&&&&lib_ds._&&&va&_vars.._lbl."
= "'" or "&&&&&&&&&lib_ds._&&&va&_vars.._lbl." = ""  %then %do;
                                        %if
%bquote(&&&&&&&&&lib_ds._&&&va&_vars.._lbl.) = %str( )  %then %do;
                                           %put &&va&_vars. does not have a label.;
                                           %let &lib_ds._&&&va&_vars.._lbl=;
                                         %end;
                                         %else %if
%qsubstr(&&&&&&&&&lib_ds._&&&va&_vars.._lbl.,1,1) = %str(%') %then %do;
                                           %put &&va&_vars. does not have a label.;
                                           %let &lib_ds._&&&va&_vars.._lbl=;
                                         %end;

                                        %if
%bquote(&&&&&&&&&lib_ds._&&&va&_vars.._lbl.) ^= %str() %then %do;
                                                %let
&lib_ds._&&&va&_vars.._lbllen=%sysfunc(length(&&&&&&&&&lib_ds._&&&va&_vars.._lbl.))
;
                                        %end;
                                        %let
&lib_ds._&&&va&_vars.._namelen=%sysfunc(length(&&va&_vars.));
```

```
                                        %put &&va&_vars.;
                                        %put &lib_ds._&&&va&_vars.._vartyp= =
&&&&&&&&&lib_ds._&&&va&_vars.._vartyp;
                                        %put &lib_ds._&&&va&_vars.._len= =
&&&&&&&&&lib_ds._&&&va&_vars.._len;
                                        %put &lib_ds._&&&va&_vars.._fmt= =
&&&&&&&&&lib_ds._&&&va&_vars.._fmt;
                                        %put &lib_ds._&&&va&_vars.._lbl= =
&&&&&&&&&lib_ds._&&&va&_vars.._lbl;
                                        %put &lib_ds._&&&va&_vars.._lbllen =
&&&&&&&&&lib_ds._&&&va&_vars.._lbllen.;
                                        %put &lib_ds._&&&va&_vars.._namelen= =
&&&&&&&&&lib_ds._&&&va&_vars.._namelen;

                                %end;
                                %else %do;
                                        %put ALERT_C: Variable &&va&_vars. does not
exist in Dataset &&data&_dt..;
                                        %let MU_VAR_ATTRIBUTES_RC=4;
                                %end;
                                %let rc=%sysfunc(close(&dsid.));

                        %end;

                %end;
                %else %do;
                        %put ALERT_C: Dataset &&data&_dt. does not exist.;
                        %let MU_VAR_ATTRIBUTES_RC=3;
                %end;

        %end;

%end;

%exit:
options &mprint_setting &mlogic_setting;
%PUT ---------------------------------------------- ;
%PUT INFO: (&SYSMACRONAME);
%PUT INFO: Version 1.0;
%put MU_VAR_ATTRIBUTES_RC=&MU_VAR_ATTRIBUTES_RC;
%PUT -END----------------------------------------- ;

%mend mu_var_attributes;

%mu_var_attributes(datasets=sashelp.bmt /*sashelp.bweight*/,
                                variables=group t/*!black married*/,
                                debug=Y, help=Y);
```