```
%macro mw_format_footnote_reference(
 in_data  =
,out_data =
,var     =
,fmtname =
,help    = N
,debug   = N
)/ store des='V1.0.0.4' ;

%*************************************************************************
************************;
%*         MACRO NAME:  mw_format_footnote_reference
                                                              ;
%*        SAS VERSION:  SAS V9.3 and SAS V9.4 on PC and SAS Grid
                                             ;
%*            PURPOSE:  Analyze a string variable and translate footnote footnote
references per MW guidance;
%*        USAGE NOTES:  A work level format (FMTNAME) must be available The macro
is stand alone and can be ;
%*                      called independently.
                                                    ;
%*

;
%*   INPUT PARAMATERS:
                                                              ;
%*        IN_DATA  = Dataset with final structure which will be displayed in output
and variable having     ;
%*                   footnote referance will be provided in this parameter
(Required).                                 ;
%*        OUT_DATA = The resulting data name which will be created after update in
footnote referance as per;
%*                                            MW requirement based on format provided in
FMTNAME patameter (Required).                  ;
%*        VAR      = The string variable name in IN_DATA to that contains footnote
references. Same variable;
%*                   in output dataset will have updated footnote referance
(Required).                          ;
%*        FMTNAME  = The work level format that will translate references in
variable provided in VAR                ;
%*                                            parameter (Required).

       ;
%*        DEBUG    = Set to Y(es) to output helpful hints to the log (Optional,
Default=N).                                       ;
%*        HELP     = Set to Y(es) to save all intermediate datasets and to turn on
mprint. Set to N(o) to   ;
%*                   delete all intermediate datasets and to not turn on mprint .
                                                  ;
%*                   (Optional, default=&_default_help)
```

```
                                                          ;
%*

                                                ;
%*       OUTPUT FILES:  See out_data
                                                              ;
%*
                                                              ;
%*             AUTHOR:   (b) (6)
                                                          ;
%*       DATE CREATED:  31Dec2015
                                                          ;
%*
                                                              ;
%*   MODIFICATION LOG:
                                                          ;
%*************************************************************************
************************;
%* DATE          INITIALS        MODIFICATION DESCRIPTION
                                                          ;
%*************************************************************************
************************;
%* 06FEB2019     (b) (6)                 Macro updated to add code as per ARTz
requirement to check parameters and  ;
%*                                                                  Display logmessage
after conditional check.
;
%* 02OCT2019     (b) (6)        Set version to Version 1.0.0.0.
                        ;
%*************************************************************************
************************;
%*  (c) 2019 PPD
                 ;
%*  All Rights Reserved.
                                                          ;
%*************************************************************************
************************;

%*-------------------------------------------------------------------------
--*;
  %PUT ------------------------------------------------ ;
  %PUT INFO: (&SYSMACRONAME) ;
  %PUT INFO: Version V1.0.0.4 ;
  %PUT -START------------------------------------------- ;

%global &sysmacroname._rc;
%let &sysmacroname._rc = 0;

%** RETURN CODES
0= ran without error
```

```
1= output data set name not provided.
2= output data set already exists and will be overwritten.
3= Format specified does not exist.
4= Required parameter FMTNAME is blank.
5= Variable specified within VAR does not exist within input dataset.
6= Required parameter VAR is blank.
7= Macro is being executed by encoding other than WLATIN1.
;

%**** STEP 1 - SETUP AND PRELIMINARY CHECKS***********;

%* set help and debug;
%mu_help_debug *;
%let abort = NO;

%* get a list of the datasets in the WORK library before starting to enable cleanup
at
   the end of this macro;

%md_workinfo(
      debug        = &debug
    ,_workdata     = WORK_DATASETS_DATA
    ,_workview     = WORK_DATASETS_VIEW
    ,_mprinttoggle = mprint_setting
   ) *;


%****************** REQUIRED PARAMETERS***************;
%mu_check_req_parameters(
 parameters_to_check = IN_DATA
,help = &help
)
;

%***********************************************;
%** Check encoding Session                     **;
%***********************************************;

%let session_encoding=%upcase(%sysfunc(getoption(encoding)));

     %mu_checkrc(condition  = "&session_encoding." ^= "WLATIN1"
                 ,setabort = Y
                 ,rc       = 7
                 ,rcprefix = MW_FORMAT_FOOTNOTE_REFERENCE
                 ,msg1     = Please note macro has not been fully validated
within &session_encoding.
                          ,msg2         = INFO: MW_FORMAT_FOOTNOTE_REFERENCE_RC =
7
                     );
```

```
%*****STEP 2 - CONDITIONAL PARAMETERS CHECKS************;
%if &OUT_DATA = %str() %then %do;
                        %mu_checkrc(condition  = &OUT_DATA =
                   ,setabort = Y
                   ,rc       = 1
                   ,rcprefix = MW_FORMAT_FOOTNOTE_REFERENCE
                   ,msg1     = Required parameter OUT_DATA is blank.
                           ,msg2         = INFO: MW_FORMAT_FOOTNOTE_REFERENCE_RC =
1
                        );
%end;
%else %if %sysfunc(exist(&out_data)) %then %do;
        %put ALERT_I: OUT_DATA=&OUT_DATA.  Data set already exists and will be
deleted/overwritten;
        %let &sysmacroname._RC = 2;
        proc datasets mt=data lib=work;
          delete &out_data;
          run;
        quit;
%end;
%if &fmtname ^=  %then %do;
        %* get RC from formats ***;
        proc format lib=work
           cntlout=fmtname
(where=(fmtname=upcase(compress(compress("&fmtname",'.'),'$')))) ;
        run ;

        %mu_nobs(FMTNAME);

        %if &fmtname_nobs = 0 %then %do;
                %let &&SYSMACRONAME._RC = 3;
        %end;
        %mu_checkrc(condition  = &fmtname_nobs = 0
                   ,setabort = Y
                   ,rc       = 3
                   ,rcprefix = MW_FORMAT_FOOTNOTE_REFERENCE
                   ,msg1     = Format specified does not exist.
                           ,msg2         = INFO: MW_FORMAT_FOOTNOTE_REFERENCE_RC =
3
                        );
%end;

%if &fmtname =  %then %do;
        %* get RC from formats ***;

        %mu_checkrc(condition  = &fmtname =
                   ,setabort = Y
                   ,rc       = 4
                   ,rcprefix = MW_FORMAT_FOOTNOTE_REFERENCE
                   ,msg1     = Required parameter FMTNAME is blank.
```

```
                           ,msg2         = INFO: MW_FORMAT_FOOTNOTE_REFERENCE_RC =
4
                       );
%end;
%if &var ^= %str() %then %do;
%* get RC from VAR not exist ***;
        proc sql;
                create table VAR as select * from dictionary.columns
                where libname='WORK' and memtype='DATA' and
memname=%upcase("&IN_DATA") and upcase(name)=%upcase("&var.");
        quit;

        %mu_nobs(VAR);

        %mu_checkrc(condition  = &var_nobs = 0
                   ,setabort = Y
                   ,rc       = 5
                   ,rcprefix = MW_FORMAT_FOOTNOTE_REFERENCE
                   ,msg1     = Variable specified within VAR does not exist within
input dataset.
                           ,msg2         = INFO: MW_FORMAT_FOOTNOTE_REFERENCE_RC =
5
                       );
%end;
%if &var = %str() %then %do;
%* get RC from VAR not exist ***;

        %mu_checkrc(condition  = &var = %str()
                   ,setabort = Y
                   ,rc       = 6
                   ,rcprefix = MW_FORMAT_FOOTNOTE_REFERENCE
                   ,msg1     = Required parameter VAR is blank.
                           ,msg2         = INFO: MW_FORMAT_FOOTNOTE_REFERENCE_RC =
6
                       );
%end;

%*-------------------------------------------------------------------------------
--*;
%* Format missing;
%if %index(&fmtname, $) eq 0 %then %do;

    %put %str(ALERT_I:
-------------------------------------------------------------------------------
--);
    %put %str(ALERT_I: FMTNAME paramenter is missing or does not specify a
Format.);
    %put %str(ALERT_I: Footnote referance will not be created);
    %if &help eq Y %then %do;
        %put %str(HELP: Make sure Format created and exists in format catalog );
```

```
        %put %str(HELP: assigned to the FMTNAME parameter.);
    %end;
    %put %str(ALERT_I:
----------------------------------------------------------------------------------
--);
%end;
 %put &fmtname;

*******STEP 3 -MW_FORMAT_FOOTNOTE_REFERENCE MACRO STARTS ******;
%mu_get_sort_order(&in_data.) *;

proc format lib=work cntlout=myfmts;
run;

data fmtlist;
 length list $32000;
 retain list;
 set myfmts(where=(upcase(fmtname)=upcase(compress("&fmtname.", "$.")))) end=eof;

 if _n_=1 then list=start;
  else list=strip(list)||', '||strip(start);

 if eof then do;
   call symput('footlist', strip(list));
 end;
run;

%put Footnote list=&footlist.;

data &out_data.(drop=footref list j sortedby=&SORT_ORDER.);
  length footref $200. list $32000;
  set &in_data.;
  list=strip(tranwrd(symget('footlist'), "'", ''));
  if _n_=1 then
    put "ALER" "T_I: Scanning Footnote Reference List: " list;

  j=1;
  do while(strip(scan(list, j, ','))^='');
    footref=strip(scan(list, j, ','));

    if index(upcase(strip(&var.)), upcase(strip(footref))) then do;
      put "ALER" "T_I: Found reserved term " footref;
          ** mark it;
      &var.=tranwrd(&var., strip(footref), strip(put(footref, &fmtname.)));
        end;
        j+1;
  end;
run;

%********** STEP 4 - CLEAN UP AND RESET******;
```

```
%md_clean_and_reset
( debug      = &debug
,_workdata   = %str(&WORK_DATASETS_DATA &out_data)
,_workview   = %str(&WORK_DATASETS_VIEW)
,resetmprint = &mprint_setting
) *;


  %PUT ---------------------------------------------- ;
  %PUT INFO: (&SYSMACRONAME) ;
  %PUT INFO: Version V1.0.0.0 ;
  %put &SYSMACRONAME._RC = &&&SYSMACRONAME._RC;
  %PUT -END------------------------------------------ ;

%mend mw_format_footnote_reference;
```