Paper 222-2011

HOW TO BUILD ADaM BDS FROM MOCK UP TABLES

Kevin Lee, Cytel, Inc., Chesterbrook, PA

ABSTRACT

The paper is intended for Clinical Trial SAS® programmers who are interested in ADaM(Analysis Data Model) BDS(Basic Data Structure). The paper will discuss basic structures of ADaM data sets. It will also discuss how ADaM data sets are built from Mock Up tables. The paper will provide examples of Mock Up tables, ADaM data sets and SAS codes that generate final numbers of Mock Up tables from ADaM data sets.

INSTRUCTION OF BDS

The BDS is the standard domain structure in ADaM. The BDS is designed as one or more records per subject per analysis parameter per analysis time point. One of the main purposes of ADaM BDS is analysis-ready, meaning that all the numbers in the final report should be calculated with one procedure in SAS. The naming convention of BDS is ADxxxxxx.

STRUCTURE OF BDS

The ADaM BDS consists of following class variables.

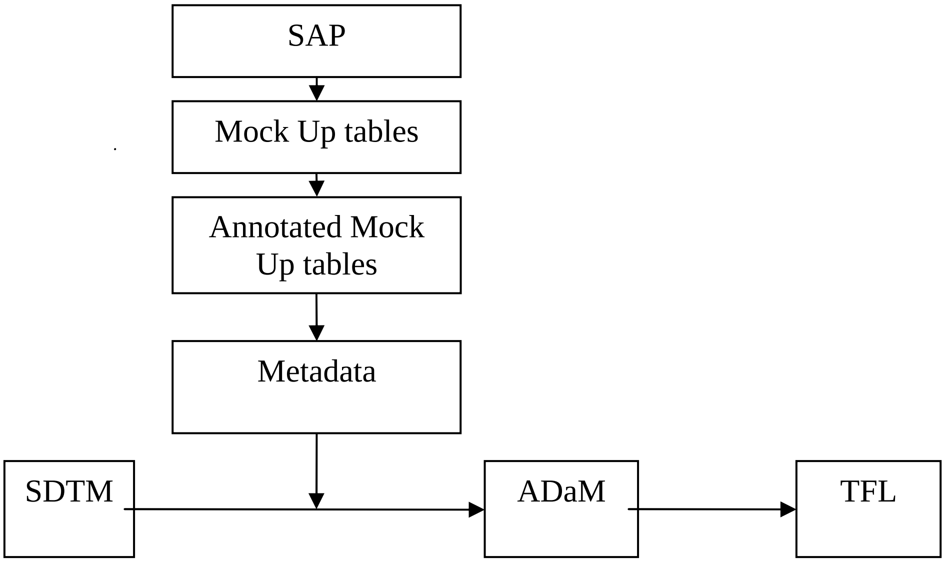
- Subject Identifier Variables
- Treatment Variables
- Timing Variables
- Analysis Parameter Variables
- Analysis Descriptor Variables
- Indicator Variables
- Analysis Enabling Variables
- Data point Traceability Variables
- SDTM Variables

STEPS TO CREATE ADaM BDS from Mock Up tables

The following steps can be used to create ADaM BDS from Mock Up tables

- Design Mock Up tables (typically created by Statistician) according to SAP
- Annotate Mock Up tables
- Design Metadata documents according to Mock Up tables
- Create ADaM BDS data sets according to Metadata

THE FLOWCHART OF THE PROCEDURE



MOCK UP TABLE

Below is the sample Mock Up tables.

Table 14.4.1
Summary of table of Creatine at baseline
(Per Protocol Population)

	Group 1: Treatment 1 (N=xxx)		**Group 2: Placebo (N=xxx)**	
	n	**Observed Mean Value**	**n**	**Observed Mean Value**
Creatine				
Log of Creatine				

In order to get final numbers in this table, SAS programmers need to use proc sql count and mean function, proc means or other procedures. In this paper, proc sql will be used. The procedure, proc sql, requires baseline, population, treatment, laboratory test and value information.

The following variables are needed.

1. Protocol population variable – PPROTFL
2. Baseline – AVISIT, AVISITN
3. Treatment variable – TRTAN, TRTA
4. Laboratory test Variable – PARAM, PARAMCD
5. Count and Observed Mean Value Variable – AVAL

The following parameters are also needed.

1. Creatine – Its paramcd is “CREAT” and its analysis values, AVAL, come from LB.LBSTRESN.
2. Log of Creatine – Its paramcd is “L10CREAT” and its analysis values, AVAL, come from log of LB.LBSTRESN.

Since it is Creatine analysis from SDTM LB, the data set is named as ADLB.

ANOTATIONS ON MOCK UP TABLE

Table 14.4.1
Summary of table of Creatine at baseline
ADLB.AVISIT=’BASELINE’
(Per Protocol Population)
ADLB.PPROTFL=’Y’

ADLB.PARAMCD	**Group 1: Treatment 1 (N=xxx)** ADLB.TRTAN = 1		**Group 2: Placebo (N=xxx)** ADLB.TRTAN = 2	
	n	**Observed Mean Value**	**n**	**Observed Mean Value**
Creatine WHERE PARAMCD=’CREAT’	COUNT(ADLB.AVAL)	MEAN(ADLB.AVAL)	COUNT(ADLB.AVAL)	MEAN(ADLB.AVAL)
Log of Creatine WHERE PARAMCD=’L10CREAT’				

The red-colored are annotations on Mock Up tables.
The above annotations help SAS programmers to create metadata – analysis dataset metadata, analysis variable metadata, analysis parameter value-level metadata and analysis results metadata (This paper does not include analysis results metadata for the sample Mock Up tables).

ANALYSIS VARIABLE METADATA including ANALYSIS PARAMETER VALUE-LEVEL METADATA

Dataset Name	Parameter Identifier	Variable Name	Variable Label	Variable Type	Display Format	Codelist / Controlled Terms	Source/Derivation
Subject Identifier Variables							
ADLB	****ALL****	STUDYID	Study Identifier	Char	$12.		ADSL.STUDYID
ADLB	****ALL****	ADDOMAIN	Analysis Domain Abbreviation	Char	$8.	ADLB	Derived
ADLB	****ALL****	USUBJID	Unique Subject Identifier	Char	$20.		LB.USUBJID
ADLB	****ALL****	SUBJID	Subject Identifier for the Study	Char	$8.		ADSL.SUBJID
ADLB	****ALL****	SITEID	Study Site Identifier	Char	$10.		ADSL.SITEID
Treatment Variables							
ADLB	****ALL****	TRTA	Actual Treatment Group	Char	$20.		ADSL.TRTP
ADLB	****ALL****	TRTAN	Actual Treatment Number	Num	8.		ADSL.TRTPN
Timing Variables							
ADLB	****ALL****	AVISIT	Analysis Timepoint Description	Char	$50.	BASELINE VISIT 1	LB.VISIT
ADLB	****ALL****	AVISITN	Analysis Timepoint Number	Num	8.	0=BASELINE 1=VISIT 1	LB.VISITNUM
Analysis Parameter Variables							
ADLB	**CREAT**	PARAM	Parameter Description	Char	$100.	Creatine (mg / dL)	LB.LBTEST + unit
ADLB	**L10CREAT**	PARAM	Parameter Description	Char	$100.	Log10 of Creatine (mg / dL)	LB.LBTEST + unit
ADLB	****ALL****	PARAMCD	Parameter Code	Char	$8.	CREAT L10CREAT	LB.LBTESTCD
ADLB	**L10CREAT**	PARAMTYP	Parameter Type	Char	$8.	DERIVED	
ADLB	**CREAT**	AVAL	Analysis Value	Num	8.		LB.LBSTRESN
ADLB	**L10CREAT**	AVAL	Analysis Value	Num	8.		log(LB.LBSTRESN)
ADLB	****ALL****	BASE	Numeric Baseline Value	Num	8.		ADLB.AVAL at AVISIT=”BASELINE”
ADLB	****ALL****	CHG	Change from Baseline	Num	8.		ADLB.AVAL – ADLB.BASE
Indicator Variables							
ADLB	****ALL****	PPROTFL	Per Protocol Population Flag	Char	$1.	Y	ADSL.PPROTFL
ADLB	****ALL****	ABLFL	Baseline Flag	Char	$1.	Y	‘Y’ at ADLB.AVISIT=’BASELINE’
Data Point Traceability Variables							
ADLB	****ALL****	SRCDOM	Source Domain	Char	$8.	LB	
ADLB	****ALL****	SRCVAR	Source Variable	Char	$8.	LBSTRESN	
ADLB	****ALL****	SRCSEQ	Source Sequence Number	Num	8.		LB.LBSEQ

Analysis variable metadata including parameter value-level metadata could be created according to the annotations on the Mock Up tables.

If ADLB is used to generate more tables, analysis descriptor variables such as DTYPE, categorical variables such as CRIT1 and CRIT1FL, or indicator variables such as ANL1FL will be added to metadata.

ANALYSIS DATASET METADATA

Dataset Name	Dataset Description	Dataset Location	Dataset Structure	Key Variables of Dataset	Class of Dataset	Documentation
ADLB	Laboratoy analysis data	ADLB.xpt	one record per subject per parameter per analysis timepoint	USUBJID, PARAM, AVISIT	BDS	ADLB.SAS

The above analysis dataset metadata could be also created according to the annotations on the Mock Up tables.

Sample codes using ADaM BDS data set

The following codes can be used to derive the count and mean value of Creatine and Log of Creatine at baseline for treatment 1 and placebo.

```
       proc sql;
              **** The Count and Mean Value for Creatine and Log of Creatine for
              protocol population at Baseline;
              create table line1_1 as
                     select trtan, paramcd, count(aval) as count, mean(aval) as mean
                     from adlb
                     where paramcd in( ‘CREAT’, ‘L10CREAT’) and avisit = ‘BASELINE’
                            and pprotfl = ‘Y’ and aval is not missing
                     group by trtan, paramcd
              ;
       quit;
```

CONCLUSION

ADaM is structured as analysis-ready. In order to build analysis-ready ADaM data sets, SAS programmers need to start from analysis, which are Mock Up tables. From Mock Up tables, SAS programmers figure out what procedures could be used for the analysis. Based on the procedure statements and analyses, SAS programmers annotate the Mock Up tables. The annotations on the Mock Up tables will help SAS programmers to find out what variables are needed in the analyses and eventually what variables and parameters should be created in ADaM data sets. SAS programmers now can create Metadata according to the annotations. According to Metadata, SAS programmers can create ADaM data sets from SDTM data sets. From ADaM data sets, SAS programmers will be able to produce all the results in the Mock Up tables using one proc procedure.

REFERENCES

The ADaM Implementation Guide, Version V 1.0 (ADaMIG v1.0)
The Analysis Data Model, Version 2.1 (ADaM 2.1)

CONTACT INFORMATION

Your comments and questions are valued and welcomed. Please contact the author at

Kevin Lee
Cytel, Inc.
Chesterbrook, PA
(610) 994 - 9840
Email:Kevin.lee@cytel.com

TRADEMARKS