



Le « b-a-ba » du standard ADaM, mise en pratique

GUF CDISC – Paris – 2025-06-24

Thierry Lambert



Plan

2

- Contenu d'une soumission
- ADaM (Analysis Data Model)
- Les différents types de fichiers
- ADRG
- Le problème de la cohérence de l'information



Le contenu d'une soumission

3

- SDTM
 - acrf.pdf : Annotated CRF
 - *.xpt
 - define.xml
 - csdrg.pdf : Reviewer's Guide
- ADaM
 - *.xpt
 - define.xml
 - adrg.pdf : Reviewer's Guide
 - (programmes)



ADaM: les Tables de la Loi

4

- Analysis Data Model (ADaM) 2.1 (2009)
- ADaM Implementation Guide 1.1-1.3 (2016-2021)
- ADaM Occurrence Data Structure 1.1 (2021)
- ADaMIG for Medical Devices 1.0 (2021)
- ADaMIG for Non-compartmental Analysis Input Data 1.0 (2021)
- BDS for ADaM popPK Implementation 1.0 (2023)
- ADaM Conformance Rules v5.0 (2023)
- ADaM Metadata Submission Guidelines 1.0 (2023)
- ADaM Examples of Traceability 1.0 (2022)
- ADaM BDS for Time-to-Event Analyses 1.0 (2012)
- ADaM Examples in Commonly Used Statistical Analysis Methods (2011)
- CDISC Define-XML Specification 2.1 (2019-2025, 2.1.0-2.1.9)
- ADRG 1.2 (2019)



Principes ADaM

5

- Les fichiers ADaM *doivent* être issus des fichiers SDTM.
- Ils doivent être “one proc away” des résultats soumis (tables, graphiques, listings).
- La traçabilité du SDTM aux ADaM doit être parfaite.



Zoom sur la partie ADaM

6

- AD*.pxt
 - fichiers d'analyse au format transport SAS v5
 - ADSL requis
 - autres fichiers: structure BDS ou OCCDS autant que possible
- define.xml
 - version 2.1.x (actuellement, x = 9)
 - <https://www.cdisc.org/standards/data-exchange/define-xml>
- adrg.pdf
 - suivant le modèle posté sur le wiki PhUSE
 - <https://advance.hub.phuse.global/wiki/spaces/WEL/pages/26804660/Analysis+Data+Reviewer+s+Guide+ADRG+Package>



Conventions de noms

7

- Fichiers: AD...
 - ADSL, ADVS, ADAE, ADTTE, etc.
- Variables
 - Nom SDTM => même contenu
 - mais la réciproque est fausse : on renomme les variables SDTM inchangées si leur nom ADaM est différent, par exemple VISITNUM/VISIT > AVISITN/AVISIT
 - mais piège classique : AVALC n'est pas fait pour xxSTRESC
 - dates: *DTC, *DT, *DTM, *TM, *DTF, *TMF
 - flags: *FL (char) et *FN (num)
 - ... (cf. ADaM IG)



3 types de fichiers

8

- ADSL
 - obligatoire
 - un enregistrement par sujet
- BDS
 - un enregistrement par paramètre et “time point”
- OCCDS
 - un enregistrement par occurrence d’événement ou intervention
 - adverse events, procédures
 - concomitant medications, exposure



ADSL

9

- obligatoire
- un enregistrement par sujet, quel que soit le dessin de l'étude
- variables "core": SUBJID, SEX, RACE, populations
- treatment variables: TRTxxP, TRTxxA, TRTSEQP, TRTSEQA...
- trial dates: TRTSDT(M), TRTEDT(M), TRxxSDT(M), TRxxEDT(M), APxxSDT(M), APxxEDT(M)...
- variables principales efficacy et safety, baselines...



BDS

10

- un enregistrement par paramètre et “time point”
- paramètre: PARAMCD, PARAM
 - “pression artérielle systolique assise” = 1 paramètre, pas le droit aux “by-variables” => PARCAT1 = systolique/diastolique, PARCAT2 = assis/debout
- timepoint: AVISITN, AVISIT, ATPTN, ATPT, ADTC/ADT(M), APERIOD, APHASE
- AVAL (valeur), BASE (baseline), CHG, PCHG
- CRITy, CRITyFL, MCRITy, MCRITyML
- SHIFTy (changements depuis baseline pour AVAL, NRIND, TOXGR...)
- windowing: AWLO, AWHI, AWTDIF...
- time to event: STARTDT, CNSR, EVNTDESC, CNSDTDSC
- flags: ABLFL, ANLzzFL, PREFL, ONTRTFL, FUPFL

BDS : une légende urbaine

11

- “On a le droit de rajouter n’importe quelle variable dans un fichier ADaM”
- Pourquoi c’est faux :
 - ADaM IG 1.1-1.3, section 4.2.1, Rules for the Creation of Rows and Columns
 - 6 règles
 - seule la première permet de rajouter des colonnes (nouvelles variables)
 - toutes les autres règles => ajouter des lignes
 - un exemple classique : on veut, pour un paramètre (valeur de PARAM) mesuré au cours du temps (AVISIT) la différence d’une visite à l’autre
 - ça ne peut pas être une colonne supplémentaire
 - soit nouveau paramètre (règle 4 : *A function of multiple rows within a parameter should be added as a new parameter*)
 - soit BASETYPE (règle 6 : *When there is more than one definition of baseline, each additional definition of baseline requires the creation of its own set of rows*)



OCCDS

12

- conventions de noms pour des fichiers d'analyse AE et CM
- en général proches des fichiers SDTM correspondants



Les autres livrables

13

- define.xml
- Analysis Data Reviewer's Guide (ADRG)
- Analysis Result Metadata



ADRG

14

- 1. Introduction
 - 1.1 Purpose
 - 1.2 Acronyms
 - 1.3 Study Data Standards and Dictionary Inventory
 - 1.4 Source Data Used for Analysis Dataset Creation
- 2. Protocol Description
 - 2.1 Protocol Number and Title
 - 2.2 Protocol Design in Relation to ADaM Concepts
- 3. Analysis Considerations Related to Multiple Analysis Datasets
 - 3.1 Comparison of SDTM and ADaM Content
 - 3.2 Core Variables
 - 3.3 Treatment Variables
 - 3.4 Subject Issues that Require Special Analysis Rules
 - 3.5 Use of Visit Windowing, Unscheduled Visits, and Record Selection
 - 3.6 Imputation/Derivation Methods
- 4. Analysis Data Creation and Processing Issues
 - 4.1 Split Datasets
 - 4.2 Data Dependencies
 - 4.3 Intermediate Datasets
 - 4.4 Variable Conventions
- 5. Analysis Dataset Descriptions
 - 5.1 Overview
 - 5.2 Analysis Datasets
 - 5.2.1 ADSL – Subject Level Analysis Dataset
 - 5.2.x Dataset – Dataset Label
- 6. Data Conformance Summary
 - 6.1 Conformance Inputs
 - 6.2 Issues Summary
- 7. Submission of Programs
- 8. Appendix



Validation P21

15

- Doit être faite
- Ne doit donner aucun message (sauf bug avéré de P21)

- L'information est répliquée à de nombreux endroits:
 - type, nom, label, longueur des variables : *programme, define.xml*
 - dataset dependencies : *programmes, ADRG*
 - core variables : *programmes, ADRG, define.xml*
 - variables copiées : *SDTM datasets, SDTM define.xml, programmes, ADaM datasets, ADaM define.xml*
 - clés : *programmes, define.xml*
 - structure (one record per...) : *define.xml, ADRG*
 - explication des dérivations : *programmes, define.xml, ADRG*
- Résultat
 - souvent define = spécifications de la programmation
 - mais logique générale: ADRG
 - les programmes évoluent, define/ADRG mal mis à jour



Une solution

17

- Centraliser le maximum dans les programmes, en dériver define.xml et plusieurs parties de l'ADRG



Exemple (1)

18

```
%declare_adam_dataset(  
  name = ADVS,  
  label = "Vital Signs Analysis Dataset",  
  key = USUBJID PARAMN AVISITN,  
  structure = %str(subject, parameter and visit), /* one record per... */  
  class = BDS,  
  analyses = safety  
)
```

```
proc format lib=adam;  
  %triple_value( $VS_PARAM, /* Vital Signs Parameters */  
    1 'HEIGHT' = 'Height (cm)'  
    2 'WEIGHT' = 'Weight (kg)'  
    (...)  
  )  
  value VS_SHIFT /* Vital Signs Range Indicator Shifts from Baseline */  
    11 = 'Low to Low'  
    12 = 'Low to Normal'  
    (...)  
  ;  
run;
```

```
/*  
<markdown>
```

This dataset contains the safety results derived from the SDTM VS domain.

Creation of the dataset

The dataset is created by selecting VS records belonging to the safety population (SAFFL=Y).

The essential ADaM variables (AVAL, BASE, CHG) are then derived as explained in the define.xml file. The normal range indicators are derived after assigning the following normal ranges:

Age	Parameter	Low	High
6-11	Systolic BP (mmHg)	97	120
	Diastolic BP (mmHg)	57	80
(...)			

```
</markdown>  
*/
```



Exemple (2)

19

```
data ADVS;
  merge sdtm.VS (in=inVS) adam.ADSL (in=inADSL keep=USUBJID SUBJID AGE);
  by USUBJID;
  (...);
run;

(...)

data adam.ADVS;
  %attrib_of(adam.ADSL, SUBJID)
  attrib
    PARAM    length=$40 label="Parameter"          /* format=$VS_PARAM.Label */
    /* Parameter Name */
    PARAMCD   length=$8  label="Parameter Code"     /* format=$VS_PARAM. */
    /* Parameter Code */
    PARAMN    length=8   label="Parameter (N)"      /* format=VS_PARAM. */
    /* Parameter Numeric Code */
    AVISIT    length=$40 label="Analysis Visit"      /* format=VISIT.Label */
    /* VS.VISIT */
    AVISITN   length=8   label="Analysis Visit (N)"  /* format=VISIT. */
    /* VS.VISITNUM */
  ;
  (...);
  merge ADVS (in=inVS) adam.ADSL (in=inADSL);
  by SUBJID;
  (...);
run;

%finalize_adam_dataset
```



Questions ?

20