SAS programs for making SDTM DM and EX datasets

Yupeng Wang, Ph.D., Data Scientist

Raw data

1. Metadata

OID	NAME	REPEATING	ISREFERENCEDATA	PURPOSE	LABEL	STRUCTURE	DOMAINKEYS	CLASS	ARCHIVELOCATIONI
AE	AE	Yes	No	Tabulation	Adverse Events	Events - One record per event per subject	STUDYID, USUBJID, AEDECOD, AESTDTC	Events	./ae
DM	DM	No	No	Tabulation	Demographics	Special Purpose - One record per event per subject	STUDYID, USUBJID	Special Purpose	./dm
EX	EX	Yes	No	Tabulation	Exposure	One record per constant dosing interval per subject	STUDYID, USUBJID, EXTRT, EXSTDTC	Interventions	./ex
LB	LB	Yes	No	Tabulation	Laboratory Tests	Findings - One record per lab test per subject	STUDYID, USUBJID, LBCAT, LBTESTCD, VISITNUM	Findings	./lb
XP	XP	Yes	No	Tabulation	Pain Scores	One record per subject per visit	STUDYID, USUBJID, XPTESTCD, VISITNUM	Findings	./xp
TA	TA	Yes	Yes	Tabulation	Trial Arms	One record per planned Element per Arm	STUDYID, ARMCD, TAETORD	Trial Design	./ta
TE	TE	Yes	Yes	Tabulation	Trial Elements	One record per planned Element	STUDYID, ETCD	Trial Design	./te
TI	TI	Yes	Yes	Tabulation	Trial Inclusion/Exclusion Criteria	One record per I/E criterion	STUDYID, IETESTCD	Trial Design	./ti
TS	TS	Yes	Yes	Tabulation	Trial Summary	One record per trial summary parameter value	STUDYID, TSSEQ	Trial Design	./ts
TV	TV	Yes	Yes	Tabulation	Trial Visits	One record per planned Visit per Arm	STUDYID, VISITNUM, ARMCD	Trial Design	./tv
SUPPDM	SUPPDM	Yes	No	Tabulation	DM - Supplemental Qualifiers	Supplemental Qualifier - One record per qualifier	STUDYID, RDOMAIN, USUBJID, IDVAR, IDVARVAL, QNAM	Supplemental Qualifier	./suppdm

DEFINE_HEADER_METADATA | TOC_METADATA | VARIABLE_METADATA | VALUELEVEL_METADATA | COMPUTATION_METHOD | CODELISTS

DOMAIN	VARNUM	VARIABLE	TYPE	LENGTH	LABEL	SIGNIFIC	ORIGIN	COMME	DISPL	COMPU	CODELISTN	MANDATORY	ROLE	ROLECODELIS	VALUELISTOID
AE	1	STUDYID	text	15	Study Identifier		Derived					Yes	Identifier	ROLECODE	
AE	2	DOMAIN	text	2	Domain Abbreviation		Derived					Yes	Identifier	ROLECODE	
AE	3	USUBJID	text	25	Unique Subject Identifier		Derived					Yes	Identifier	ROLECODE	
AE	4	AESEQ	integer	8	Sequence Number		Derived					Yes	Identifier	ROLECODE	
AE	5	AETERM	text	200	Reported Term for the Adverse Event		CRF page 6					Yes	Topic	ROLECODE	
AE	6	AEDECOD	text	200	Dictionary-Derived Term		Derived				AEDECOD	Yes	Synonym Qualifier	ROLECODE	
AE	7	AEBODSYS	text	200	Body System or Organ Class		Derived				AEBODSYS	No	Record Qualifier	ROLECODE	
AE	8	AESEV	text	40	Severity/Intensity		CRF page 6				AESEV	No	Record Qualifier	ROLECODE	
AE	9	AESER	text	40	Serious Event		CRF page 6				NY	No	Record Qualifier	ROLECODE	
AE	10	AEACN	text	40	Action Taken with Study Treatment		CRF page 6				ACN	No	Record Qualifier	ROLECODE	
AE	11	AEREL	text	40	Causality		CRF page 6				AEREL	No	Record Qualifier	ROLECODE	
AE	12	AESTDTC	date	16	Start Date/Time of Adverse Event		CRF page 6					No	Timing	ROLECODE	
AE	13	AEENDTC	date	16	End Date/Time of Adverse Event		CRF page 6					No	Timing	ROLECODE	
AE	14	AESTDY	integer	8	Study Day of Start of Adverse Event		Derived					No	Timing	ROLECODE	
AE	15	AEENDY	integer	8	Study Day of End of Adverse Event		Derived					No	Timing	ROLECODE	
DM	1	STUDYID	text	15	Study Identifier		Derived					Yes	Identifier	ROLECODE	
DM	2	DOMAIN	text	2	Domain Abbreviation		Derived					Yes	Identifier	ROLECODE	
DM	3	USUBJID	text	25	Unique Subject Identifier		Derived					Yes	Identifier	ROLECODE	
DM	4	SUBJID	text	7	Subject Identifier for the Study		CRF page 1					Yes	Identifier	ROLECODE	
DM	5	RFSTDTC	date	16	Subject Reference Start Date/Time		CRF page 1					No	Record Qualifier	ROLECODE	
DM	6	RFENDTC	date	16	Subject Reference End Date/Time		Derived					No	Record Qualifier	ROLECODE	
DM	7	SITEID	text	7	Study Site Identifier		CRF page 1					Yes	Record Qualifier	ROLECODE	
DM	8	BRTHDTC	date	16	Date/Time of Birth		CRF page 1					No	Record Qualifier	ROLECODE	

DEFINE_HEADER_METADATA | TOC_METADATA | VARIABLE_METADATA | VALUELEVEL_METADATA | COMPUTATION_METHOD | CODELISTS

2. DM raw data

	A	B	C	D	E	F	G
1	101	0	1	3	BRAZILIAN	2/5/1974	4/2/2010
2	102	1	2	1		11/2/1946	2/13/2010
3	103	1	1	2		5/1/1979	5/16/2010
4	104	0	2	1		5/1/1972	1/2/2010
5	105	1	1	3	ABORIGIN	3/2/1979	4/20/2010
6	106	0	2	1		5/1/1977	4/1/2010
7	201	1	1	3	LIBYAN	4/28/1949	6/11/2010
8	202	0	2	1		1/13/1967	2/23/2010
9	203	1	1	2		1/1/1971	6/10/2010
10	204	0	2	1		4/17/1950	2/3/2010
11	205	1	1	3	HMONG	5/13/1978	4/13/2010
12	206	1	2	1		2/9/1948	7/1/2010
13	301	0	1	1		4/12/1941	2/20/2010
14	302	0	1	2		7/2/1978	5/12/2010
15	303	1	1	1		3/2/1967	2/19/2010
16	304	0	1	1		3/3/1958	5/19/2010
17	305	1	1	1		2/28/1966	6/10/2010
18	306	0	1	2		1/2/1960	5/23/2010
19	401	1	2	1		10/31/1970	6/13/2010
20	402	0	2	2		10/12/1980	1/2/2010

3. Dosing raw data

	A	B	C	D
1	subject	startdt	enddt	dailydose
2	101	4/2/2010	7/26/2010	2
3	101	7/31/2010	10/10/2010	3
4	102	2/13/2010	3/20/2010	2
5	102	3/25/2010	8/10/2010	1
6	103	5/16/2010	11/14/2010	1
7	104	1/2/2010	1/10/2010	2
8	104	1/15/2010	5/25/2010	1
9	104	5/26/2010	7/4/2010	2
10	105	4/20/2010	7/20/2010	2
11	105	7/21/2010	10/19/2010	1
12	106	4/1/2010	10/10/2010	2
13	201	6/11/2010	12/11/2010	1
14	202	2/23/2010	5/19/2010	2
15	203	6/10/2010	6/20/2010	1
16	204	2/3/2010	5/4/2010	2
17	204	5/5/2010	8/3/2010	2
18	205	4/13/2010	10/10/2010	1
19	206	7/1/2010	10/1/2010	2
20	206	10/2/2010	12/27/2010	2

Programs

Program 1: make_empty_dataset.sas

```
/*make_empty_dataset.sas creates a zero record dataset based on a dataset metadata
spreadsheet.  The dataset created is called EMPTY_ where "" is the name of the
dataset.  This macro also creates a global macro variable called KEEPSTRING that
holds the dataset variables desired and listed in the order they should appear.
[The variable order is dictated by VARNUM in the metadata spreadsheet.]

MACRO PARAMETERS:
metadatafile = the MS Excel file containing the dataset metadata
dataset = the dataset or domain name you want to extract*/

%macro make_empty_dataset(metadatafile=,dataset=);

   proc import
      datafile="&metadatafile"
      out=_temp
      dbms=xlsx
      replace;
      sheet="VARIABLE_METADATA";
   run;

   ** sort the dataset by expected specified variable order;
   proc sort
     data=_temp;
               where domain = "&dataset";
      by varnum;
   run;

   ** create keepstring macro variable and load metadata
   ** information into macro variables;
   %global &dataset.KEEPSTRING;
   data _null_;
     set _temp nobs=nobs end=eof;

      if _n_=1 then
        call symput("vars", compress(put(nobs,3.)));

      call symputx('var'    || compress(put(_n_, 3.)), variable);
      call symputx('label'  || compress(put(_n_, 3.)), label);
      call symputx('length' || compress(put(_n_, 3.)), put(length, 3.));

      ** valid ODM types include TEXT, INTEGER, FLOAT, DATETIME,
      ** DATE, TIME and map to SAS numeric or character;
      if upcase(type) in ("INTEGER", "FLOAT") then
        call symputx('type' || compress(put(_n_, 3.)), "");
      else if upcase(type) in ("TEXT", "DATE", "DATETIME",
                          "DATE", "TIME") then
        call symputx('type' || compress(put(_n_, 3.)), "$");
      else
        put "ERR" "OR: not using a valid ODM type.  " type=;

      ** create **KEEPSTRING macro variable;
      length keepstring $ 32767;
```

```
    retain keepstring;
    keepstring = compress(keepstring) || "|" || left(variable);
    if eof then
      call symputx(upcase(compress("&dataset" || 'KEEPSTRING')),
                left(trim(translate(keepstring," ","|"))));
  run;

  ** create a 0-observation template data set used for assigning
  ** variable attributes to the actual data sets;
  data EMPTY_&dataset;
    %do i=1 %to &vars;
      attrib &&var&i label="&&label&i" length=&&type&i.&&length&i...;
      %if &&type&i=$ %then
        retain &&var&i '';
      %else
        retain &&var&i .;
        ;
    %end;
    if 0;
  run;

%mend make_empty_dataset;
```

Program 2: dm.sas

```
/*Process the raw demographic data from a CSV file*/

%include '/folders/myfolders/test1/common.sas';
%common;

filename infl '/folders/myfolders/test1/dm.csv';

proc format;
  value trt
    1 = "Active"
    0 = "Placebo";
  value gender
    1 = "M"
    2 = "F"
    3 = "U";
  value race
    1 = "White"
    2 = "Black"
    3 = "Other";
run;

data source.demographic;
infile infl dlm='2C0D'x dsd missover;
length dob1 $10
     randdt1 $10;
input subject trt gender race orace $ dob1 $ randdt1 $;
dob=input(dob1,mmddyy10.);
randdt=input(randdt1,mmddyy10.);
format dob randdt mmddyy10.;
uniqueid = 'UNI' || put(subject,3.);
gender1=put(gender,gender.);
```

```
race1=put(race,race.);
trt1=put(trt,trt.);
label subject  = "Subject Number"
      trt      = "Treatment"
      gender   = "Gender"
      race     = "Race"
      orace    = "Oher Race Specify"
      dob      = "Date of Birth"
      uniqueid = "Company Wide Subject ID"
      randdt   = "Randomization Date";
drop dob1 randdt1 gender race;
rename gender1=gender race1=race;
run;
```

Program 3: ds.sas

```
/*Process the raw dose data from an Excel file.
Key points: 1) make up missing dates; 2) convert Excel dates to SAS dates*/

%include '/folders/myfolders/test1/common.sas';
%common;

libname ds xlsx "/folders/myfolders/test1/ds.xlsx";

data source.dosing;
set ds.Sheet1;
if find(startdt,'/') then
do;
  array var1(3) $4.;
  do i=1 to 3;
    var1(i)=scan(startdt,i,"/");
    if var1(i)=' ' then
    var1(i)='1';
  end;
  startdt1=mdy(input(var1(1),$4.), input(var1(2),$4.) , input(var1(3),$4.));
end;
else
  startdt1=input(startdt,$10.)-21916;
if find(enddt,'/') then
do;
  array var2(3) $4.;
  do i=1 to 3;
    var2(i)=scan(enddt,i,"/");
    if var2(i)=' ' then
    var2(i)='1';
  end;
  enddt1=mdy(input(var2(1),$4.), input(var2(2),$4.) , input(var2(3),$4.));
end;
else
  enddt1=input(enddt,$10.)-21916;
uniqueid = 'UNI' || put(subject,3.);
format startdt1 enddt1 mmddyy10.;
drop i startdt enddt var11-var13 var21-var23;
rename startdt1=startdt enddt1=enddt;
run;
```

Program 4: make_sort_order.sas

```
/* make_sort_order.sas creates a global macro variable called  SORTSTRING where ** is the name of the dataset that
contains the metadata specified sort order for a given dataset.

MACRO PARAMETERS:
metadatafile = the file containing the dataset metadata
dataset = the dataset or domain name*/

%macro make_sort_order(metadatafile=,dataset=);

   proc import
      datafile="&metadatafile"
      out=_temp
      dbms=xlsx
      replace;
      sheet="TOC_METADATA";
   run;

   ** create **SORTSTRING macro variable;
   %global &dataset.SORTSTRING;
   data _null_;
     set _temp;

      where name = "&dataset";

      call symputx(compress("&dataset" || "SORTSTRING"),
                translate(domainkeys," ",","));
   run;

%mend make_sort_order;
```

Program 5: STDM_DM.sas

```
/* STDM_DM.sas creates the SDTM DM and SUPPDM datasets and saves them as permanent SAS datasets to the target
libref */

%include '/folders/myfolders/test1/common.sas';
%common;

**** CREATE EMPTY DM DATASET CALLED EMPTY_DM;
%include '/folders/myfolders/test1/make_empty_dataset.sas';
%make_empty_dataset(metadatafile=/folders/myfolders/test1/SDTM_METADATA.xlsx,dataset=DM);


**** GET FIRST AND LAST DOSE DATE FOR RFSTDTC AND RFENDTC;
proc sort
  data=source.dosing(keep=subject startdt enddt)
  out=dosing;
    by subject startdt;
run;

**** FIRSTDOSE=FIRST DOSING AND LASTDOSE=LAST DOSING;
data dosing;
  set dosing;
    by subject;
```

```
   format firstdose lastdose mmddyy10.;
   retain firstdose lastdose;

   if first.subject then
     do;
       firstdose = .;
       lastdose = .;
     end;

   firstdose = min(firstdose,startdt,enddt);
   lastdose = max(lastdose,startdt,enddt);
   drop startdt enddt;
   if last.subject;
run;

**** GET DEMOGRAPHICS DATA;
proc sort
  data=source.demographic
  out=demographic;
    by subject;
run;

data demog_dose;
  merge demographic
        dosing;
    by subject;
run;

**** DERIVE THE MAJORITY OF SDTM DM VARIABLES;
data dm;
  set EMPTY_DM
    demog_dose;
    studyid = 'XYZ123';
    domain = 'DM';
    usubjid = left(uniqueid);
    subjid = put(subject,3.);
    rfstdtc = put(firstdose,yymmdd10.);
    rfendtc = put(lastdose,yymmdd10.);
    siteid = substr(subjid,1,1) || "00";
    brthdtc = put(dob,yymmdd10.);
    age = floor ((intck('month',dob,firstdose) -
         (day(firstdose) < day(dob))) / 12);
    if age ne . then
       ageu = 'YEARS';
    country = "USA";
    sex=gender;
    arm=trt1;
    armcd=put(trt,3.);
    drop gender trt trt1;
run;

%include '/folders/myfolders/test1/make_sort_order.sas';
%make_sort_order(metadatafile=/folders/myfolders/test1/SDTM_METADATA.xlsx,dataset=DM);

proc sort
  data=dm(keep = &DMKEEPSTRING)
```

```
  out=target.dm;
    by &DMSORTSTRING;
run;

**** CREATE EMPTY SUPPDM DATASET CALLED EMPTY_DM;
%make_empty_dataset(metadatafile=/folders/myfolders/test1/SDTM_METADATA.xlsx,dataset=SUPPDM);

data suppdm;
  set EMPTY_SUPPDM
      dm;

    keep &SUPPDMKEEPSTRING;

    **** OUTPUT OTHER RACE AS A SUPPDM VALUE;
    if orace ne '' then
      do;
        rdomain = 'DM';
        qnam = 'RACEOTH';
        qlabel = 'Race, Other';
        qval = left(orace);
        qorig = 'CRF';
        output;
      end;

    **** OUTPUT RANDOMIZATION DATE AS SUPPDM VALUE;
    if randdt ne . then
      do;
        rdomain = 'DM';
        qnam = 'RANDDTC';
        qlabel = 'Randomization Date';
        qval = left(put(randdt,yymmdd10.));
        qorig = 'CRF';
        output;
      end;
run;

%make_sort_order(metadatafile=/folders/myfolders/test1/SDTM_METADATA.xlsx,dataset=SUPPDM);

proc sort
  data=suppdm
  out=target.suppdm;
    by &SUPPDMSORTSTRING;
run;
```

Program 6: STDM_EX.sas

```
/* STDM_EX.sas creates the SDTM EX dataset and saves it as a permanent SAS dataset to the target libref */

%include '/folders/myfolders/test1/common.sas';
%common;

**** CREATE EMPTY DM DATASET CALLED EMPTY_DM;
%include '/folders/myfolders/test1/make_empty_dataset.sas';
```

```
%make_empty_dataset(metadatafile=/folders/myfolders/test1/SDTM_METADATA.xlsx,datase
t=EX);
%include '/folders/myfolders/test1/make_sdtm_dy2.sas';
%include '/folders/myfolders/test1/make_sort_order.sas';

**** DERIVE THE MAJORITY OF SDTM EX VARIABLES;
data ex;
  set EMPTY_EX
      source.dosing;

    studyid = 'XYZ123';
    domain = 'EX';
    usubjid = left(uniqueid);
    exdose = dailydose;
    exdostot = dailydose;
    exdosu = 'mg';
    exdosfrm = 'TABLET, COATED';
    exstdtc=put(startdt,yymmdd10.);
    exendtc=put(enddt,yymmdd10.);
run;

proc sort
  data=ex;
    by usubjid;
run;

**** CREATE SDTM STUDYDAY VARIABLES AND INSERT EXTRT;
data ex;
  merge ex(in=inex) target.dm(keep=usubjid rfstdtc arm);
    by usubjid;

    if inex;

    %make_sdtm_dy(refdate=rfstdtc,date=exstdtc);
    %make_sdtm_dy(refdate=rfstdtc,date=exendtc);

    **** in this simplistic case all subjects received the treatment they were
randomized to;
    extrt = arm;
run;


**** CREATE SEQ VARIABLE;
proc sort
  data=ex;
    by studyid usubjid extrt exstdtc;
run;

OPTIONS MISSING = ' ';
data ex;
  retain STUDYID DOMAIN USUBJID EXSEQ EXTRT EXDOSE EXDOSU EXDOSFRM EXDOSTOT
         EXSTDTC EXENDTC EXSTDY EXENDY;
  set ex(drop=exseq);
    by studyid usubjid extrt exstdtc;

    if not (first.exstdtc and last.exstdtc) then
      put "WARN" "ING: key variables do not define an unique record. " usubjid=;
```

```
    retain exseq;
    if first.usubjid then
      exseq = 1;
    else
      exseq = exseq + 1;

    label exseq = "Sequence Number";
run;


**** SORT EX ACCORDING TO METADATA AND SAVE PERMANENT DATASET;
%make_sort_order(metadatafile=/folders/myfolders/test1/SDTM_METADATA.xlsx,dataset=E
X);


proc sort
  data=ex(keep = &EXKEEPSTRING)
  out=target.ex;
    by &EXSORTSTRING;
run;
```