Paper 206-2008

CDISC: Pains and Pitfalls to Dataset Creation

Janet Stuelpner, Left Hand Computing, Inc., New Canaan, CT
Steven Michaelson, SimulStat Inc., San Diego, CA

ABSTRACT

Standardization is a great thing. Although it causes great pain at the beginning, the rewards are well worth the effort in the end. That being said, creating datasets in the CDISC SDTM format is difficult and leads to a great deal of interpretation by the sponsor. There are areas that are very specific and others that are vague and general. This paper will try to de-mystify the process and show you how to create files in CDISC format that satisfy the data management and regulatory staff at your company.

INTRODUCTION

At first glance, one would think that it is easy to create a dataset that conforms to the CDISC standards. The number of domains is limited, the number of variables is limited and everything is specified in terms that are easy to understand. Or is it? Each protocol has its own idiosyncrasies. In other words, they are all different. For each study, the data can be different, the variables can be defined differently and the information that is needed for safety can vary widely. Now the trick is to take the information that is collected and turn it into a format that is standard. Is this an easy task? This paper will try to point out the pitfalls and sometimes the pain that is encountered when making the transformation of data to the CDISC format.

OBSERVATIONS AND VARIABLES

Let's begin with the variables. The CDISC SDTM Guidelines have a metadata dictionary for each domain. In this dictionary the attributes of the variables are defined and examples given where necessary. Below is the format of the items included in the metadata dictionary for each domain.

1. Variable Name
2. Variable Label
3. Type – character or numeric
4. Controlled Terms or Format – presentation format for the variable
5. Origin – CRF, Derived or Sponsor Defined
6. Role – a classification of the type of information that is contained in that variable and how that variable can be used.
   a. Identifier: identify the study, subject, domain and sequence number
   b. Topic: focus of the observation
   c. Timing: timing such as start or stop dates
   d. Qualifier: describe the results or additional traits
   e. Rule: algorithm or executable method to define start, stop or looping conditions
7. CDISC Notes – comments or other relevant information about the variable
8. Core – how variable is classified according to CDISC as a measure of compliance and to provide general guidance.
   a. Required: needed to distinguish observations, they are essential and required by regulatory authorities
   b. Expected: necessary to make the record useful to the domain and are assumed to be present in each domain, even if they are specified as null.
   c. Permissible: appropriate when collected or derived
9. References – section of guidelines for reference

Dates are required to be in character format. The time variables (which may not always be collected depending on the type of study) are also in character format. To address the FDA's request to provide a uniform date/time representation, all date and time variables in CDISC Version 3.0 were replaced with a single date/time variable ending in 'DTM'. These date and time values were stored as SAS dates and times. As you know, SAS dates and times are stored as a number (the number of seconds since midnight on January 1, 1960). In order to see a meaningful date and time, these numbers had to be formatted. This was changed in CDISC Version 3.1. Now all date variable names end in ‘DTC’ to indicate that they are a character field. Some examples are: RFSTDTC (reference start date/time), BRTHDTC (birth date/time), CMENDTC (End date/time of medication). According to the SDTM guideline the format of the date is stated as, “Dates and time of day are now stored according to the international standard ISO 8601.” (Reference URL: http://www.iso.ch/iso/en/ISOOnline.openerpage) This is a hierarchy example of available date/time components and how they are to be presented in a CDISC format.

Complete date and time	2001-10-11T11:12:15
Unknown seconds	2001-10-11T11:12
Unknown minutes	2001-10-11T11
Unknown time	2001-10-11
Unknown day	2001-10
Unknown month	2001

DATASETS AND DOMAINS

The domains determine the number and kind of datasets that are contained in a submission. Domains are defined as data that have some topic-specific commonality about a subject. It is possible that data that refers to the same topic can be spread among several datasets. All information about demographics will be found in the DM domain while all information about the concomitant medications will be found in the CM domain. Each domain is represented by a separate dataset. The standard domains that are defined are as follows:

1. Special purpose domains
    a. DM Demographics
    b. CO Comments
2. Interventions
    a. CM Concomitant Medications
    b. EX Exposure
    c. SU Substance Use
3. Events
    a. AE Adverse Events
    b. DS Disposition
    c. MH Medical History
4. Findings
    a. EG EKG test results
    b. IE Inclusion/Exclusion Exceptions
    c. LB Laboratory results
    d. PE Physical Exam
    e. QS Questionnaires
    f. SC Subject Characteristics
    g. VS Vital Signs

As you can see from the list above, there are specific naming conventions for these datasets and there are also specific conventions for the variables that are contained in them. In some cases, this actually makes it easier to program. In some cases it makes it much more difficult. Each domain is specified by a two character code. This code is also used to name most of the variables that are found in that domain. For example, the two character code of AE is used for the variables in the AE domain. Examples of them are: AETERM (reported term for the adverse event), AEDECOD (dictionary derived term), AESER (serious event), AEACN (action taken with study treatment).

Each row in a dataset (or table) represents one single observation. Depending on the specific dataset, this can include the data for one record for each patient or many records for each patient. The number of records is dependent on the type of data that it is and is driven by the SDTM Guidelines. As specified in the SDTM definition for ‘Origin’, data that is stored in the domain datasets can be raw data that is taken directly from the Case Report Form (CRF) or it can be derived. As mentioned above, the metadata dictionary contains the algorithms used to create the derived variables. There are some variables that must be converted to standard units, such as laboratory data. In some cases, there is a discrete set of values for a variable. For example, the variable that specifies whether an adverse event is serious or not can only have a Y or N as its value. There are other variables that have suggestions for the value like AESEV is suggested to be Mild, Moderate or Severe. These are only suggestions because the data can be collected with several more categories of severity.

SUGGESTIONS

MACROS

Once you get started coding the programs, you will find that the task of doing the conversion is very repetitive. There are many ways in which this repetitive process can become easier. The creation of macro and format libraries allows you to write the code once and then pick and choose what you need to complete the program. For example, for each of the protocols, you will need a demography domain. All of the variables in the output dataset are the same. A common ATTRIB statement can be written, placed in a macro and accessed in a macro library. Then, when there is a global change, it can be made in the macro library and picked up by each program without rewriting that one section of code in each program. The next time that program is run, it will pick up the change. It saves a great deal of writing and rerunning of the programs. Here is an example of an ATTRIB macro that can be used for the DM domain.

```
%macro attr_dm;
Attrib STUDYID      format=$15.  label='Study Identifier'
       DOMAIN       format=$2.   label='Domain Abbreviation'
       USUBJID      format=$15.  label='Unique Subject Identifier'
       SUBJID       format=$15.  label='Subject Identifier for the Study'
       RFSTDTC      format=$40.  label='Subject Reference Start Date/Time'
       RFENDTC      format=$40.  label='Subject Reference End Date/Time'
       SITEID       format=$20.  label='Study Site Identifier'
       BRTHDTC      format=$20.  label='Date/Time of Birth'
       AGE          format=8.    label='Age in AGEU at RFSTDTC'
       AGEU         format=$10.  label='Age Units'
       SEX          format=$10.  label='Sex'
       RACE         format=$20.  label='Race'
       ETHNIC       format=$40.  label='Ethnicity'
       ARMCD        format=$8.   label='Planned Arm Code'
       ARM          format=$80.  label='Description of Planned Arm'
       COUNTRY      format=$40.  label='Country'
       DMDTC        format=$40.  label='Date/Time of Collection'
       DMDY         format=8.    label='Study Day of Collection'
       INVID        format=$40.  label='Investigator Identifier'
       INVNAM       format=$40.  label='Investigator Name'
;
%mend attr_dm;
```

DATE/TIME CODING

When writing code for dates and times, you will be very lucky if you have complete date and time values available in the raw data. In some studies, times are not collected. In some cases you will only have two variables, one for date and one for time. However, most often, you will have a variable for each component of the date and time: month, day, year, hours, minutes, seconds. In the raw data they may also be character or numeric. All date/time variable values must be converted to the ISO 8601 standard character format. Here is one example of the derived coding you could use to create your date/time variable.

Assumptions:
** date format is character and is stored like this 10/09/2002 **;
** time format is character and is stored like this 11:30:00 **;

```
data t1;
      set raw.demo(keep=aedt_dt aedt_tm);
      aestdtc= left(trim(compress(scan(ae_dt,3,'/')))) || "-"
      || left(trim(compress(scan(ae_dt,1,'/')))) || "-"
      || left(trim(compress(scan(ae_dt,2,'/')))) || "T"
      || ae_tm;
run;
proc print noobs data=t1;
      var ae_dt ae_tm aestdtc;
run;
```

OUTPUT:

```
AE_DT                       AE_TM                       AESTDTC
12/13/2004                  08:10:00                    2004-12-13T08:10:00
12/09/2004                  08:30:00                    2004-12-09T08:30:00
01/18/2005                  11:00:00                    2005-01-18T11:00:00
01/18/2005                  11:30:00                    2005-01-18T11:30:00
```

FORMATS

Although most of the variables in each domain are character variables, it is still possible to utilize format libraries when creating the datasets for submissions. If the data that is collected from the CRF is coded, the formats can be used to decode the data when creating the domain datasets. Some sample formats that might be of great use during the dataset creation process are below.

```
proc format;
      value gender  0=Unknown
                    1=Male
                    2=Female
                    ;
      value severe  1=Mild
                    2=Moderate
                    3=Severe
                    ;
      value action  1=’drug withdrawn’
                    2=’dose reduced’
                    3=’dose increased’
                    4=’dose not changed’
                    5=’unknown’
                    6=’not applicable’
                    ;
      value race    1=White
                    2=Black
                    3=Hispanic
                    4=Asian
                    5=Other
                    ;
run;
```

TRANSPORT FILES

Data that is sent to the FDA must be in version 5 transport files. One assumption that is made is that all of the variables must be no longer than 8 characters. Labels cannot be any longer than a length of 40. One other thing that we found out the hard way is that you can only place one dataset in one transport file per study. In other words, if there are 10 studies and you are submitting 15 domains for each study, you need to create 150 transport files. If you create one transport file per study, only one dataset will be able to be read for each study unless the transport file is brought back into SAS datasets. When you double click on a transport file, only the first file is visible. Therefore, you must create one file for each domain for each study. In this way, it is not necessary to know how the transport file was created and the reviewers can look at each one without any knowledge of SAS.

CONCLUSION

There is a great deal of interpretation that must be done to create a submission in compliance with the CDISC SDTM V3.1 model. There are guidelines, however, they are just that, guidelines to help us develop datasets that are in a standard format for the reviewers. Hopefully, we have given you some lessons learned and techniques with which to create your submission. There are some great resources on the CDISC website and many places where questions are entertained and answered. Take advantage of these resources so that your submission is the best that it can be and so that regulatory agencies will be pleased with your effort.

REFERENCES

CDISC Submissions Standards Team, *Study Data Tabulation Model Implementation Guide: Human Clinical Trials*, June 2004

ACKNOWLEDGMENTS

I would like to take this opportunity to acknowledge all of the help and support given to me by my husband, Bob Stuelpner. He diligently read and re-read this paper in an effort to correct the obvious errors and not so obvious errors and keep us on the right track. His criticisms were constructive and his support never ending.

AUTHOR CONTACT

Janet Stuelpner
Left Hand Computing, Inc.
326 Old Norwalk Road
New Canaan, CT 06840
(203) 966-7520 phone
(203) 966-8027 fax
janet@lefthandcomputing.com

Steven Michaelson
SimulStat, Inc.
4370 La Jolla Drive, Suite 400
San Diego, CA 92122
(760) 622-7009
(760) 758-9239
smichaelson@simulstat.com