PharmaSUG 2019 - Paper DS-55

Achieving Zen: A Journey to ADaM Compliance

Kirsty Lauderdale, Kjersten Offenbecker, and Alice Ehmann, Clinical Solutions Group

Abstract

For many programmers and statistician creating compliant ADaM specifications, programs and data sets is confusing and a bit overwhelming. What should be included versus what to leave out? What level of traceability is needed? What information should be presented in the specs, and what efficiencies can be utilized in the code? Follow us as we work through some of the common pitfalls and map out a path which will help you navigate this winding road to come out with a more compliant product which is clearer for everyone to follow and understand. We will shine light on how to create compliant specifications which will lead you to compliant data sets that are everything the FDA is looking for.

Introduction

According to CDISC[1], ADaM is defined as dataset and metadata standards that support efficient generation, replication and review of clinical trial statistical analyses; and traceability among analysis results, analysis data and data represented in the SDTM.

Unlike SDTM where the data set and metadata standards are very clearly defined, with specific requirements for all standard domains as well as guidelines outlining custom domains and how to deal with adherent data, ADaM standards are a bit more open to interpretation.

This is an intentional approach, since analysis data tends differ from study to study and does not conform to a one size fits all model. In fact, there are only three data structures defined in the ADaM documentations[2]; the Subject-Level Analysis Data set (ADSL), Basic Data Structures (BDS) and Occurrence Data Structures (OCCDS). The “ADaM Implementation Guide” (ADaM-IG v1.1) defines the ADSL and BDS portions and the “ADaM Structure for Occurrence Data (OCCDS) v1.0” covers the OCCDS.

The expectation is all ADaM data will fit into one of these three defined structures and follow the basic guidelines outlined. That sounds easy enough, right?

Unfortunately it’s not always as straight-forward as we would like, so much so that the US Food & Drug Administration (FDA) released a “Study Data Technical Conformance Guide”[3] and they specifically highlight the following categories of interest; data set labels, subject-level analysis data, core variables, key efficacy and safety variables, timing variables, numeric date variables, and imputed data.

The above noted items are referenced below, although grouped slightly differently and we have included a section on study specification requirements versus define details. In the following sections we will highlight the areas where we have seen the most compliance issues.

Specifications vs. define.xml

Study specifications are typically used to tell the programmers what they need to do, while the define.xml tells the FDA what was done. Although these documents sound like they should be identical, don't let that fool you!

Area of Turmoil	Zen State
Define.xml contains too much SAS code/pseudo-code Here is a typical specification example for TRT01P: if ARMCD='A' and VISITNUM=100 then TRT01P='Treatment A'; if ARMCD='B' and VISITNUM=100 then TRT01P='Treatment B'; Often what is in the define.xml is simply a copy of the derivation from the Specification.	The expectation is that what appears in the define.xml should primarily be text with minimal code. Add a column to the specifications document for the define.xml text and populate this during spec development, so it is ready at the time of eCRT creation.
Keys are inaccurate --SEQ is not a value-added key. The keys do not define how to identify unique records; ADaM Guidance states – "ideally uniquely identifies and indexes each record in the data set".	Be specific with the variables listed as part of the key Aim to make the key unique but do not include AVAL in your key, even though some ADaM can't be made unique without using AVAL or something similar.
Inaccurate length values Length is defined within the specs rather than being programmatically populated from the data.	Do not include Length as part of your specifications, as length changes due to the need to truncate them in the final step. Populate this information in the define file by using a programmatic process for determining the actual lengths.
ADaM dataset labels are identical to SDTM dataset labels Each dataset should have its own unique internal label, both SDTM.AE and ADaM.ADAE should not both have a label of "Adverse Events".	Ensure datasets have unique labels, and that labels are not stripped during truncation portion.
Variables directly copied from SDTM Variables that are copied directly from SDTM must match what is in SDTM exactly. Do not change the attributes or the values. If changes are needed you should create a new variable.	Simply copy SDTM variables into ADaM and make no changes.
Numeric variables used for sorting on displays It is vital that there is a one-to-one correlation between the numeric and character values. When integrating studies ensure this item is closely monitored.	Create unique pairings and harmonize as needed.

Unnecessary ADaM data sets ADaM data sets are needed to support Tables and Figures, not Listings. Listing can be created directly from SDTM, unless a variable needs to be derived. Do not create an ADaM if it's only use is to produce Listings (i.e. ADPE, ADIE).	Use SDTM to create Listings, where possible, and only create ADaM data sets to support the Tables and Figures.
Missing necessary variables for analysis One PROC away is intended to ensure that all analysis variables are present in ADaM and not derived within TLF code. If the variable is created in your TLF code, how do you explain to a reviewer the derivation, its intent and how do you ensure consistent updates.	Ensure all variables needed for analysis are present in your ADaM data sets. Do not create any new variables within your TLF code.

Core Variables

These variables, which are necessary to analyze the data, should be present in each ADaM dataset, and are typically already represented in the ADSL dataset.

Area of Turmoil	Zen State
CORE variables The purpose of the CORE variables is not to include every demographic and baseline variable. Instead it should be a list of variables that are needed in all the other domains (or a majority of them). Typically we would expect to see variables that represent study/protocol numbers, site numbers, subject identifiers, treatment assignment info, demographic analysis qualifiers, and analysis population flags. However additional identifiers such as ARM/ARMCD should not be included in your list of core variables.	Look at your ADSL variables in conjunction with your TLFs and decide exactly what is needed across the data sets. Do you really need WEIGHTBL in all your data sets, or SEX?, are you doing a gender analysis? If not, don't include them. Use the ADaM treatment variables across data sets (TRTSEQP, TRT01A, TRT01AN).

ADSL

ADSL is a subject-level dataset, consisting of one record per subject containing variables needed for analysis; such as population flags, planned and actual treatment variables, demographic information, important dates, etc.

Area of Turmoil	Zen State
Baseline values *xx*BLFL typically flags the last observation prior to dosing but sometimes the definition of an Analysis Baseline is far more complicated. If it is different than SDTM be sure your metadata is clear.	Double check the SAP to see how baseline is defined for your study. Don't simply assume it is the value where *xx*BLFL='Y'.

Population flags Just because a population flag is defined in the IG does not mean you have to produce it in your ADSL. Are you really summarizing the Per Protocol Population (PPROTFL)? If not, then exclude it. If you need to add a population flag that is not defined in the IG be sure to follow the guidance. Subject-level flags should be populated with a value; Y or N, they cannot be null. Parameter-level and record-level flags can be null. Don't define two flags when one flag can easily be used; do you really need COMPFL (completed) and DISCFL (discontinued)? Either you completed, or you didn't! Do you have two flags that are defined exactly the same? RANDFL and ITTFL? Define and keep one.	Create only what you need Add study specific flags using the guidance as your roadmap Be sure to populated subject-level and parameter-level flags correctly. Create only what you need, don't be redundant.
Treatment variables ARM/ARMCD should not be included in your list of core variables. No need to create a numeric representation of ARM/ARMCD (i.e. ARMN). Do not use ARM to group treatment variables (i.e. ARMGR1). ARM is not an analysis variable and should not be used for grouping. Treatment sequencing should be in the order of treatments given; TRT01P is the first treatment, TRT02P is the second treatment, and so on. The associated dates should be in sequential order as well (i.e. TRT01SDT, TRT02SDT).	Use the ADaM treatment variables across data sets (TRTSEQP, TRT01A, TRT01AN). Create only what you need, don't add numeric variables just to do it. Use the analysis treatments for grouping (i.e. TRTGR1). Ensure consistency and accuracy when assigning treatment pairings.
Date/time variables Variable names are very specific in the ADaM IG and they are different from SDTM. For instance, TRTSTDT and TRTENDT rather than TRTSDT and TRTEDT. Dates should be stored in ISO 8601 format, not date9.	Follow the guidance exactly when creating date/time variables. Date formats should be consistent with CDISC standards.
Variable naming conventions Baseline variables start or end with BL (i.e. WEIGHTBL, ABLFL). Flag variables end in FL (i.e. COMPFL).	Follow ADaM IG naming conventions to ensure reviewers understand your intent.

Basic data structure data sets (BDS)

A BDS contains one or more records per subject, per analysis parameter, per analysis timepoint.

Area of Turmoil	Zen State
Creating surplus variables AVALU is not a valid ADaM variable; if the unit is relevant it should be included in PARAM, if not, just keep ORRESU/STRESU.	Create what we need but not more than we need and don't create variables just because, they should have a specific purpose.
ADLB – PARAM/PARAMCD LBTEST/LBTESTCD is typically not sufficient to create PARAM/ PARAMCD given the fact that due to the type of test (chemistry/hematology/urinalysis) they are not unique. PARAM/ PARAMCD cannot have two different PARCAT1 values. Capturing LBCAT as part of PARAM is a great solution to ensuring uniqueness, but should only be added for those values that are not unique without it.	Ensure that how PARAM/PARAMCD is created ensures unique records can be identified easily.
BASETYPE BASETYPE should be populated for all values within a PARAM.	When using BASETYPE ensure it is populated for all values within a parameter.
BASE and CHG These variables go hand-in-hand, if you are not doing a change from baseline analysis then BASE is not needed, baseline values can be easily identified using AVISIT and/or ABLFL. BASE must be associated with ABLFL not AVISITN	Think about the needed analysis, does it include a change from baseline set of TLFs. If not, then don't create BASE just to create something new.
AVALCATy –vs- CRITy AVALCATy should be a categorization of AVAL/AVALC, but often is used to define criteria evaluations.	AVALCATy is used to categorize values within a Parameter. CRITy is used to depict criteria defined to group results based on the collected value's relationship to one+ algorithmic conditions.
DTYPE This variable is used to identify records, within a given parameter that contain these special-case analysis values, and describes/traces the derivation. Three common situations when DTYPE should be populated: 1. A new row is added within a parameter with the analysis value populated based on other rows within the parameter. 2. A new row is added within a parameter with the analysis value populated based on a constant value or data from other subjects. 3. An analysis value (AVAL/AVALC) on an existing record is being replaced with a value based on a pre-specified algorithm. DTYPE is not used to identify records across all parameters like SUM or EOT.	Clearly understand how DTYPE is intended and use appropriately.

Occurrence Data Structure data sets (OCCDS)

Occurrence analysis is the counting of subjects with a given record or term, and often includes a structured hierarchy of dictionary coding categories. Some examples of data that fit into this structure include those used for Adverse Events, Concomitant Medications, and Medical History.

Area of Turmoil	Zen State
Adding records for subjects with no values Do not create records where AETERM, CMTRT, etc. = None or are null.	Do not create records you do not need. If you need a record to show that a subject had no Adverse Events or received no medications this can be done in the TLF code easily.
Imputed dates If you impute the date you must include the imputations flags.	Include the appropriate Imputation Flags when dates are imputed. Ensure imputation is correctly defined in the define.xml/ADRG.

Conclusion

The point of ADaM is not to include every possible piece of data to cover every possible scenario; instead it is to include what is needed to produce the planned analysis. It can be hard for programmers and statisticians to think this way, we want to plan for the unknown - but that is not what ADaM is intended for. Remember your file is being sent to the FDA, everything you send can be analyzed and checked, do you want to get comments back on data that are not relevant to the study analysis?

Follow these four simple rules and you won't need to remember to do your meditative breathing;

- Do you have everything you need to produce the TLFs?
- Don't create variables in your TLF code.
- Do you have information that is never used anywhere and adds no value?
- Will the FDA need to see the information, or do you want them to review it if you did not analyze it?

Contact Information

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

Kirsty Lauderdale
Clinical Solutions Group
klauderdale@csg-inc.com

Kjersten Offenbecker
Clinical Solutions Group
koffenbecker@hotmail.com

Alice Ehmann
Clinical Solutions Group
aehmann@csg-inc.com

References

[1] CDISC homepage: https://www.cdisc.org/
[2] ADaM Standards Download Area: https://www.cdisc.org/standards/foundational/adam
[3] Study Data Technical Conformance Guide – Jan2019: https://www.fda.gov/downloads/ForIndustry/DataStandards/StudyDataStandards/UCM624623.pdf