

Date Conversions in SDTM and ADaM Datasets

Huei-Ling Chen, Merck & Co., Inc., Rahway, NJ

Lily Zhang, Merck & Co., Inc., Rahway, NJ

Lili Chen, Merck & Co., Inc., Rahway, NJ

ABSTRACT

CDISC (Clinical Data Interchange Standards Consortium) has recommended using the SDTM (Study Data Tabulation Model) data structure to store source data, and the ADaM (Analysis Data Model) data structure to store data for statistical analysis and reporting. The SDTM IG (Implementation Guide) requires using the international standard ISO8601 format to store dates and times. The ISO8601 standard represents dates and times as text. However, dates and times in ADaM datasets are numeric. The purpose of this paper will be to answer how to convert a text date or time from source data into the ADaM date/time format and how to convert a numeric date or time to ISO8601 standard format. This paper provides simple and straightforward syntax and examples to illustrate the conversion methods. In addition, methods to impute the partial date or time are also discussed in this paper.

KEYWORDS

SDTM, ADaM, ISO8601 Format, PUT, INPUT, INTNX

INTRODUCTION

EXAMPLES OF SDTM DTC VARIABLE

A SDTM dataset is source data that will be converted to an analysis dataset (ADaM) to produce the final report and listing. The collected date and time in the SDTM dataset is represented as a character variable. As an example, the date/time variables AESTDTC (Start Date/Time of Adverse Event) and AEENDTC (End Date/Time of Adverse Event) collected in the SDTM dataset AE (Adverse Experience) are defined as characters in the following:

SUBJID	DOMAIN	AESTDTC	AEENDTC
11111	AE	2007-06-03T15:22:39	2009-03-15T10:52:07
22222	AE	2007-06-03T15:22	2009-03-15T10:52
33333	AE	2007-06-03T15	2009-03-15T10
44444	AE	2007-06-03	2009-03-15
55555	AE	2007-03	2009-03
66666	AE	2007-06	2009-06
77777	AE	2007-02	2009-02
88888	AE	2006-02	2008-02
99999	AE	2007	2009

Attributes:

Variable	Type	Len	Format	Label
AESTDTC	Char	19	\$Y9.	Start Date/Time of Adverse Event
AEENDTC	Char	19	\$Y9.	End Date/Time of Adverse Event

Here ISO8601 format [YYYY-MM-DDThh:mm:ss] is used for SDTM DTC variables, "T" is used as a separator to divide the date and time portions.

EXAMPLES OF ADaM DATE / TIME VARIABLE

The following example shows the date/time variable CFDTC from the CF domain and the ADaM variables ADTM, ADT, and ATM derived from CFDTC. The ADaM Date/Time variables are numeric variables, formatted so as to be more easily readable.

SUBJID	DOMAIN	CFDTC	ADTM	ADT	ATM
11111	CF	2009-05-15T21:27:00	2009-05-15T21:27:00	2009-05-15	21:27:00
22222	CF	2009-06-12T09:31:00	2009-06-12T09:31:00	2009-06-12	09:31:00
33333	CF	2010-07-13T20:05:00	2010-07-13T20:05:00	2009-07-13	20:05:00

Attributes:

Variable	Type	Label
ADTM	Num	Analysis Datetime
ADT	Num	Analysis Date
ATM	Num	Analysis Time

Forms of numeric date/time fields can be used in ADaM include DATEw, DDMMYYw, IS8601xx, etc. Examples in this paper show the numeric fields with IS8601xx formats which are preferred here. To convert the date variables defined in SDTM datasets to numeric variables in ADaM datasets, a flexible and simple way can be used – SAS **IS8601** format.

SAS® IS8601 FORMAT

Starting with SAS version 8.2, the IS8601 FORMATS and INFORMATS are available to the SAS user. The IS8601DT informat is used to create an ADTM variable. Both the IS8601DA and IS8601DN formats can be used to create the ADT variable. The difference between these two Date formats will be discussed later in this paper. The IS8601TM format is for ATM variables.

The table below lists the description and the value pattern for the IS8601 INFORMATS SAS provides.

Informat	Description	Pattern
IS8601DA	date	YYYY-MM-DD
IS8601DN	datetime with date portion	YYYY-MM-DD
IS8601DT	datetime	YYYY-MM-DDThh:mm:ss[.fffff]
IS8601TM	time	hh:mm:ss[.fffff]

SAMPLE CODE: CONVERTING SDTM DTC VARIABLES TO ADAM DATE/TIME VARIABLES

- For a SDTM DTC variable with time portion (YYYY-MM-DDThh:mm:ss), the following three INPUT statements are sufficient to convert this text variable to numeric variables.

```

DATETIME:
    ADTM = input(CMSTDTC, is8601dt.);

DATE:
    ADT  = input(CMSTDTC, is8601da.);

TIME:
    ATM  = input(substr(CMSTDTC, 12, 8), is8601tm.);

```

The IS8601DT informat converts the character CMSTDTC date-time value in YYYY-MM-DDThh:mm:ss format to a numeric date-time value. The IS8601DA informat extracts and converts the character CMSTDTC date part in YYYY-MM-DD format to a numeric date. The IS8601TM informat extracts and converts the character CMSTDTC time part in hh:mm:ss format to a numeric time.

- For a SDTM DTC variable without time portion (YYYY-MM-DD), a single INPUT statement is sufficient to convert this text variable to numeric variable.

```

DATE:
    ADT  = input(CMSTDTC, is8601da.);

```

The IS8601DA informat converts the character CMSTDTC date in YYYY-MM-DD format to a numeric date.

Note that an error message will be generated when the DTC variables are not complete.

The following example shows a partial value for date/time variable CFDTTC in CF:

SUBJID	DOMAIN	CFDTTC
22222	CF	2009-06-12

The function call resulted in the error:

```

ADTM = input(CFDTTC, is8601dt.);

```

NOTE: Invalid argument to function INPUT at line 24 column 13
 subjid=22222 cfdtc=2009-06-12 adtm=. _ERROR_=1 _N_=2
 NOTE: Mathematical operations could not be performed at the following places.
 The results of the operations have been set to missing values.
 Each place is given by: (Number of times) at (Line):(Column).

When collected datetime may be incomplete, caution is recommended. If an ADaM numeric datetime variable is desired, but only the date has been collected, imputation of the missing time portion should be imputed. Other suggestions are to fill in ":00" for a value with missing seconds (YYYY-MM-DDThh:mm).

More details about imputation will be discussed later in this paper.

SAMPLE CODE: CONVERTING NUMERIC DATA/TIME TO ISO8601 STANDARD FORMAT

- o The PUT statements can convert a numeric Data/Time to ISO8601 standard format.

```
MYDATEC = put(MYDATE, is8601da.);
if MYTIME ne . then MYTIMEC = put(MYTIME, is8601tm.);
if MYTIMEC = '' then CFDTC = MYDATEC;
else CFDTC = trim(MYDATEC) || "T" || trim(MYTIMEC);
```

The IS8601DA format converts the numeric MYDATE date variable in YYYY-MM-DD format to a character date. The IS8601TM format converts the numeric MYTIME time variable in hh:mm:ss format to a character time. Create CFDTC in an ISO8601 standard format variable by connecting the Date and Time with 'T' in between date part and time part.

IMPUTE PARTIAL DATE OR TIME

In situations with SDTM DTC variables containing partial date or time, incomplete records cannot be converted to a numeric value by using the ISO8601 format directly. The following tables show some common incomplete date/time value format:

Date/time formats	Example	Comments
YYYY-MM-DDThh:mm:ss	2007-06-03T15:22:39	Complete format
YYYY-MM-DDThh:mm	2007-06-03T15:22	Missing seconds
YYYY-MM-DDThh	2007-06-03T15	Missing minutes and seconds
YYYY-MM-DD	2007-06-03	Missing hours, minutes and seconds
YYYY-MM	2007-06	Missing day and time
YYYY	2007	Missing day, month and time

These partial date/time records should be imputed before they are converted to numeric values. The rules to impute partial date/time often depend on therapeutic area convention. For time part, some assign a missing part to 0 if seconds, minutes or hours are missing, some assign hour as 12, and minutes/second as 30 if they are missing. For the date part, some impute the day with the middle of month if day is missing, and date as July 1 if the month is missing. Imputation rules are protocol specific. This paper will illustrate one method to impute the partial dates of AESTDTC and AEENDTC variables based on the following assumptions:

- 1) If seconds, minutes, and hours are missing, assume the time part as midnight (T00:00:00).
- 2) If day of AESTDTC is missing, impute it as the first day of the month. If day of AEENDTC is missing, impute it as the end of the specific month. For example, there are 31 days in January, March, May, July, August, October and December, and 30 days in April, June, September and November, and 28 or 29 days in February,
- 3) If the start month is missing, impute it as January; if the end month is missing, impute it as December.

Below is a sample dataset in which some records have partial values in AESTDTC and AEENDTC.

SUBJID	DOMAIN	AESTDTC	AEENDTC
11111	AE	2007-06-03T15:22:39	2009-03-15T10:52:07
22222	AE	2007-06-03T15:22	2009-03-15T10:52
33333	AE	2007-06-03T15	2009-03-15T10

44444	AE	2007-06-03	2009-03-15
55555	AE	2007-03	2009-03
66666	AE	2007-06	2009-06
77777	AE	2007-02	2009-02
88888	AE	2006-02	2008-02
99999	AE	2007	2009

First, the partial date variables should be separated into year, month, day, hour, minute and second and stored as numeric variables. The following SAS statements illustrates this.

Separate date part and time part:

```
stdatec=scan(aestdct, 1, 'T');
sttimec=scan(aestdct, 2, 'T');
```

Create numeric variables for year, month, and day:

```
styear=input(scan(stdatec, 1, '-'), best8.);
stmonth=input(scan(stdatec, 2, '-'), best8.);
stday=input(scan(stdatec, 3, '-'), best8.);
```

Create numeric variables for hour, minute and second:

```
sthour=input(scan(sttimec, 1, ':'), best8.);
stmin=input(scan(sttimec, 2, ':'), best8.);
stsec=input(scan(sttimec, 3, ':'), best8.);
```

Second, fill in the specific value in missing year, month, day, hour, minute and second based on the above assumptions. Since the end day of each month is not a constant, the function INTNX is used to calculate the end day in each month. Refer to the SAS statements below for this imputation.

Date Part:

```
if enday ne . and enmonth ne . and enyear ne . then endate=mdy(enmonth, enday, enyear);
else if enday = . and enmonth = . then endate=mdy(12, 31, enyear);
else if enday = . then endate=intnx('month', mdy(enmonth, 01, enyear), 0, 'E');
```

Time Part:

```
if ensec ne . and enmin ne . and enhour ne . then entime=hms(enhour, enmin, ensec);
else if ensec = . and enmin = . and enhour = . then entime=hms(0, 0, 0);
else if ensec = . and enmin = . then entime=hms(enhour, 0, 0);
else if ensec = . then entime=hms(enhour, enmin, 0);
```

The following output shows imputed date and time part as is8601da and is8601tm formats:

Imputed date and time part									
Obs	subjid	stdatec	stdate	endatec	endate	sttimec	sttime	entimec	entime
1	11111	2007-06-03	2007-06-03	2009-03-15	2009-03-15	15:22:39	15:22:39	10:52:07	10:52:07
2	22222	2007-06-03	2007-06-03	2009-03-15	2009-03-15	15:22	15:22:00	10:52	10:52:00
3	33333	2007-06-03	2007-06-03	2009-03-15	2009-03-15	15	15:00:00	10	10:00:00
4	44444	2007-06-03	2007-06-03	2009-03-15	2009-03-15		00:00:00		00:00:00
5	55555	2007-03	2007-03-01	2009-03	2009-03-31		00:00:00		00:00:00
6	66666	2007-06	2007-06-01	2009-06	2009-06-30		00:00:00		00:00:00
7	77777	2007-02	2007-02-01	2009-02	2009-02-28		00:00:00		00:00:00

8	88888	2006-02	2006-02-01	2008-02	2008-02-29	00:00:00	00:00:00
9	99999	2007	2007-01-01	2009	2009-12-31	00:00:00	00:00:00

At this point, the SAS code has imputed all partial dates and times and the dates and times can now be combined as complete date variables. We can now use the ISO8601 format to convert them to numeric date variables as illustrated in the code which follows.

```
stdtc=put(stdate, is8601da.)||'T' ||put(sttime, is8601tm.);
endtc=put(endate, is8601da.)||'T' ||put(entime, is8601tm.);
aestdtm=input(stdtc, is8601dt.);
aeendtm=input(endtc, is8601dt.);
```

Following are the imputed date variables and the DTF / TMF variables which are mandatory variables when the date/time variables are imputed.

Imputed result for partial date/time variables						
Obs	subjid	AESTDTC	AESTDTM	AESTDTF	AESTTMF	
1	11111	2007-06-03T15:22:39	2007-06-03T15:22:39			
2	22222	2007-06-03T15:22	2007-06-03T15:22:00			S
3	33333	2007-06-03T15	2007-06-03T15:00:00			M
4	44444	2007-06-03	2007-06-03T00:00:00			H
5	55555	2007-03	2007-03-01T00:00:00	D		H
6	66666	2007-06	2007-06-01T00:00:00	D		H
7	77777	2007-02	2007-02-01T00:00:00	D		H
8	88888	2006-02	2006-02-01T00:00:00	D		H
9	99999	2007	2007-01-01T00:00:00	M		H

IS8601DA VS. IS8601DN

When using IS8601DA or IS8601DN formats to retrieve the date part from a SDTM DTC variable, note that both the IS8601DA format and IS8601DN format look similar but the numeric value is actually different. IS8601DA is number of days since January 1, 1960. IS8601DN is number of seconds since January 1, 1960. In the example below, both IS8601DA and IS8601DN formats retrieve the correct date part from the DTC DATE/TIME string.

SDTM DTC Variable	IS8601DA	IS8601DN
2009-05-15T21:27:00	2009-05-15	2009-05-15
2009-06-12	2009-06-12	2009-06-12
2010-07-13T21:27	2010-07-13	2010-07-13

CAUTION

However, IS8601DN is a DATETIME value with date portion which is number of seconds since January 1, 1960. When comparing IS8601DN with other date variables and when the date variable is a DATE value, it will produce an unexpected outcome.

In the following syntax, this paper compares the date with a target date (variable FLAGDATE with value '30jun2010'd).

```
data adcf;
  set cf;
  format is8601da is8601da. is8601dn is8601dn. flagdate mmddyy8.;
  is8601da = input(cfdtc ,is8601da.);
  is8601dn = input(cfdtc ,is8601dn.);

  flagdate = '30jun2010'd;
```

```

if is8601da > flagdate then adtfn = 1;
if is8601dn > flagdate then adt1fn = 1;
run;

```

The output is as follows:

Obs	subjid	cfdtc	is8601da	is8601dn	flagdate	adtfn	adt1fn
1	11111	2009-05-15T21:27:00	2009-05-15	2009-05-15	06/30/10	.	1
2	22222	2009-06-12	2009-06-12	2009-06-12	06/30/10	.	1
3	33333	2010-07-13T21:27	2010-07-13	2010-07-13	06/30/10	1	1

The date of the first and the second record occur earlier than the FLAGDATE of 06/30/10. However, the comparison result returned for IS8601DN is not correct. The reason for this is that IS8601DN is actually a DATETIME value with the time part set to '00:00:00'.

Behind the format, the value of the IS8601DA and IS8601DN, and IS8601DT are as follows:

Obs	cfdtc	is8601dt	is8601da	is8601dn
1	2009-05-15T21:27:00	1558042020	18032	1557964800
2	2009-06-12	1560384000	18060	1560384000
3	2010-07-13T21:27	1594675620	18456	1594598400

To clarify this situation, use IS8601DT format for the value of IS8601DN.

```
format is8601dn is8601dt. is8601dt is8601dt. is8601da is8601da.;
```

IS8601DN is a value representing the number of seconds between January 1, 1960 within a specified date.

Obs	cfdtc	is8601dt	is8601da	is8601dn
1	2009-05-15T21:27:00	2009-05-15T21:27:00	2009-05-15	2009-05-15T00:00:00
2	2009-06-12	2009-06-12T00:00:00	2009-06-12	2009-06-12T00:00:00
3	2010-07-13T21:27	2010-07-13T21:27:00	2010-07-13	2010-07-13T00:00:00

SUGGESTED SOLUTION

It is highly recommended that IS8601DA be used for converting a SDTM DTC DATETIME value to an ADT variable in an ADaM dataset. Care should be taken so that the IS8601DN format is not misinterpreted.

INTNX - HOURS VS. MINUTES

In the case where the time portion needs to be considered, variables ADTM and ATM in an ADaM dataset are typically used in deriving an analysis endpoint. In situations involving creation of a threshold, such as censor time for an event that has not occurred, the INTNX function is frequently used for assigning a DATETIME value with intervals and is expected to return a DATETIME value.

Extra caution should be taken when using the INTNX statement for the "hours" interval.

EXAMPLE:

An intermediate threshold variable A6DTM will be generated as six hours after variable ADTM.

ADTM	Expected A6DTM value
2009-05-15T21:27:00	2009-05-16T03:27:00
2009-06-12T00:00:00	2009-06-12T06:00:00

Intuitively, the following statement should be used:

```
adtm6=intnx('hours', adtm, 6);
```

In return, the output result is not what is expected.

ADTM	Expected A6DTM value	A6DTM value (using INTNX HOURS Statement)
2009-05-15T21:27:00	2009-05-16T03:27:00	2009-05-16T03:00:00
2009-06-12T00:00:00	2009-06-12T06:00:00	2009-06-12T06:00:00

The reason for the resulting output is the 'HOURS' option in INTNX function calculates the 'INCREMENT' on the 'HOURS' interval, and before doing it, the minutes and seconds have been erased. In the example here, the INTNX function adds six (INCREMENT) HOURS (INTERVAL) to the starting point, HOURS, which is '21:00:00'. Hence, the value, '03:00:00', is returned which is six hours after the starting point of '21:00:00'.

SUGGESTED SOLUTION

When seconds are not collected, we recommend using MINUTES as the interval.

In the situation demonstrated here, the statement should be fixed as follows:

```
A6dtm=intnx('minutes', adtm, 360);
```

When seconds are collected, we recommend using SECONDS as the interval.

```
A6dtm=intnx('seconds', adtm, 360*60);
```

CONCLUSION

SAS IS8601 format utilized in the INPUT statement is a convenient tool for SAS users to convert SDTM DTC variables to ADaM numeric variables, such as ADTM, ADT, ATM. When needing to convert non-SDTM data to SDTM format data, the SAS IS8601 format utilized in a PUT statement can be used for date/time variables. When using various IS8601 feature formats, extra caution should be taken to avoid misinterpretation, particularly using the IS8601DA and IS8601DN formats.

REFERENCES

- Analysis Data Model Version 2.1 Final, (CDISC website: <http://www.cdisc.org/adam>)
- ADaM Implementation Guide, Version 1.0 (ADaMIG v1.0), (CDISC website: <http://www.cdisc.org/adam>)
- Study Data Tabulation Model, Version 1.2 (SDTM v1.2), (CDISC website: <http://www.cdisc.org/sdtm>)
- SDTM Implementation Guide for Human Clinical Trials (SDTMIG v.3.1.2), (CDISC website: <http://www.cdisc.org/sdtm>)

ACKNOWLEDGMENTS

The authors would like to thank the management team of Merck Research Laboratories for their advice on this paper/presentation.

Contact Information

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

Huei-Ling Chen
 Merck & Co., Inc.
 126 Lincoln Avenue
 P.O. Box 2000
 Rahway, NJ 07065
 Phone: 732-594-2249
 e-mail: Huei-Ling_Chen@merck.com

Lily Zhang
 Merck & Co., Inc.
 126 Lincoln Avenue
 P.O. Box 2000
 Rahway, NJ 07065
 Phone: 732-594-5413
 e-mail: Lily_Zhang2@merck.com

Lili Chen
 Merck & Co., Inc.

126 Lincoln Avenue
P.O. Box 2000
Rahway, NJ 07065
Phone: 732-594-0613
e-mail: Lili.Chen@merck.com

TRADEMARK

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.