

Get your SAS in gear – Automate the Production of Analysis Datasets

Liz Taylor, Endo Pharmaceuticals, Chadds Ford, PA

ABSTRACT

The production of Analysis Datasets can be greatly simplified by using SAS Macros to perform the following functions:

- Automatically generate a Decode Variable for every Coded Variable in the Raw Dataset or any Work Dataset
- Reorganize the order of the variables in the Analysis Dataset so that the Coded and Decode variables are together

For an Analysis Dataset that does not contain any derived variables, these Macros can be used to read in a Raw Dataset.

For an Analysis Dataset that requires derived variables, these Macros can be used to read in a Work Dataset in which the derived variables have already been generated.

INTRODUCTION

This paper discusses how to use SAS Macros to make it easier to generate Analysis Datasets by performing the standard tasks that are required.

Important considerations:

- How to Define a Decode Variable
 - Manually
 - Automatically
 - How do I add a single Decode Variable?
 - How do I identify which Variables need to be decoded?
 - How do I define the characteristics of the Decode Variables?
 - Decode Variable Name
 - Decode Variable Label
 - Decode Variable Type and Length
 - How do I add the Decode Variables to my Analysis Dataset?
- How to reposition the Variables so that Code and Decode variables are next to each other?
 - Manually
 - Automatically
- How to integrate these procedures into an Analysis Dataset program

How to Define a Decode Variable

Defining a Decode Variable involves the following steps:

- Read in the Input File
- Identify variables that are formatted (Coded Variables)
- Define the a variable to contain the formatted value (Decode Variable)
- If the Coded Variable is not a Date/Time Variable, unformat the Coded Variable
- Place the formatted value in the Decode Variable
- Define a label for the Decode Variable
- Write the Output File

Manually

The following Code will add several Decode Variables to a File

```
data adata.demog;
  set rdata.demog;
  length BIRTH_DT $9. DEMEDC_D $21. GENDER_D $6.;
  BIRTH_DT = trim(left(put(BIRTHDT,date9.)));
  DEMEDC_D = trim(left(DEMEDCON,diag21.));
  GENDER_D = trim(left(GENDER,sex6.));
  label BIRTH_DT = 'Date of Birth (TEXT)'
        DEMEDC_D = 'Study Indication (DECODE)'
        GENDER_D = 'Gender (DECODE)';
run;
```

Note: Decode Variables will be positioned at the end of the dataset. This involves a lot of work for the programmer if there are a lot of Coded Variables

Automatically

How do I add a single Decode Variable?

The following Macro %cd2dc will add a Decode Variable to a File

```
/* ***** */
/* Add a Decode Variable to Dataset */
/* ***** */
%macro cd2dc (inds=, outds=, cvar=, dvar=, fmt=, dvarlen=, label=, unfmt=Y);
  /* Parameters */
  /* inds - Input Dataset Name - name of a Permanent or Work SAS File */
  /* outds - Output Dataset Name - name of a Permanent or Work SAS File */
  /* cvar (Coded Var Name) - Name of a Var that has a Format associated with it */
  /* dvar (Decode Var Name) - Name of the Var that will contain the Decode Value */
  /* fmt - Format value as used in a PUT statement - such as 'elig_d = put(elig,YESNO3.);' */
  /* dvarlen - Length of var as used in a Length statement - such as 'length elig_d $3.;' */
  /* label - Label of Decode Var */

  /* Define Decode Variable (&dvar) using formatted length (&dvarlen) */
  /* Remove format from Code Variable (&cvar) if not a Date/Time Var */
  /* Move formatted value to Decode Variable */
  /* Define Label */
  data &outds.;
    set &inds.;
    /* Define Decode Variable (&dvar) using formatted length (&dvarlen) */
    length &dvar. &dvarlen.;

    /* Remove format from Code Variable (&cvar) if not a Date/Time Var */
    %if %upcase(&unfmt.) = Y %then
```

```

%do;
  format &cvar.;
%end;

/* Move formatted value to Decode Variable */
&dvar. = trim(left(put(&cvar., &fmt.)));

/* Define Label */
label &dvar. = "&label";

run;
%mend;

```

Using the %cd2dc Macro, the following code will add the same Decode Variables to the Demog File:

```

%cd2dc (inds=adata.demog, outds=adata.demog, cvar=BIRTHDT, dvar=BIRTH_DT, fmt=date9.,
dvarlen=$9., label= Date of Birth (TEXT), unfmt=N);

%cd2dc (inds=adata.demog, outds=adata.demog, cvar=DEMEDCON, dvar=DEMEDC_D, fmt=diag21.,
dvarlen=$21., label= Study Indication (DECODE), unfmt=Y);

%cd2dc (inds=adata.demog, outds=adata.demog, cvar=GENDER, dvar=GENDER_D, fmt=sex6., dvarlen=$6.,
label= Sex (DECODE), unfmt=Y);

```

Note: Decode Variables will still be positioned at the end of the File and %cd2dc must be called for every Coded Variable.

This still involves a lot of work for the programmer if there are a lot of Coded Variables..

How do I identify which Variables need to be decoded?

Decode Variables need to be generated for any Variable that has a format associated with it (Coded Variable)

- Date and Time Decode Variables contain the character representation of the Coded Variable
- Other Decode Variables contain the Formatted Value (taken from the Format Library) and are either character or numeric depending upon the Format of the Coded Variable

-----Alphabetic List of Variables and Attributes-----

#	Name	Type	Len	Pos	Format	Label
.
11	AMEND	Num	8	32	1.	Amendment
7	BIRTHDT	Num	8	8	DATE9.	Date of Birth
14	DEMEDCON	Num	8	56	DIAG.	Study Indication
15	DIAGDT	Num	8	64	DATE9.	Low Back Pain Diagnosis Date
16	DIAGDTC	Char	10	232	\$10.	Low Back Pain Diagnosis Date - Character
17	ETIOL1	Num	8	72	YESNO.	Etiology: Osteoarthritis
18	ETIOL2	Num	8	80	YESNO.	Etiology: Rheumatoid Arthritis
19	ETIOL3	Num	8	88	YESNO.	Etiology: Herniated Disc
24	ETIOL4S	Char	100	242	\$100.	Etiology: Trauma, Specify
25	ETIOL5S	Char	100	342	\$100.	Etiology: Other, Specify
10	GENDER	Num	8	24	SEX.	Sex
12	INFCON	Num	8	40	DATE9.	Date Informed Consent Signed
13	INFCONTM	Num	8	48	TIME5.	Time Informed Consent Signed
4	INIT	Char	3	164	\$3.	Init
27	INV	Char	50	442	\$50.	Investigator

Shaded variables are formatted and will require a Decode variable

How do I define the characteristics of the Decode Variable?

- Using Proc Contents, Generate a File containing a List of Decode Variables with the following information:
 - Coded Variable Name (cvar) – Renamed from the original Coded Variable Name (name)
 - Decode Variable Name (dvar) - Add a suffix to the Coded Variable Name depending upon the Format (Date, Time, DateTime, Other)
 - Date=_DT, Time=_TM, Datetime=_XX, Other=_D
 - Decode Variable Label (label) - Add a suffix to the end of the Coded Variable Label
 - Label Length (lbl_len) - The length of the Decode Variable Label is also generated in order to be able to flag it later on in case the length exceeds 40 characters

varnum	cvar	format	dvar	lbl_len	label
7	BIRTHDT	DATE	BIRTH_DT	20	Date of Birth (TEXT)
14	DEMEDCON	DIAG	DEMEDC_D	25	Study Indication (DECODE)
15	DIAGDT	DATE	DIAGD_DT	35	Low Back Pain Diagnosis Date (TEXT)
17	ETIOL1	YESNO	ETIOL1_D	33	Etiology: Osteoarthritis (DECODE)
18	ETIOL2	YESNO	ETIOL2_D	34	Etiology: Rheumatoid Arth (DECODE)
19	ETIOL3	YESNO	ETIOL3_D	33	Etiology: Herniated Disc (DECODE)
10	GENDER	SEX	GENDER_D	12	Sex (DECODE)
12	INFCON	DATE	INFCO_DT	35	Date Informed Consent Signed (TEXT)
13	INFCONTM	TIME	INFCO_TM	35	Time Informed Consent Signed (TEXT)

- Using Proc Format, Generate a File containing a List of Formats with a single record for each Format in the Format Library and the following information:
 - Format Name (format)
 - Length of the Format (dvarlen) – Length of the Format + ‘.’
 - Format (fmt) – Combined Format Name and Length of the Format

format	dvarlen	fmt
DIAG	21.	diag21.
SEX	6.	sex6.
YESNO	3.	yesno3.

- Merge the List of Decode Variables with the List of Formats by Format Name to add dvarlen and fmt to each Decode Variable
 - Since Date and Time Formats will not be in the Format library, dvarlen and fmt must be hardcoded

varnum	cvar	format	dvar	lbl_len	label	dvarlen	fmt
7	BIRTHDT	DATE	BIRTH_DT	20	Date of Birth (TEXT)	\$9.	date9.
14	DEMEDCON	DIAG	DEMEDC_D	25	Study Indication (DECODE)	\$21.	diag21.
15	DIAGDT	DATE	DIAGD_DT	35	Low Back Pain Diagnosis Date (TEXT)	\$9.	date9.
17	ETIOL1	YESNO	ETIOL1_D	33	Etiology: Osteoarthritis (DECODE)	\$3.	yesno3.
18	ETIOL2	YESNO	ETIOL2_D	34	Etiology: Rheumatoid Arth (DECODE)	\$3.	yesno3.
19	ETIOL3	YESNO	ETIOL3_D	33	Etiology: Herniated Disc (DECODE)	\$3.	yesno3.
10	GENDER	SEX	GENDER_D	12	Sex (DECODE)	\$6.	sex6.
12	INFCON	DATE	INFCO_DT	35	Date Informed Consent Signed (TEXT)	\$9.	date9.
13	INFCONTM	TIME	INFCO_TM	35	Time Informed Consent Signed (TEXT)	\$5.	time5.

How do I add the Decode Variables to my Analysis Dataset?

- For each Decode Variable generate Macro Variables to define:
 - Coded Variable Name (&cvar)
 - Decode Variable Name (&dvar)
 - Format Length (&fmt)

- Combined Format Name and Length (&dvarlen)
- Decode Variable Label (&label)
- Execute a Macro that will Read a Raw or Work Dataset and execute the following statements:
 - length &dvar &dvarlen.;
 - &dvar. = trim(left(put(&cvar.,&fmt.));
 - label &dvar. = "&label.";
- Repeat this procedure for each Decode Variable
- The following Macro (%cd2dc_auto) contains the code to perform this task.
- In addition to generating Decode Variables, it also generates a Report identifying all the Coded Variables for which a Decode Variable needs to be generated and it identifies any problems such as:
 - Labels that exceed 40 characters – the Decode Variable will be generated, but the Label will have to be modified manually
 - Coded Variables that have a Format for which there is no corresponding Format in the Format Library – the Decode Variable will not be generated
 - Coded Variables for which the Decode Variable Name is not unique – the Decode Variable will not be generated
 - For example: Coded Variables INCLUS1, INCLUS2, and INCLUS3 will all have a Decode Variable Name of INCLUS_D

```

/* ***** */
/* Automatically add Decode Vars to a File */
/* ***** */
%macro cd2dc_auto (inds=, outds=);
  /* Parameters */
  /* inds - Input Dataset Name - name of a Permanent or Work SAS File */
  /* outds - Output Dataset Name - name of a Permanent or Work SAS File */

  /* Copy Input Dataset to Output Dataset */
  data &outds.;
    set &inds.;
  run;

  /* Create a Dataset (contents) containing the Output from Proc Contents */
  proc contents data=&inds. noprint out=contents;
  run;

  /* Create a Dataset (code_vars_all) with the following variables: */
  /* cvar (Coded Var Name) - Name of a Var that has a Format associated with it */
  /* dvar (Decode Var Name) - Name of the Var that will contain the Decode Value */
  /* This is done by adding a suffix to end of the Coded Var Name replacing some */
  /* characters in the Coded Var Name if necessary */
  /* Note: This can cause a problem if Coded Var Names are similar */
  /* For example: exclus1 & exclus2 will both generate a Decode Var Name of exclus_d */
  /* One solution would be to change the Coded Var name prior to executing the macro */
  /* For example: exclus1 renamed to excl1 generates a Decode Var name of excl1_d */
  /* format (Format Name) */
  /* label (Label of Decode Var) */
  /* lbl_len (Length of Decode Var Label) */
  /* varnum (Variable Number) */
  data code_vars_all (keep=cvar dvar format label lbl_len varnum);
    length label $50.;
    set contents end=eof;
    where not missing(format) and format ne '$';
    length cvar dvar $8. lbl_len 8.;

    cvar = name;

    format = trim(left(uppercase(format)));

```

```

/* Generate Decode Variable Name and Label */
/* Suffix for Date Decode Variable Name is _DX, Suffix for Label is '(TEXT)' */
if format = 'DATE' then
do;
dvar = trim(left(substr(name,1,5))) || '_DX';
label = trim(left(label)) || ' (TEXT)';
end;
else

/* Suffix for Time Decode Variable Name is _TX, Suffix for Label is '(TEXT)' */
if format = 'TIME' then
do;
dvar = trim(left(substr(name,1,5))) || '_TX';
label = trim(left(label)) || ' (TEXT)';
end;
else

/* Suffix for DateTime Decode Variable Name is _XX, Suffix for Label is '(TEXT)' */
if format = 'DATETIME' then
do;
dvar = trim(left(substr(name,1,5))) || '_XX';
label = trim(left(label)) || ' (TEXT)';
end;
else

/* Suffix for any other Variable Name is _D, Suffix for Label is '(DECODE)' */
else
do;
dvar = trim(left(substr(name,1,6))) || '_D';
label = trim(left(label)) || ' (DECODE)';
end;

lbl_len = length(label);
run;

proc format lib=work cntlout=fmts;
run;

proc sort data=fmts;
by fmtname type;
run;

/* Generate a file containing a single record/format */
data fmt_defs (keep=fmtname format fmt dvarlen);
set fmts;
by fmtname type;
length format $8. fmt $12. dvarlen $5.;

if first.type then
do;
format = trim(left(uppercase(fmtname)));

/* Add a '$' to the beginning of Format Name if a Char Format */
if type eq 'C' then
format = '$' || trim(left(format));

/* Combine Format Name and Format Length - such a DATE9. or YESNO3. */
fmt = trim(left(format)) || trim(left(put(length,3.))) || '.';

/* Define Length of Decode Var - such as $9. or $3. */
dvarlen = '$' || trim(left(put(length,3.))) || '.';

output;
end;
run;

/* Merge Code Vars All & Fmt_Defs to add Format (fmt) & Decode Length (dvarlen) */
proc sort data=fmt_defs;
by format;
run;

```

```

proc sort data=code_vars_all;
  by format;
run;

data code_vars_all;
  merge code_vars_all (in=var)
        fmt_defs;
  by format;
  if var;

  length fmt $12. varlen dvarlen $5.;
  /* Date/Time Formats will not be in the Format Definitions File and must be hardcoded */
  if format eq 'DATE' then
    varlen = '9.';
  if format eq 'TIME' then
    varlen = '5.';
  if format eq 'DATETIME' then
    varlen = '16.';
  if format in ('DATE','TIME','DATETIME') then
  do;
    fmt = trim(left(format)) || trim(left(varlen));
    dvarlen = '$' || trim(left(varlen));
  end;

  /* Identify problems */
  if missing(dvar) then
    code_flag = 'M';
  /* No matching Format */
  if missing(fmt) then
    code_flag = 'F';
  /* Decode var Label too long */
  if lbl_len gt 40 then
    code_flag = 'L';
run;

proc sort data=code_vars_all;
  by dvar;
run;

/* Generate a File (Code Vars Sel) containing only Code Vars that have no problems */
/* If Label is too long generate Decode anyway and modify Label later using Proc Datasets */
data code_vars_sel
  code_vars_all;
  set code_vars_all;
  by dvar;
  /* Identify additional problems */
  /* Duplicate Decode Vars */
  if not (first.dvar and last.dvar) then
    code_flag = 'D';
  if missing(code_flag) or code_flag eq 'L' then
    output code_vars_sel; /* Do not generate Decode if a problem */
  output code_vars_all;
run;

proc sort data=code_vars_sel;
  by varnum;
run;

/* Generate a Macro Variable containing the Number of Records in the Code_Vars Dataset */
%global num_obs;
data _null_;
  set sashelp.vtable;
  if upcase(libname)='WORK' and upcase(memname) eq 'CODE_VARS_SEL' then
    call symput('num_obs',nobs);
run;

/* For each record in the Code_Vars Dataset, generate the Decode Var */
%do i = 1 %to &num_obs;
  data _null_;
    set code_vars_sel;

```

```

        if _n_ eq &i;
        length unfmt $1.;
        /* Remove Format unless Code Var is a Date or Time */
        if format in ('DATE', 'TIME', 'DATETIME', 'MMDYY') then
            unfmt = 'N';
        else
            unfmt = 'Y';
        call symput('cvar',cvar);
        call symput('dvar',dvar);
        call symput('fmt',fmt);
        call symput('dvarlen',dvarlen);
        call symput('label',label);
        call symput('unfmt',unfmt);
        %cd2dc (inds=&outds, outds=&outds);
    run;
%end;

/* Print Code/Decode List */
title "Derived Decode Variables (&inds.)";
proc report data=code_vars all &clin headline headskip split='#' spacing=1 nowindows missing;
    columns varnum cvar dvar dvar_auto fmt label lbl_len code_flag;
    define varnum /order noprint;
    define cvar /width=8 'Code Var';
    define dvar /width=8 'DeCode Var';
    define dvar_auto /width=8 'Auto DeCode Var';
    define fmt /width=12 'Format';
    define label /width=40 flow 'Label';
    define lbl_len /width=3 'Len';
    define code_flag /width=3 'Flg';
run;
%mend;

```

How to reposition the Variables so that Code and Decode variables are next to each other?

Manually

Generate a RETAIN statement to reposition the variables.

```

data demog;
    retain protocol siteid subjid init birthdt birth_dt gender gender_d amend infcon infco_dt
        infcotm infco_tm demedcon demedc_d diagdt diagd_dt etiol1 etiol1_d etiol2 etiol2_d etiol3
        etiol3_d etiol4s etiol5s inv;
run;

```

This is time consuming if there are a lot of variables.

Automatically

The following Macro (%reorg) contains the code to automate this task

- Read the list of Decode variables which contains the Variable Number from the Coded Variable and increment the Variable Number by .5
- Combine the List of Decode Variables with the List of Variables from the Proc Contents and sort the new list by Variable Number.
- Generate a Macro Variable containing all the Variable names in order by Variable Number
- Use the Macro Variable in a RETAIN Statement to reposition the variables

```

/* ***** */
/* Reorganize Dataset to put Code and Decode Variables together */
/* ***** */
%macro reorg(inds=, outds=);
  /* Parameters */
  /* inds = Input Dataset Name - maybe a Permanent or Work SAS File */
  /* outds = Output Dataset Name - maybe a Permanent or Work SAS File */

  /* Create a list of Decode Vars incrementing var num of Coded Var by .5 */
  data decode_vars_lst (keep=name varnum);
    set code_vars_sel;
    /* Increment Varnun (which comes from the Code Var) by .5 */
    varnum = varnum + .5;
    /* Rename dvar to name */
    rename dvar = name;
  run;

  /* Combine List of Vars from Proc Contents (which contains original Vars) */
  /* with List of Decode Vars */
  data var_lst (keep=name varnum);
    set contents (in=contents)
        decode_vars_lst;
  run;

  proc sort data=var_lst;
    by varnum;
  run;

  /* Generate a Macro Variable (var1_lst, var2_lst) containing a list of Var Names in order by
  Varnum */
  data var_lst;
    set var_lst end=eof;
    length var1_lst var2_lst $500.;
    retain var1_lst var2_lst;
    if length(var1_lst) le 490 then
      var1_lst = trim(left(var1_lst)) || ' ' || trim(left(name));
    else
      var2_lst = trim(left(var2_lst)) || ' ' || trim(left(name));
    if eof then
      do;
        var1_lst = trim(left(var1_lst));
        call symput('var1_lst',var1_lst);
        var2_lst = trim(left(var2_lst));
        call symput('var2_lst',var2_lst);
      end;
  run;

  /* Rewrite file using Retain statement to organize Vars */
  data &outds.;
    retain &var1_lst &var2_lst;
    set &outds.;
  run;
%mend;

```

How to integrate these procedures into an Analysis Dataset program

Combine these two procedures.

```

%cd2dc_auto(inds=rdata.demog, outds=demog);
%reorg(inds=demog, outds=adata.demog);

```

CONCLUSION

These Macros can be used to dramatically reduce the time and effort required to decode each formatted variable in a dataset.

CONTACT INFORMATION

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

Liz Taylor
Endo Pharmaceuticals Inc.
100 Endo Blvd.
Chadds Ford, PA 19317

Phone: (610) 558-9800 x4268
Email: taylor.liz@endo.com