Paper 195-2009

Mapping CDISC Metadata Attributes: Using Data _Null_ and Proc Datasets in SAS®.

Rita Tsang, Averion International Corp., Southborough, Massachusetts

ABSTRACT

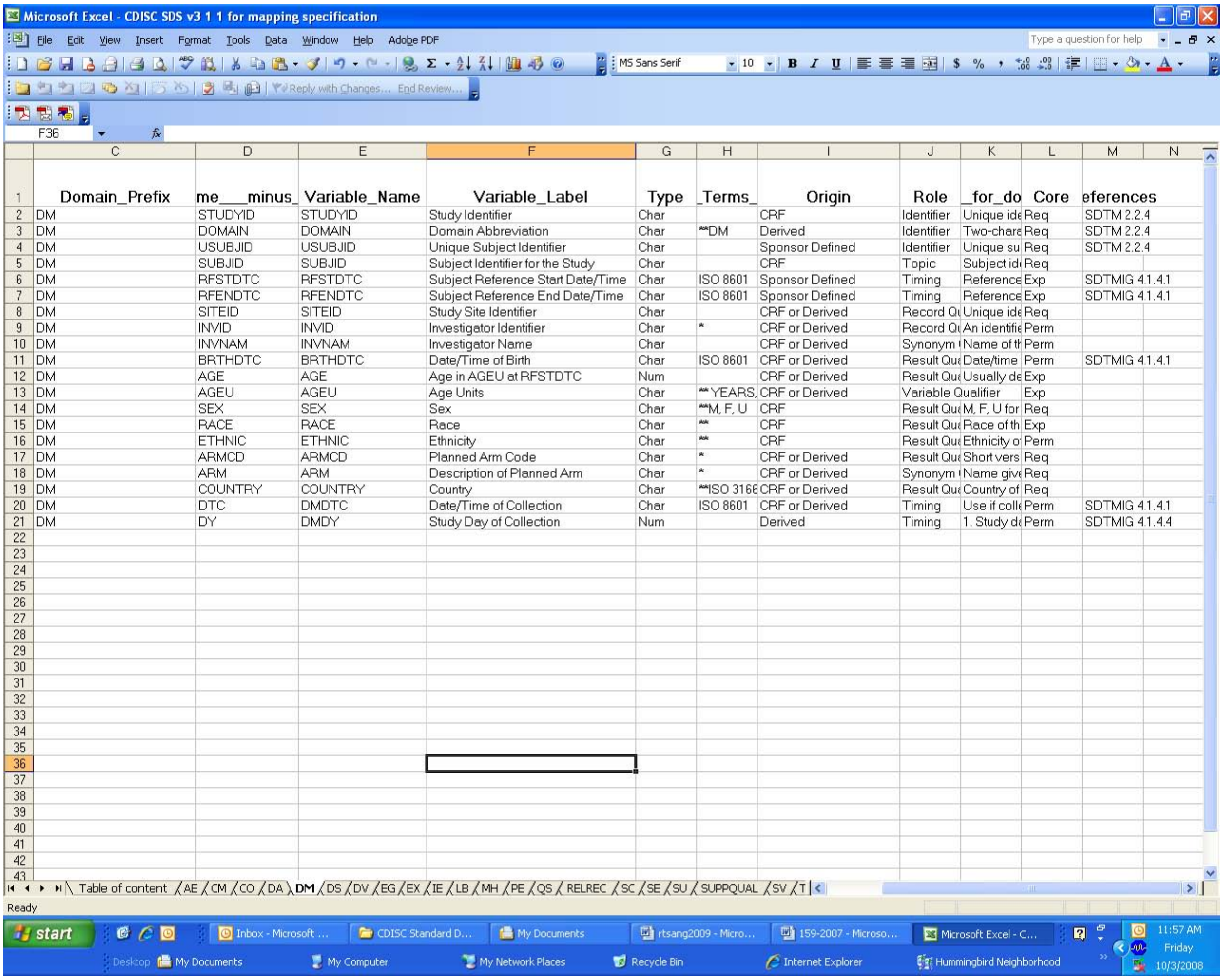
In the pharmaceutical environment, the CDISC Study Data Tabulation Model (SDTM) provides the framework for how clinical data should be submitted to the regulatory authority, such as the US Food and Drug Administration (FDA). Specific metadata attributes, such as variable type, length, control terminology, and variable label, are established to standardize the data format in the industry. The process of mapping these attributes can be tedious and time-consuming. This paper will walk you through how it can be made more automated by using Data _Null_ and Proc Datasets in a step-by-step approach. This automated process can help save time in manual programming and to ensure the accuracy of the updates.

INTRODUCTION

The task here is to map our case report form (CRF) data to CDISC-standard SAS data sets based on the mapping specification in Microsoft Excel format. Individual domain programs will be set up for the mapping purpose. Derived variables can be added in these individual domain programs. Data _Null_ will be used to create SAS programs for applying the metadata attributes to the CDISC domains using Proc Datasets.

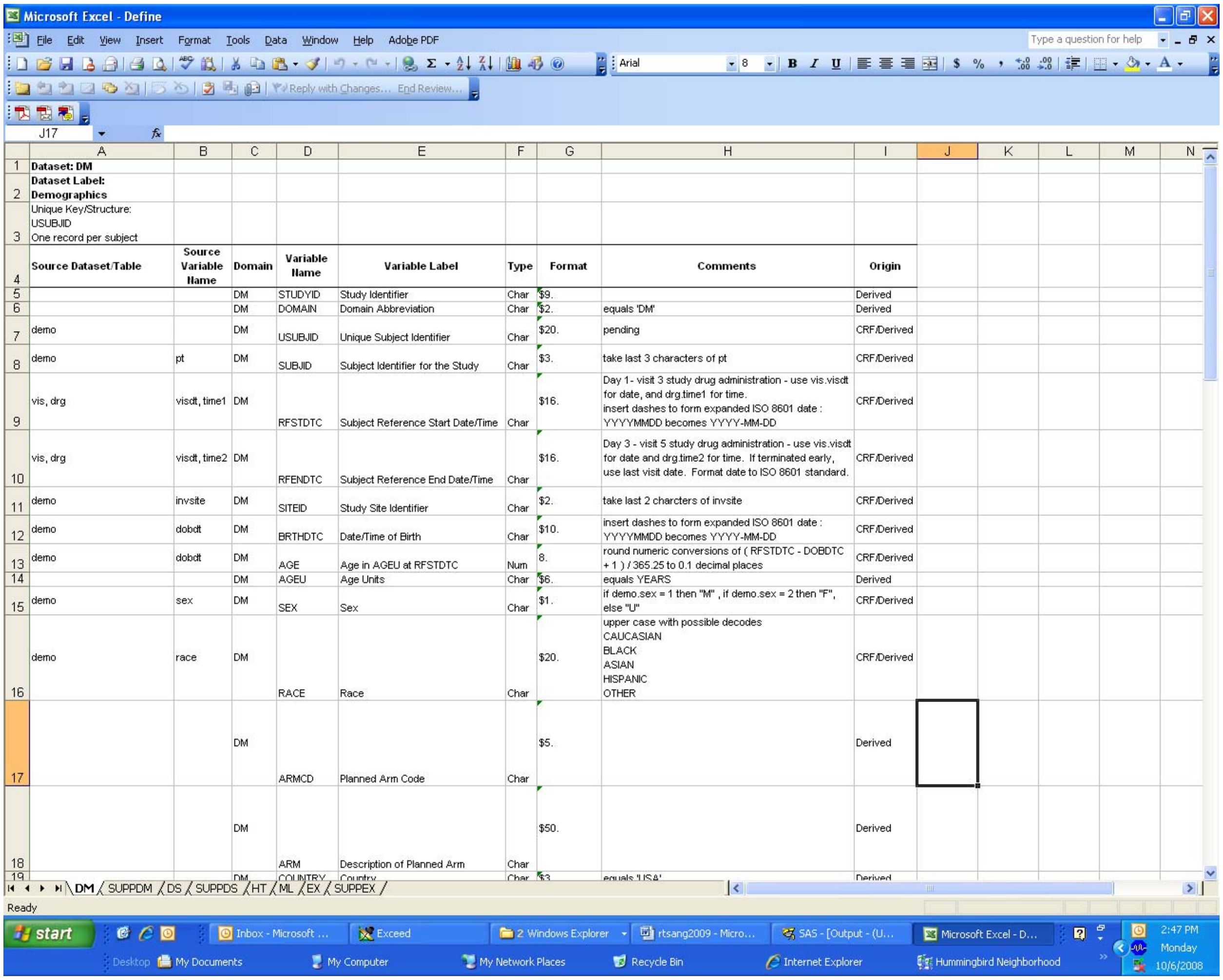
Step 1 – From the Mapping Specification to SAS data:

We can read in the Microsoft Excel mapping specification into a SAS data set (define.sas7bdat) using Proc Import or DDE. Here is an example of the CDISC standard mapping specification for the DM domain from CDISC.ORG:



	C	D	E	F	G	H	I	J	K	L	M	N
1	Domain_Prefix	me___minus_	Variable_Name	Variable_Label	Type	_Terms_	Origin	Role	_for_do	Core	eferences	
2	DM	STUDYID	STUDYID	Study Identifier	Char		CRF	Identifier	Unique ide	Req	SDTM 2.2.4	
3	DM	DOMAIN	DOMAIN	Domain Abbreviation	Char	**DM	Derived	Identifier	Two-chara	Req	SDTM 2.2.4	
4	DM	USUBJID	USUBJID	Unique Subject Identifier	Char		Sponsor Defined	Identifier	Unique su	Req	SDTM 2.2.4	
5	DM	SUBJID	SUBJID	Subject Identifier for the Study	Char		CRF	Topic	Subject id	Req		
6	DM	RFSTDTC	RFSTDTC	Subject Reference Start Date/Time	Char	ISO 8601	Sponsor Defined	Timing	Reference	Exp	SDTMIG 4.1.4.1	
7	DM	RFENDTC	RFENDTC	Subject Reference End Date/Time	Char	ISO 8601	Sponsor Defined	Timing	Reference	Exp	SDTMIG 4.1.4.1	
8	DM	SITEID	SITEID	Study Site Identifier	Char		CRF or Derived	Record Qu	Unique ide	Req		
9	DM	INVID	INVID	Investigator Identifier	Char	*	CRF or Derived	Record Qu	An identifie	Perm		
10	DM	INVNAM	INVNAM	Investigator Name	Char		CRF or Derived	Synonym	Name of th	Perm		
11	DM	BRTHDTC	BRTHDTC	Date/Time of Birth	Char	ISO 8601	CRF or Derived	Result Qua	Date/time	Perm	SDTMIG 4.1.4.1	
12	DM	AGE	AGE	Age in AGEU at RFSTDTC	Num		CRF or Derived	Result Qua	Usually de	Exp		
13	DM	AGEU	AGEU	Age Units	Char	** YEARS,	CRF or Derived	Variable Qualifier		Exp		
14	DM	SEX	SEX	Sex	Char	**M, F, U	CRF	Result Qua	M, F, U for	Req		
15	DM	RACE	RACE	Race	Char	**	CRF	Result Qua	Race of th	Exp		
16	DM	ETHNIC	ETHNIC	Ethnicity	Char	**	CRF	Result Qua	Ethnicity o	Perm		
17	DM	ARMCD	ARMCD	Planned Arm Code	Char	*	CRF or Derived	Result Qua	Short vers	Req		
18	DM	ARM	ARM	Description of Planned Arm	Char	*	CRF or Derived	Synonym	Name give	Req		
19	DM	COUNTRY	COUNTRY	Country	Char	**ISO 3166	CRF or Derived	Result Qua	Country of	Req		
20	DM	DTC	DMDTC	Date/Time of Collection	Char	ISO 8601	CRF or Derived	Timing	Use if colle	Perm	SDTMIG 4.1.4.1	
21	DM	DY	DMDY	Study Day of Collection	Num		Derived	Timing	1. Study da	Perm	SDTMIG 4.1.4.4	

The template from CDISC.ORG may be adapted for your protocol and sponsor needs. The key is to see the flow of data from the CRF source data to the final CDISC data. Here is an example of the DM domain in the mapping specification.



Dataset: DM
Dataset Label: Demographics
Unique Key/Structure: USUBJID
One record per subject

Source Dataset/Table	Source Variable Name	Domain	Variable Name	Variable Label	Type	Format	Comments	Origin
		DM	STUDYID	Study Identifier	Char	$9.		Derived
		DM	DOMAIN	Domain Abbreviation	Char	$2.	equals 'DM'	Derived
demo		DM	USUBJID	Unique Subject Identifier	Char	$20.	pending	CRF/Derived
demo	pt	DM	SUBJID	Subject Identifier for the Study	Char	$3.	take last 3 characters of pt	CRF/Derived
vis, drg	visdt, time1	DM	RFSTDTC	Subject Reference Start Date/Time	Char	$16.	Day 1- visit 3 study drug administration - use vis.visdt for date, and drg.time1 for time. insert dashes to form expanded ISO 8601 date : YYYYMMDD becomes YYYY-MM-DD	CRF/Derived
vis, drg	visdt, time2	DM	RFENDTC	Subject Reference End Date/Time	Char	$16.	Day 3 - visit 5 study drug administration - use vis.visdt for date and drg.time2 for time. If terminated early, use last visit date. Format date to ISO 8601 standard.	CRF/Derived
demo	invsite	DM	SITEID	Study Site Identifier	Char	$2.	take last 2 charcters of invsite	CRF/Derived
demo	dobdt	DM	BRTHDTC	Date/Time of Birth	Char	$10.	insert dashes to form expanded ISO 8601 date : YYYYMMDD becomes YYYY-MM-DD	CRF/Derived
demo	dobdt	DM	AGE	Age in AGEU at RFSTDTC	Num	8.	round numeric conversions of (RFSTDTC - DOBDTC + 1) / 365.25 to 0.1 decimal places	CRF/Derived
		DM	AGEU	Age Units	Char	$6.	equals YEARS	Derived
demo	sex	DM	SEX	Sex	Char	$1.	if demo.sex = 1 then "M" , if demo.sex = 2 then "F", else "U"	CRF/Derived
demo	race	DM	RACE	Race	Char	$20.	upper case with possible decodes CAUCASIAN BLACK ASIAN HISPANIC OTHER	CRF/Derived
		DM	ARMCD	Planned Arm Code	Char	$5.		Derived
		DM	ARM	Description of Planned Arm	Char	$50.		Derived
		DM	COUNTRY	Country	Char	$3	equals 'USA'	Derived

DM / SUPPDM / DS / SUPPDS / HT / ML / EX / SUPPEX

The following SAS code reads in the Excel mapping specification into a SAS data set (DEFINE.SAS7BDAT) using PROC IMPORT.

```
******************************************************************************************************************************
Proc SQL is used to create macro variables for the CDISC domains (&_dslist) and the number of domains (&_nds) in
the RAW library. There are two domains in the RAW library – DM and SUPPDM. The macro variable _DSLIST
resolves to DM SUPPDM, and the macro variable _NDS resolves to 2.
******************************************************************************************************************************;

%let _dslist =;

proc sql noprint;
 select trim(left(memname)) into :_dslist separated by ' '
 from dictionary.tables
 where libname = 'RAW'
 order by memname;
quit;
```

```
%let _nds = &sqlobs;
%let _dslist = %upcase(&_dslist);
%put &_dslist;

******************************************************************************************************************************************
The following macro DEFINE uses a do-loop to read in the attributes of each CDISC domain from the individual Excel
worksheet in the Excel workbook (define.xls).  The attributes that are kept in the SAS data DEFINE.SAS7BDAT are
domain name (DOMAIN), variable name (VNAME), variable label (VLABEL), variable type (VTYPE), and variable
format (VFORMAT).
******************************************************************************************************************************************;

%macro define;


%do i = 1 %to &_nds;


  %let _cds = %scan(&_dslist,&i,%str( ));


      proc import out= work.&_cds (keep=f3-f7 rename=(f3=domain f4=vname
       f5=vlabel f6=vtype f7=vformat) where=(domain>' ' and upcase(domain) ne
       'DOMAIN')) datafile= "C:\cdisc\define.xls" dbms=EXCEL replace;
        sheet="&_cds$";
      run;


      %if &i=1 %then %do;
        data cdisc.define;
         set &_cds;
        run;
      %end;
      %else %do;
        data cdisc.define;
         set cdisc.define &_cds;
        run;
      %end;

  %end;
%mend define;
%define;
```

This is an example of the contents of DEFINE.SAS7BDAT. This SAS data set will contain attributes of the metadata.

Obs	domain	vname	vlabel	vtype	vformat
1	DM	STUDYID	Study Identifier	Char	$9.
2	DM	DOMAIN	Domain Abbreviation	Char	$2.
3	DM	USUBJID	Unique Subject Identifier	Char	$20.
4	DM	SUBJID	Subject Identifier for the Study	Char	$3.
5	DM	RFSTDTC	Subject Reference Start Date/Time	Char	$16.
6	DM	RFENDTC	Subject Reference End Date/Time	Char	$16.
7	DM	SITEID	Study Site Identifier	Char	$2.
8	DM	BRTHDTC	Date/Time of Birth	Char	$10.
9	DM	AGE	Age in AGEU at RFSTDTC	Num	8.
10	DM	AGEU	Age Units	Char	$6.
11	DM	SEX	Sex	Char	$1.
12	DM	RACE	Race	Char	$20.
13	DM	ARMCD	Planned Arm Code	Char	$5.
14	DM	ARM	Description of Planned Arm	Char	$50.
15	DM	COUNTRY	Country	Char	$3.
16	DM	DMDTC	Date/Time of Collection	Char	$16.
17	SUPPDM	STUDYID	Study Identifier	Char	$9.
18	SUPPDM	RDOMAIN	Related Domain Abbreviation	Char	$2.
19	SUPPDM	USUBJID	Unique Subject Identifier	Char	$20.
20	SUPPDM	IDVAR	Identifying Variable	Char	$8.
21	SUPPDM	IDVARVAL	Identifying Variable Value	Char	$200.
22	SUPPDM	QNAM	Variable Name	Char	$8.

Obs	domain	vname	vlabel	vtype	vformat
23	SUPPDM	QLABEL	Variable Label	Char	$40.
24	SUPPDM	QVAL	Data Value	Char	$200.
25	SUPPDM	QORIGIN	Origin	char	$40.
26	SUPPDM	QEVAL	Evaluator	Char	$40.

Step 2 – Using Data _Null and the PUT statement to create SAS code:

With the Define SAS data set now created, we can use Data _Null_ and the PUT statement to create SAS code to apply the metadata attributes using Proc Datasets. The following code will create a program called LABEL.SAS for labeling variables based on the information in the Define data set (DEFINE.SAS7BDAT):

```
proc sort data=cdisc.define out=label;
 by domain;
run;
```

****
Data _null_ is just a simple SAS statement that asks SAS not to create a data set when executing the DATA step, since our main interest here is really to create a SAS program. The FILE statement when used in conjunction with the PUT statement, tells SAS to write lines of text to an external location, a SAS program in this case.
**;

```
data _null_;
 set label end=eof;
  by domain;
  file "C:\cdisc\label.sas";
```

****
By using the PUT statement, we write the Proc Datasets syntax at the first few lines of the program. Note that there are line pointer controls (/) in some of the PUT statements. Each line pointer control instructs SAS to advance the pointer to column 1 of the next line. As a result, blank lines can be inserted into the program.
**;

```
if (_n_ = 1) then do;

   put "proc datasets memtype=data;" ;
   put "  copy in=raw out=cdisc;" ;
   put "run;" //;
   put "proc datasets library=cdisc memtype=data;" /;
end;
```

****
In the following example, we are combining both the character constant (e.g. "   modify ") and a variable (e.g. DOMAIN), and followed by another character constant (";")in the PUT statement.   When a variable (e.g. DOMAIN, VNAME, VLABEL) is being used as an argument of the PUT statement, the value of the variable will be written in the file.

Note that by using the format $8. after VNAME, the output style is formatted. The value of the variable VNAME will have a width of 8 characters in the SAS program.

Also note that the +(-1) is a pointer control that moves the pointer backward to remove the unwanted blank space that occurs between the value of VLABEL and the double-quotes ('"').
***;

```
            if (first.domain) then do;
                  put "   modify " domain ";" ;
                  put "   label " vname $8. ' = "' vlabel +(-1) '"';
            end;
            else  put "         " vname $8. ' = "' vlabel +(-1) '"';

            if (last.domain) then put "        ;" /;


         if eof then do;
             put "run;";
             put "quit;";
         end;
        run;
```

Similarly, Data _Null_ and the PUT statement can also be used to generate program code to format variables in the metadata.

Step 3 – Running Proc Datasets to apply the CDISC metadata attributes

The program LABEL.SAS generated by Data _Null_ and the PUT statement in Step 2 is shown below. Proc Datasets is a versatile procedure in SAS. It can be used for copying datasets from library to library, renaming and deleting data sets within a data library, as well as modifying the attributes (such as labels, formats, informats) in a data library.

```
proc datasets memtype=data;
  copy in=raw out=cdisc;
run;


proc datasets library=cdisc memtype=data;

   modify DM ;
   label STUDYID  = "Study Identifier"
         DOMAIN   = "Domain Abbreviation"
         USUBJID  = "Unique Subject Identifier"
         SUBJID   = "Subject Identifier for the Study"
         RFSTDTC  = "Subject Reference Start Date/Time"
         RFENDTC  = "Subject Reference End Date/Time"
         SITEID   = "Study Site Identifier"
         BRTHDTC  = "Date/Time of Birth"
         AGE      = "Age in AGEU at RFSTDTC"
         AGEU     = "Age Units"
         SEX      = "Sex"
         RACE     = "Race"
         ARMCD    = "Planned Arm Code"
         ARM      = "Description of Planned Arm"
         COUNTRY  = "Country"
         DMDTC    = "Date/Time of Collection"
         ;

   modify SUPPDM ;
   label STUDYID  = "Study Identifier"
         RDOMAIN  = "Related Domain Abbreviation"
         USUBJID  = "Unique Subject Identifier"
         IDVAR    = "Identifying Variable"
         IDVARVAL = "Identifying Variable Value"
         QNAM     = "Variable Name"
         QLABEL   = "Variable Label"
         QVAL     = "Data Value"
         QORIGIN  = "Origin"
         QEVAL    = "Evaluator"
         ;

run;
quit;
```

CONCLUSION:

In the process of CDISC mapping, the CDISC mapping specification document is a living document that may be updated based on project team discussion. This paper has shown you an example of the automated process that can help save programming time and avoid manual errors. It can also help accommodate for numerous updates in the mapping specification. More importantly, the consistency between the mapping specification and the final CDISC domains can be more assured.

REFERENCES

Clinical Data Interchange Standards Consortium (CDISC) (2005), Study Data Tabulation Model Implementation Guide: Human Clinical Trials, Austin, TX: CDISC Inc.

SAS Institute (2007), SAS Online Documentation for SAS 9.1.3 release, Cary, NC: SAS Institute Inc.

ACKNOWLEDGEMENT

The author would like to express her appreciation to the following individual for her invaluable comments and suggestions in this paper:

Shannon Escalante, Ikaria

CONTACT INFORMATION

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

Rita Tsang
Averion International Corp.
225 Turnpike Road
Southborough, MA 01772
Rita.Tsang@averionintl.com