A Macro Mapping Date and Time Variable to CDISC Date and Time Variable

Song Liu, Biogen Idec, San Diego, California
Brett Sellars, Biogen Idec, San Diego, California

ABSTRACT

The Clinical Data Interchange Standards Consortium (CDISC) has been developing industry standards to support the processing of clinical data. The Study Data Tabulation Model (SDTM), which divides the clinical data by three classes, Intervention, Events, and Findings, defines a standard structure for study data tabulation that are to be submitted as part of a product application to the Food and Drug Administration. Timing variables, which describe the timing of an observation, are specified in the format of ISO 8601 across all classes under SDTM.

This paper is to discuss how to convert date/time variables collected from a clinical trial to date/time variables as defined in CDISC SDTM Version 1.1 (V1.1).

INTRODUCTION

It has been a long time coming for pharmaceutical companies and contract research organizations to find industry wide standard data definitions. The STDM provides a general framework for describing the organization of data collected during human and animal studies. A collection of observations on a particular topic is considered a domain. Typically, each domain is represented by a dataset, which is distinguished by a unique, two-character DOMAIN code that should be used consistently throughout the submission. This DOMAIN code is used as the dataset name, the value of the DOMAIN variable within that dataset, and as a prefix for most variable names in the dataset. Any valid time variable is permissible for use in any submission dataset. For CDISC models, dates and time of day need to be stored according to the international standard ISO 8601.

The template for ISO 8601 dates/times is as follows:

YYYY-MM-DDThh:mm:ss

Where

YYYY = four-digit year
MM = two-digit representation of the month (01=January,etc)
DD = two-digit day of month (01 through 31)
T = separator to indicate time information follows
hh = two digits of hour (00 through 23) (am/pm is NOT allowed)
mm = two digits of minute (00 through 59)
ss = two digits of second (00 through 59)

Key aspects of the ISO 8601 standard are as follows:

- ISO 8601 represents dates as a text string using the notation YYY-MM-DD.
- ISO 8601 represents times as a text string using the notation hh:mm:ss.
- ISO 8601 does not require dashes to separate date parts nor colons to separate time parts;
- When a date is stored with a time in the same variable (as a date-time), the date is written in front of the time and the time is preceded with 'T' using the notation YYYY-MM-DDThh:mm:ss (e.g. 20050921T00:00:01).

Implementation of the ISO standard means that nominal date/time variables are no longer numeric data types but will now have to be character/text data types. The date/time variables are called --DTC, where '--' is the two letter domain code [with or without the 'ST' or 'EN' strings indicating start or stop dates].

START CONVERTING FROM TEXT DATE/TIME VALUES

Biogen Idec collects date in yyyymmdd and time in hhmmss on the Case Report Form (CRF). Since there are partial dates in data such as medical history, the derived numeric date or time variable is missing when it is not a complete date. Using text date/time variables in the raw data to create ISO 8601 date/time can show what is best known for a date or time. SDTM V.1.1 has defined how to represent this Date/Time precision. According to ISO 8601, precision is specified by the number of parts present in the date, time, or date-time value. Whatever is unknown is not included. If the date is completely missing then the date value should be null. For example, if month is unknown, then only year is included in the value string. If day is unknown, only year and month are included in the value string. This works sufficiently for clinical trials, as it is very unlikely that a time would be known if the entire date is not known, or that a day would be known without knowing the month.

ISO 8601 representation of complete and partial date/time values: see the table below for examples:

	Date and time as originally recorded	Interval of Uncertainty	Nominal Date/Time
1	June 29, 1956 11:32:09	Complete date	1956-06-29T11:32:09
2	June 29, 1956 11:32	Unknown seconds	1956-06-29T11:32
3	June 29, 1956 11	Unknown minutes	1956-06-29T11
4	June 29, 1956	Unknown time	1956-06-29
5	June, 1956	Unknown day	1956-06
6	1956	Unknown month	1956

A MACRO CREATING SDTM ISO 8601 COMPLAINT DATE/TIME VARIABLE

The macro below converts character date and time variables into a SDTM ISO 8601 compliant character date/time variable. Since Biogen Idec stores dates in the clinical trial database as a character field with a missing hierarchy from right to left (i.e. month or day cannot exist if year is not present), the macro is built based on that assumption.

```
/*==========================================================
 This macro is to create a SAS® date time variable from
 a text date, impute the date if necessary, and generate
 text datetime variable
 date_in: input character date in yyyymmdd fashion
 time_in: input character time value
 for: indicate imputation
      for=START, missing month imputed as January, and missing
          day as the 1st.
      for=STOP, missing month imputed as December, and missing
          day as the last day of the month
      for=  ,default. No imputation. Partial date returns missing
          date time value.
 v_dt: output SAS date variable in the format of date9.
 var_txt: output CDISC date in character format
==========================================================*/
%macro dt_time(date_in=evld, time_in=, for=, var_out=, v_dt=);
  length &var_out $19;
  if length(&date_in)=8 then  date_n=input(&date_in, yymmdd8.); ❶
  else if &date_in='' then do;                                  ❷
       date_n=.;
       &var_out='';
  end;
  else do;
  %if %upcase(&for)=STOP %then %do;
       if length(&date_in)=4 then date_n=input(compress(&date_in||'1231'),yymmdd8.); ❸
      else if length(&date_in)=6 then
      date_n=intnx('month',input(&date_in,yymmn6.),0,'end');                      ❹
      &var_out=put(date_n,yymmdd10.);
  %end;
  %if %upcase(&for)=START %then %do;
  if length(&date_in)=4 then date_n=input(compress(&date_in||'0101'),yymmdd8.);
```

```
      else if length(&date_in)=6 then date_n=input(&date_in,yymmn6.);
      &var_out=put(date_n,yymmdd10.);
   %end;
   %if %upcase(&for)= %then %do;
     date_n=.;
     if length(&date_in)=4 then &var_out=&date_in;    ❺
     else if length(&date_in)=6
       then &var_out=compbl(substr(&date_in,1,4)||'-'||substr(&date_in,5,2));  ❻
   %end;
   end;

  %if %length(&time_in)>0 %then %do;
       h=input(substr(&time_in,1,2),2.);
       m=input(substr(&time_in,3,2),2.);
       s=input(substr(&time_in,5,2),2.);
       dtm=dhms(date_n,h,m,s);    ❼
  if length(&date_in)=8 then do;
   if h ne . and m ne . and s ne . then &var_out=put(dtm,is8601dt.);  ❽
   else if h ne . and m ne . and s= .
     then &var_out=compbl(put(date_n,yymmdd10.)||'T'||
                   substr(&time_in,1,2)||':'||substr(&time_in,3,2));
   else if h ne . and m= .    and s= .
     then &var_out=compbl(put(date_n,yymmdd10.)||'T'||substr(&time_in,1,2));
   else &var_out=put(date_n,yymmdd10.);
   drop h m s dtm;
  end;
  %end;
  %else %if %length(&time_in)=0 %then %do ;
    if date_n ne .  then &var_out=put(date_n,yymmdd10.);
  %end;
  %if %length(&v_dt)>0 %then %do;
     if &date_in='' then &v_dt=.;
     else &v_dt=date_n;
     format &v_dt date9.;
  %end;
  drop date_n;
%mend;
```

❶ When date is complete in yyyymmdd, create SAS date with a format of yymmdd8., which shows date as yyyy- mm-dd.

❷ When the entire date is missing, the output date is missing

❸ When imputation is necessary, impute the stop date to the last date of a year, and start date the first date of a year.

❹ When only year and month are known, impute the date to the last date of the month. When date is missing, using **yymmn6**. format returns the first date of the month. **INTNX** function returns the last date of the month.

❺ When only year is available, only keep the year part.

❻ When year and month available, display the date in yyyy-mm.

❼ Use **dhms** function to create SAS date and time variable.

❽ Create ISO 8601 date/time.

STUDY DAY CALCULATION

The Study Day variable (--DY) describes the relative day of the observation starting with the reference date as Day 1. All Study Day values are integers. It is defined as follows:

--DY = (date portion of --DTC) - (date portion of RFSTDTC) + 1; * If --DTC is on or after RFSTDTC;
--DY = (date portion of --DTC) - (date portion of RFSTDTC) + 1; * If --DTC proceeds RFSTDTC;

where, RFSTDTC is the subject reference start date/time from demographics domain. It is usually equivalent to date/time of first treatment date.

MACRO CALCULATING STUDY DAY

```
/*============================================================================
This macro is to calculate Study Day (--DY) as defined in SDTM V1.1
endc = date/time of an end point, can be in ISO 8601 or yyyymmdd depending on
        their form in the raw data
refd = subject’s reference date, set default as first treatment date(RFSTDTC) and
       assume &refd is merged in the dataset
studyday = output variable (--DY), -- should be replaced by a proper domain name
=============================================================================*/
%macro defdates(studyday=, endc=, refd=rfstdtc);
   if &endc='' then &studyday=.;
    else do;
    if length(&endc)=10 then do;
    if input(substr(&endc,1,10),yymmdd10.)<input(substr(&refd,1,10),yymmdd10.)
       then &studyday=input(substr(&endc,1,10), yymmdd10.) -
                      input(substr(&refd,1,10), yymmdd10.);

       else &studyday=input(substr(&endc,1,10), yymmdd10.) -
                     input(substr(&refd,1,10), yymmdd10.) + 1;
    end;
    else if length(&endc)=8 then do;
     if input(&endc, yymmdd8.)<input(substr(&refd,1,10), yymmdd10.)
     then &studyday=input(&endc, yymmdd8.) - input(substr(&refd,1,10),yymmdd10.);
     else &studyday=input(&endc, yymmdd8.) - input(substr(&refd,1,10), yymmdd10.)+1;
    end;
    end;
drop &refd;
%mend;
```

CONCLUSION

The macros present some approaches to creating study day and ISO 8601 CDISC date/time variables. The input character date variable has an eight-digit number for a complete date. If the input raw data changes from data collection of the raw date, the macro would need to be changed accordingly.

When imputation of the date is required, the rule can be different from what we have done. We can make the dates to be imputed macro variables. Since the imputation is not often, we didn’t go to great lengths in setting that up.

With more companies in the industry using CIDISC, we hope there are better tools and process to make the converting better.

REFERENCES

CDISC website, www.cdisc.org
The Study DATA Tabulation Model (SDTM), final version 1.1 April 28, 2005
The Study DATA Tabulation Model (SDTM), version 1.1 draft, June 25, 2004

ACKNOWLEDGMENTS

Many thanks to Jo-wei Wan and Eric Ding who gave us valuable feedback on these macros.

CONTACT INFORMATION

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

Song Liu
Biogen Idec
5200 Research Place
San Diego, CA 92122
Work Phone: 858-401-8935
Email: Song.Liu@biogenidec.com

Brett Sellars
Biogen Idec
5200 Research Place
San Diego, CA 92122
Work Phone: 858-401-5786
Email: Brett.Sellars@biogenidec.com