PhUSE 2010

Paper CD05

How to go from an SDTM Finding Domain to an ADaM-Compliant Basic Data Structure Analysis Dataset: An Example

Qian Wang, MSD, Brussels, Belgium
Carl Herremans, MSD, Brussels, Belgium

ABSTRACT

The pharmaceutical industry is embracing the Clinical Data Interchange Standards Consortium (CDISC) Study Data Tabulation Model (SDTM) standard and is actively converting its Clinical Data Management Systems (CDMS) databases into SDTM-compliant data structures. The next challenge for the industry will be to adjust its processes and to produce CDISC Analysis Data Model (ADaM)-compliant analysis datasets from the SDTM datasets. This paper describes, in a systematic way, the steps needed to convert an SDTM finding data structure into ADaM-compliant analysis datasets.

INTRODUCTION

In December 2009, the *Analysis Data Model (ADaM) v2.1* and the *ADaM Implementation Guide v1.0* were finalized and posted on the CDISC website. The *Analysis Data Model (ADaM) v2.1* describes the fundamental principles and standards to follow in the creation of analysis datasets as well as their associated metadata. The *ADaM Implementation Guide v1.0* specifies, in further detail, ADaM standard dataset structure and variables. Now with recommended standards in place for both study tabulation data (SDTM) and analysis datasets (ADaM), one challenge for the industry lies in the implementation of these standards, especially in the derivation of ADaM analysis datasets from SDTM datasets.

This paper proposes, in a systematic way, the steps needed to convert an SDTM finding data structure into an ADaM-compliant Basic Data Structure (BDS). The SDTM lab domain (LB) is used as an example to construct an ADaM-compliant lab analysis dataset (ADLB). The process described is generic and can be applied to any SDTM finding domain in order to construct its corresponding ADaM BDS analysis dataset when applicable.

FUNDAMENTALS OF THE ADaM STANDARD

The Analysis Data Model v2.1 and ADaM IG v1.0 mandate that ADaM compliant analysis datasets adhere to certain fundamental principles as summarized by Becker [2010] below:

1. "Standardize" delivery to regulatory agencies;
2. Provide clear documentation of the content, source and quality of the analysis datasets;
3. Provide clear documentation of the results of a clinical trial (statistical methods, transformations, assumptions, derivations, imputations);
4. Provide a "roadmap" of how metadata, programs and documentation translate the Statistical Analysis Plan (SAP) to the statistical results;
5. ADaM datasets should be usable by current tools (e.g. SAS®);
6. Provide XML metadata for future analysis tool development;
7. Analysis-ready or "one proc away": This means ADaM datasets incorporate derived and collected data (from various SDTM domains, other ADaM datasets, or any combination thereof) into one dataset that permits analysis with little or no additional programming.

Although both included in a CDISC compliant submission, ADaM dataset structure is not the same as SDTM. Some main differences include:

1. ADaM datasets use redundancy for easy analysis – common variables may be found across all analysis datasets (e.g., population flags, subject identifiers, etc.);
2. ADaM datasets have a greater number of numeric variables (e.g., SAS formatted dates, numeric representation of a character grouping variable from SDTM);
3. ADaM datasets may combine variables from multiple SDTM domains;
4. ADaM datasets are named AD<xxxxxx>.

Traceability is key to ADaM and enables the reviewers to understand the relationship between the analysis results, the analysis datasets, and the SDTM domains.

The ADaM IG describes two ADaM standard data structures: the subject level analysis dataset (ADSL) and the Basic Data Structure (BDS).

ADSL contains one record per subject and describes the subjects' baseline characteristics, subject status, applicable subject-level population flags, subject-level treatment variables, and other relevant subject variables. Typically ADSL will be created via a merge of data copied or derived from the SDTM DM, SC, EX, SV and DS domains. The exact process for creation of the ADSL analysis dataset depends on a company's implementation and interpretation of the SDTM standard, as well as the kinds of subject-level facts required for the analysis and review of the particular study.

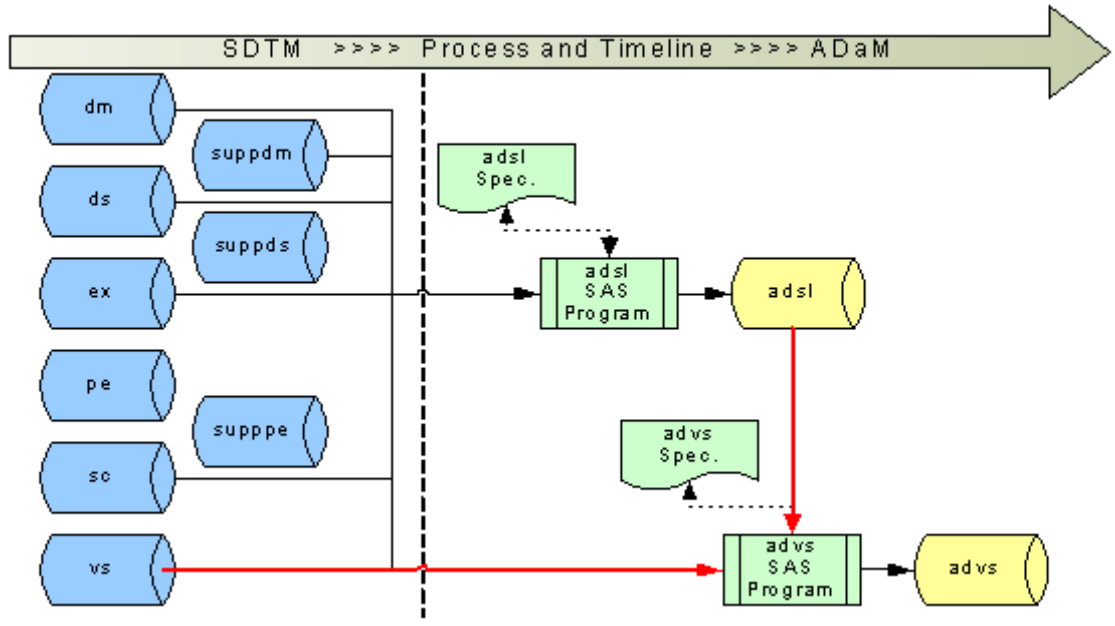
A BDS dataset contains one or more records per subject, per analysis parameter and per analysis timepoint (depending on the analysis). It describes the data being analyzed and also includes variables to support the analysis (e.g. covariates) as well as information to facilitate traceability. Although a BDS dataset also has a long and skinny structure similar to a SDTM finding domain, it may be derived from all classes of SDTM domains, other ADaM datasets and combination of those.

CONSTRUCTING AN ADaM BDS DATASET

A general process is described by Peterson and Izard [2010] in Figure 1 using VS as an example in the creation of ADVS. However few publications are available which go into the detailed implementation. Steffensen and Jepsen [2009] discussed some challenges encountered in the design of ADaM compliant datasets, but didn't give much detail how a BDS dataset can be constructed. Dey and Pyle [2010] described a solution to implement a BDS dataset, but the discussion is specific to the creation of response datasets for oncology trials.

In this paper, it's assumed that ADSL already exists and the BDS dataset is derived from a single finding domain for simplicity. However the methodology serves a good starting point and can be easily extended to support more complex derivations. SDTM lab domain (LB) is used as an example to construct a BDS lab analysis dataset (ADLB).

Figure 1



STEP 1: PREPARING FOR ADaM PRECURSOR BY ADDING SUPPQUAL TO SDTM

As the first step, the corresponding supplemental variables stored in the SUPPQUAL domain are added to the parent SDTM domain of interest to construct an analysis dataset precursor. This can be achieved by the following reproducible process:

1. Select the relevant records from the SUPPQUAL for the domain of interest ;
   In the case of SDTM LB domain, those records are therefore selected from the SUPPQUAL with the condition RDOMAIN='LB';
2. Transpose the dataset with composite key Unique Subject Identifier (USUBJID) and Identifying Variable Value (IDVARVAL), so that variables defined in QNAME are created with values from QVAL;
3. Join the transposed dataset with the corresponding parent domain based on Unique Subject Identifier (USUBJID) and Identifying Variable Value (IDVARVAL).

For the LB domain , it can be implemented in SAS with the following code:

```
proc transpose data=supplb
               out=supplb2 (rename=(idvarval=lbseq));
    by usubjid idvarval;
    var qval;
    id qnam;
    idlabel qlabel;
run;
```

A similar implementation in a SAS macro can be found in [Shostak, 2005].

One drawback of the implementation above is that it does not take into account the fact that some variables stored in SUPPQUAL are meant to be used as numerical variables when transposed. For example the variable LBSEQ is expected to be in numeric type by SDTM requirements as specified in the define.xml. To account for this need, some metadata information is needed to support the transposition. Below is a proposed solution using a SAS macro:

```
%addsuppqual(input_libary =,
             input_dataset_suppqual =,
             input_dataset_define =,
             output_library =
            );
```

Where

- input_library is the libname where the SDTM parent datasets resides;
- input_dataset_suppqual provides the SUPPQUAL dataset to be transposed;
- input_dataset_define provide the metadata information in which the variable type is specified;
- Output_library gives the libname where the SDTM dataset with SUPPQUAL addition will reside.

STEP 2: CONVERTING ISO8601 DATES

SDTM contains date and time values in the ISO8601 format (YYYY-MM-DDThh:mm:ss) as character variables. ADaM stores date and time values in numeric variables with a display format (e.g. yymmdd10) which will render the numeric values in human-readable fashion when printed or viewed. Hence the creation of an ADaM compliant dataset structure requires the conversion of the ISO8601 character date/time/datetime format to its corresponding native SAS numeric form. This can be achieved by a macro below:

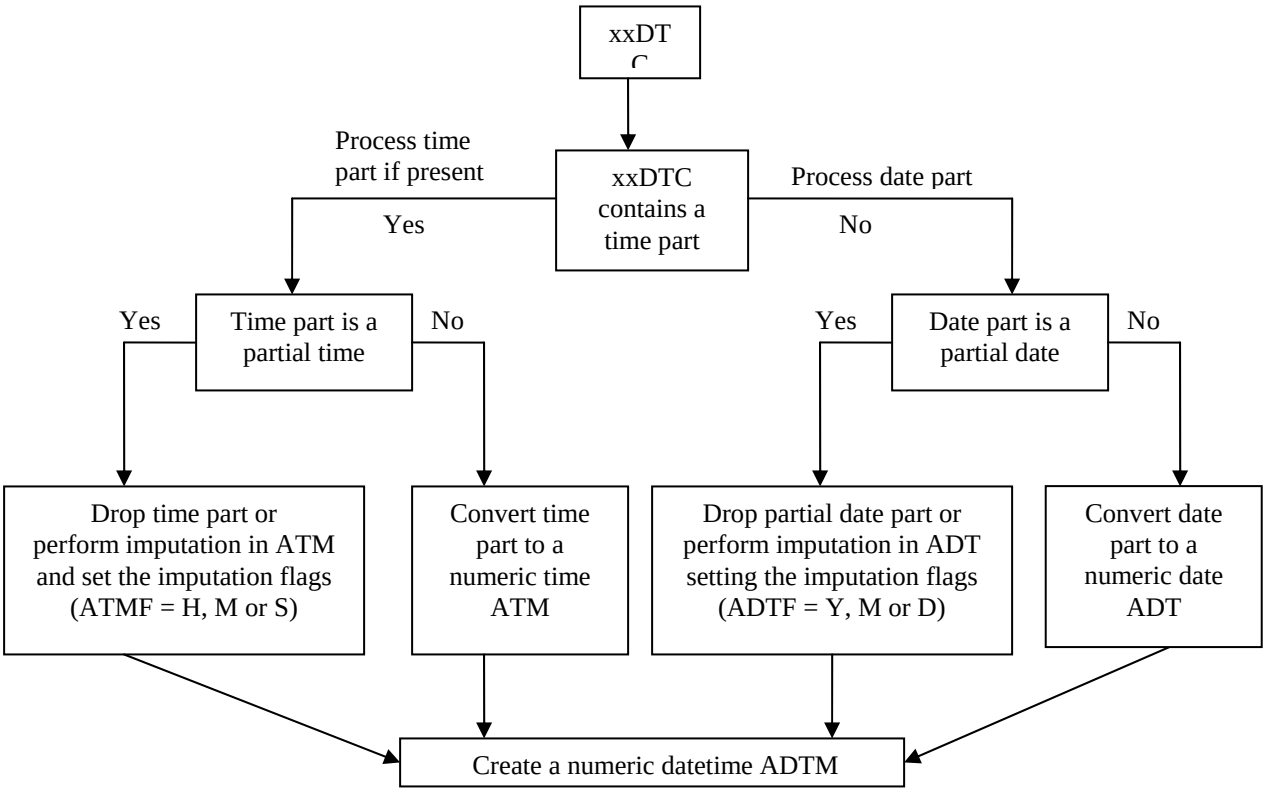
```
%convert2date(input_dataset=,
              Output_dataset=
             );
```

Although SAS provides informats to convert ISO8601 dates, times, and datetimes to SAS dates, time, and/or datetime (for example IS8601DA.), the macro should be able to handle complex data situation like missing or partial dates and time:

- If time is not collected, this is a straight-forward situation in which the use of the SAS informat IS8601DA is sufficient;
- If both date and time are collected, SDTM datetime variables may contain a mixture of date and time information, and both date and time can be partial or unknown. In order to deal with this lack of information, a series of variables are derived to capture date (DT), time (TM), datetime (DTM) as well as respective flags to indicate imputation if any.

The following comprehensive schema (Figure 2) describes the steps required to convert an SDTM XXDTC variable into numerical date/time variables ADT/ATM/ADTM. Similar steps can be used to convert other datetime variables in SDTM, e.g. STDTC.

Figure 2



In Table 1 the above scheme is illustrated with some examples assuming certain imputation logic in the case of partial date and time. Please note that the example gives one possible imputation implementation which might vary under different circumstances.

Table 1

--DTC	ADT	ADTF	ATM	ATMF	ADTM
2003-12-15T13:14:17	15Dec2003		13:14:17		15Dec2003:13:14:17
2003-12-15T13:14	15Dec2003		13:14:00	S	15Dec2003:13:14:00
2003-12-15T13	15Dec2003		13:00:00	M	15Dec2003:13:00:00
2003-12-15	15Dec2003				
2003-12	01Dec2003	D			
2003	01Jul2003	M			

STEP 3: CREATING ANALYSIS RELATIVE DAY

When a timing variable is expected in the BDS dataset, the variable Analysis Relative Day ADY is useful as it describes the number of days from Analysis Date ADT to a reference date. In a typical study the reference date may be defined as the date of the first exposure to treatment (ADSL.TRTSDT). A typical implementation to compute the relative day may be:

ADY = LBDT - ADSL. TRTSDT + (LBDT > ADSL.TRTSDT)

The addition of a binary value with the statement LBDT>ADSL.TRTSDT takes care of the detail that reference day is considered Day 1 instead of Day 0 in the analyses.

STEP 4: CREATING KEY ANALYSIS PARAMETER VARIABLES PARAMCD/PARAM AND AVAL

The Analysis Data Model Version 2.1 states: *The variable PARAM contains a unique description for every analysis parameter included in that dataset. The variable PARAMCD contains the short name of the analysis parameter in PARAM, with a one-to-one mapping between the two variables. Each value of PARAM identifies a set of one or more rows in the dataset.*

In a simplified situation in the context of the LB domain this can be achieved by

1. renaming LBTESTCD to PARAMCD;
2. combining LBTEST and LBSTRESU into PARAM;
3. renaming LBSTRESN to AVAL.

This is illustrated in the example below:

SDTM LB

LBTESTCD	LBTEST	LBSTRESN	LBSTRESU
NA	Sodium	139	mmol/L
K	Potassium	3.5	mmol/L

ADaM ADLB

PARAMCD	PARAM	AVAL
NA	Sodium (mmol/L)	139
K	Potassium (mmol/L)	3.5

The above implementation can be further generalized to support more complex data scenarios. As an example, the described dataset assumes that only blood chemistry results are collected. However, sometimes certain tests, e.g. Leukocytes (WBC) may need to be performed in both blood and urine. If both results are to be retained in the analysis dataset, PARAM and PARAMCD should be constructed in a way to allow easy distinguishing of the results from two specimen types.

Sometimes the endpoints can only be derived from the collected information. In the example below, certain lab tests are performed multiple times on a day, and the Statistical Analysis Plan specifies that the mean result is to be used in the analyses. When translating into analysis dataset creation, this may be implemented by inserting a new record with derivation method indicated in the Derivation Type variable (DTYPE). Typically an analysis flag (ANL01FL in this case) is also added to in the analysis of a particular day.

SDTM LB

LBDY	LBTESTCD	LBTEST	LBSTRESN	LBSTRESU
1	NA	Sodium	141	mmol/L
1	NA	Sodium	139	mmol/L
14	NA	Sodium	145	mmol/L

ADaM ADLB

ADY	PARAMCD	PARAM	AVAL	DTYPE	ANL01FL
1	NA	Sodium (mmol/L)	141		
1	NA	Sodium (mmol/L)	139		
1	NA	Sodium (mmol/L)	140	AVERAGE	Y
14	NA	Sodium (mmol/L)	145		Y

Step 4 with the example dataset illustrated in this section can be implemented in a macro with the following parameters:

```
%finding(input_dataset=,
         domain=,
         summary_function= AVERAGE,
         output_dataset=
        );
```

Where

- input_dataset provides the name of the SDTM dataset to be processed;
- domain specifies the 2 letter prefix that identifies the variables (for example LB);
- summary_function defines the type of action to be taken in case of multiple records on the same day (to calculate the mean in this case);
- output dataset provides the name of the ADaM dataset to be created.

STEP 5: ADDING TIME WINDOWS

Analysis visit window is used to categorize events/findings into respective analysis time periods. A typical timing variable is AVISIT. The CDISC ADaM implementation guide states: *AVISIT represents the analysis visit of the record, but it does not mean that the record was analyzed. There are often multiple records for the same subject and parameter that have the same value of AVISIT. ANLzzFL and other variables may be needed to identify the records selected for any given analysis.*

In case multiple events/findings occur within the same time interval, conventions may be applied to identify one event/finding for this time interval to be used in the analysis. Some possible conventions are, for example:

- last event within a time window; or
- event closest to the target day within a time window

The example below illustrates how the time window can be added to the dataset from Step 4.

Table 2: Time window

Time Point (Target Day)	Time window in Days
Baseline (1)	-70 to 7
Treatment 1 (45)	8 to 90
Treatment 2 (135)	91 to 180

Table 2 is an example of a time window definition for lab measurement from a Statistical Analysis Plan. It can be translated to a dataset as shown below. The AVISIT presents the analysis visit description. The corresponding AVISITN specifies the sorting order for the AVISIT variables.

Time window dataset

Start day	Stop day	Target	AVISIT	AVISITN	Baseline
-70	7	1	Baseline	0	Y
8	90	45	Treatment 1	1	
91	180	135	Treatment 2	2	

The analysis window can then be applied to the dataset from step 4 as follows:

Input dataset

ADY	PARAMCD	AVAL
-18	NA	141
1	NA	140
14	NA	145
46	NA	149

⇩ Apply time windows

Output dataset

ADY	PARAMCD	AVAL	AVISIT	AVISITN	AWTARGET	AWRANGE	ABLFL	ANL02FL
-18	NA	141	Baseline	0	1	-70 to 7		
1	NA	140	Baseline	0	1	-70 to 7	Y	Y
14	NA	145	Treatment 1	1	45	8 to 90		
46	NA	149	Treatment 1	1	45	8 to 90		Y

By comparing ADY with the Start Day and Stop day of each analysis window range (AWRANGE), each record falls into one window. Applying the convention to select the event closest to the target day within a time window for analysis, the variable ABLFL indicates the record containing the baseline values and the flag variable ANL02FL indicates the records to select for the particular time window. All collected records are retained in this case to support traceability of the derivation.

STEP 6: DERIVING CHANGE FROM BASELINE

With a baseline record identified in the previous step, further derivations e.g. baseline value (BASE) and change from baseline (CHG) can be calculated:

Output dataset

ADY	PARAMCD	AVAL	AVISIT	AVISITN	ABLFL	BASE	CHG
-18	NA	141	Baseline	0		140	1
1	NA	140	Baseline	0	Y	140	0
14	NA	145	Treatment 1	1		140	5
46	NA	149	Treatment 1	1		140	9

Combining all 6 steps so far, the output dataset is a preliminary ADaM BDS ADLB with the following fields:

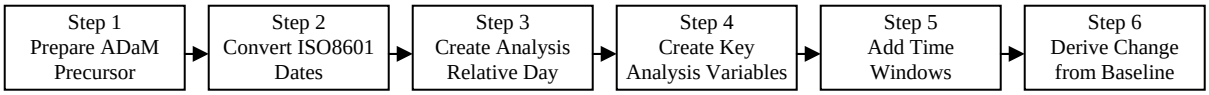
Variable	Label
USUBJID	Unique Subject Identifier
SUBJID	Subject Identifier for the Study
AVISIT	Analysis Visit
AVISITN	Analysis Visit (N)
ADT	Analysis Date
ADY	Analysis Relative Day
PARAMCD	Parameter Code
PARAM	Parameter
AVAL	Analysis Value
ABLFL	Baseline Record Flag
BASE	Baseline Value
CHG	Change from Baseline
AWTARGET	Analysis Window Target
AWRANGE	Analysis Window Valid Relative Range
ANL01FL	Analysis Record Flag
ANL02FL	Analysis Record Flag 2

Starting from this basic structure, further derivations can be added. Some possible additions are:

1. other study-specific flags, e.g. criteria for predefined limits of change;
2. additional rows, e.g. additional endpoint derived as the mean value by PARAMCD for the entire treatment period;
3. additional columns, e.g. TRTP.

CONCLUSION

In this paper, six basic steps are described to transform an SDTM finding dataset into an ADaM-compliant BDS analysis dataset using lab data as an example. These six steps are again summarized below in a flow chart:



The implementation proposed is generic and can be incorporated into a macro library as standard tools to support efficient construction of ADaM analysis datasets across projects.

References

Analysis Data Model (ADaM) Implementation Guide, Version 1.0. Final version, published by CDISC December 17, 2009. Available for download at http://www.cdisc.org. Note: membership and/or registration required as of March 10, 2010.

Analysis Data Model (ADaM), Version 2.1. Final version, published by CDISC December 17, 2009. Available for download at http://www.cdisc.org. Note: membership and/or registration required as of March 10, 2010.

Becker, Matthew, "Insights into ADaM" PharamSUG 2010, available for download at http://www.lexjansen.com/pharmasug/2010/hw/hw06.pdf

Brucken, Nancy and Slagle, Paul, "From SAP to ADaM: The Nuts and Bolts", PharmaSug 2010 available for download at http://www.lexjansen.com/pharmasug/2010/hw/hw02.pdf

Dey, Mei and Pyle, Lisa, "Applying ADaM Principles in Developing a Response Analysis Dataset". PharmaSUG 2010, available for download at http://www.lexjansen.com/pharmasug/2010/cd/cd03.pdf

Peterson, Terek and Izard, David , "The 5 Biggest Challenges of ADaM", PharmaSug 2010, available for download at http://www.lexjansen.com/pharmasug/2010/cd/cd10.pdf

Shostak, Jack, "Implementation of the CDISC SDTM at the Duke Clinical Research Institute", PharmaSug 2005, available for download at http://www.lexjansen.com/pharmasug/2005/fdacompliance/fc01.pdf

Steffensen, Karin and Jepsen, Gitte, "An Implementation of ADaM standards NOT driven by a submission", PHUSE 2009, available for download at http://www.lexjansen.com/phuse/2009/cd/cd13.pdf

ACKNOWLEDGMENTS

The authors would like to express appreciation to John Troxell for his input during the preparation of this paper.

CONTACT INFORMATION

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

Qian Wang
Statistical Programming Analyst
MSD
Clos du Lynx 5
B-1200 Brussels [Belgium]
Tel: ++32 27766303
E-mail: qian_wang@merck.com
Web: http://www.merck.com

Carl Herremans
Senior Statistical Programming Analyst
MSD
Clos du Lynx 5
B-1200 Brussels [Belgium]
Tel: ++32 27766305
E-mail: carl_herremans@merck.com
Web: http://www.merck.com