

## Clarifications About ADaM Implementation Provided in ADaMIG Version 1.1

John Troxell, Accenture

### ABSTRACT

Since the publication of version 1.0 of the CDISC (Clinical Data Interchange Standards Consortium) ADaM (Analysis Data Model) Implementation Guide (ADaMIG) in December 2009, experience has shown that there are some misunderstandings among practitioners about ADaM implementation. The ADaM Team realized that the best way to address these misunderstandings was to clarify the ADaMIG. In fact, although additional variables were added in version 1.1, published in 2016, the primary motivation was to clarify version 1.0. This paper describes important clarifications contained in version 1.1, and what they mean for implementers of both ADaMIG versions.

### INTRODUCTION

Sir Karl Raimund Popper, a great philosopher of science, said:

*“It is impossible to speak in such a way that you cannot be misunderstood.”*

While Sir Popper may seem to have provided an excuse for misunderstandings out there, the ADaM Team is still at least partially responsible. The ADaM Team tries to prevent any misunderstanding by carefully crafting our documents. We spend hours and hours refining what we say. Others will never fully appreciate how much time is spent on this task. Even variable names can be the subject of hours of discussion. Nevertheless, we are not perfect in communicating what we are thinking. Moreover, sometimes we realize that the problem is not so much that we didn't express something well enough; sometimes our thinking was not clear enough or in sufficient depth to begin with. Even among the authors of ADaMIG1.0, there can be differences in understanding. The task of clarification involved much more than mere wordsmithing.

This paper highlights some of the more important clarifications the ADaM Team made in ADaMIG1.1. These clarifications should be taken to heart by implementers of both versions.

### OPINION: ADAM HAS TO BE UNDERSTOOD ON ITS OWN TERMS

Before delving into specific clarifications in ADaMIG1.1, it is good to take a step back for some perspectives on ADaM that might be useful when trying to understand ADaM and the ADaMIG. These perspectives are the author's opinion, and are not part of the ADaMIG. The author believes that lack of these perspectives is one of the reasons that some of the clarifications in ADaMIG1.1 proved necessary.

### EVEN IF CDISC WENT AWAY, ADAM WOULD STILL BE USEFUL

As a data standard for analysis datasets, ADaM could exist quite well without SDTM. ADaM fundamental principles and data structures are solutions to problems that are independent of the format of the tabulation. Although the ADaMIG does state that compliant ADaM presumes that ADaM is derived from SDTM, it would take a minimal effort to refit ADaM to work with any other data tabulation, and the underlying ADaM data structures would not need to change. ADaM is a solution to universal problems of statistical analysis of clinical data.

### DATA ARE NOT PUSHED TO ADAM, THEY ARE PULLED FROM SDTM AS NEEDED

Much of clinical data management and CDISC data flow involves mapping from collected data to SDTM, in which data are pushed forward from one place to another. There is a natural tendency to assume that the next step is also essentially a push-forward operation, and that what is required is to push all of the data from SDTM into yet another destination, called ADaM. However, there is nothing in the ADaM standard that requires or assumes that all data from SDTM be pushed into ADaM. ADaM was not developed to be yet another destination for the collected data.

ADaM datasets are created to meet specific analysis needs. If there is no analysis need, there is no need for an analysis dataset to support it. On the other hand, there is no prohibition against bringing into ADaM datasets all data in SDTM. One cannot design an ADaM dataset based solely on what is in SDTM. One must also know the analysis need or needs that must be supported by the ADaM dataset. With the analysis in mind, it is possible to select the appropriate generic standard ADaM data structure that is best suited to meet the analysis need, and tailor it to meet that need; or if no generic standard structure meets the need, then design a new structure. Data are pulled from SDTM as needed, to support needed data derivations, and enable analysis, traceability, and review.

ADaM data structures are in general content-neutral; they are not about blood pressures, they are general-purpose containers. They are tools for solving analysis problems.

Some of the most common misunderstandings about ADaM are rooted in the unfortunate tendency to believe that creation of ADaM is simply a sterile push-forward mapping exercise from SDTM.

## **UNDERSTANDING SDTM CAN INSPIRE FALSE CONFIDENCE**

ADaM must be understood from the ground up, on its own terms, as a solution for analysis needs. If not, a natural shortcut is to try to view ADaM through SDTM-tinted goggles. Often, people who have learned SDTM see some superficial similarities and think they just need to rename some SDTM variables. That way lies disaster.

## **THE OVERALL THEME OF CLARIFICATION IN ADAMIG VERSION 1.1**

New with ADaMIG1.1 is Appendix B, “Revision History”. One can begin to get a sense that clarification was a main goal by noting that of the 110 revisions listed, 38 include one or more of the words clarification, clarity, clarify, or clarified. But that is just the start. Examples were added, sections were added, and language was improved, as noted in Appendix B and otherwise, throughout the document. In many cases, these changes also contribute to the goal of clarification.

What follows is a selection of only some of the highlights.

## **SELECTED CLARIFICATIONS IN ADAMIG VERSION 1.1**

### **ANALYSIS DATASETS AND ADAM DATASETS**

There has been some confusion and abuse of vocabulary about when an analysis dataset is an ADaM dataset and when not.

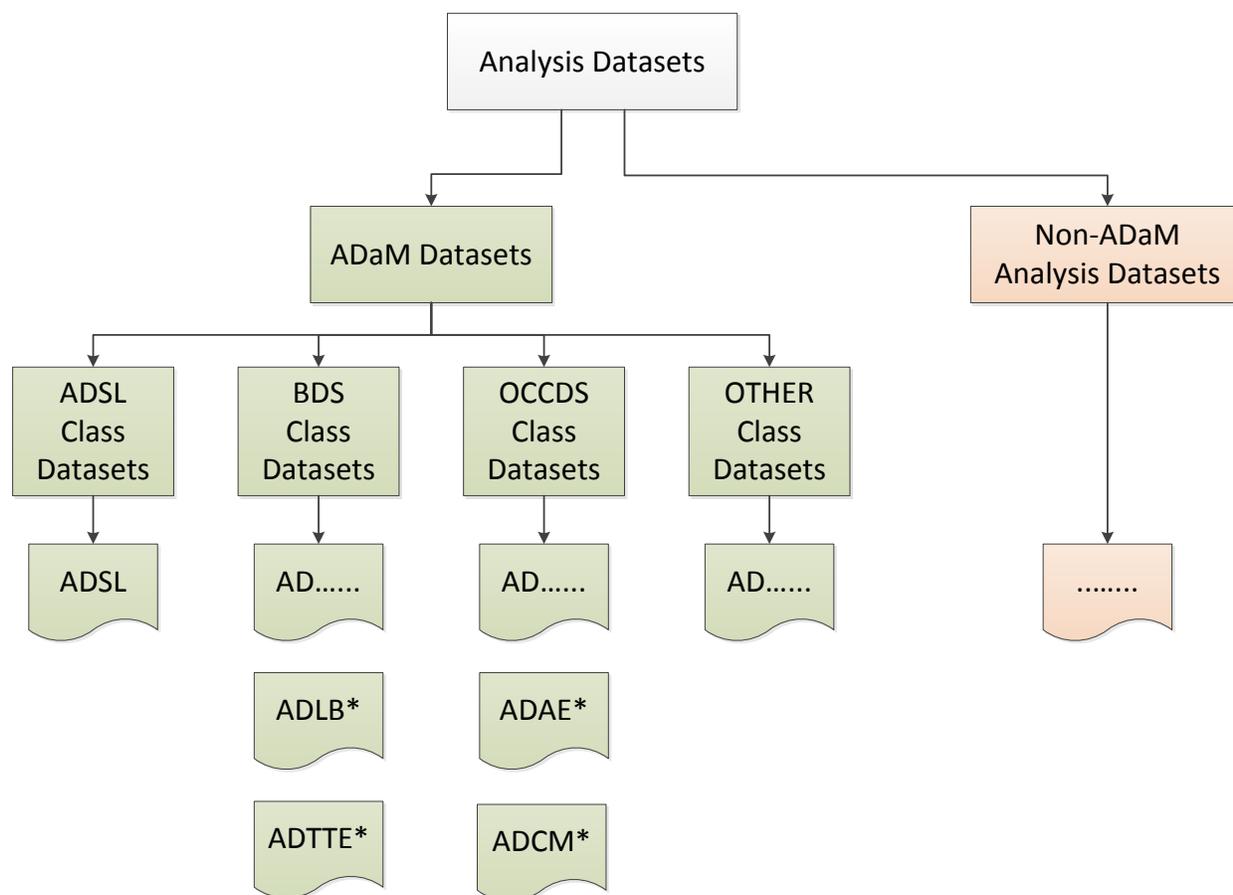
Section 1.6 of ADaMIG1.1 was added to clarify the distinction. Paraphrasing, the distinction involves whether or not the ADaM fundamental principles and naming conventions are adhered to.

Throughout ADaMIG 1.1, the words “analysis dataset” have been replaced by the more specific “ADaM dataset” when referring to an ADaM dataset.

ADaMIG1.1, however, does not address when it is appropriate to create an ADaM dataset that does not use one of the ADaM standard data structures (ADSL, BDS, and OCCDS) defined to date. The paper “What is the “ADAM OTHER” Class of Datasets, and When Should it be Used?” (Troxell 2015) provides advice about that important topic.

These topics are more relevant to ADaM than to SDTM, because ADaM does not have subject-matter-related domains. Except for ADSL, ADaM does not even have standard datasets. Instead, it has standard dataset *structures* and fundamental principles.

Figure 1 is a slightly modified version of a diagram from the 2015 paper:



**Figure 1: The Universe of Analysis Datasets**

In the figure, dots in a dataset name represent characters that are specified in an actual dataset name. An asterisk after a dataset name means that it is an example name of a dataset of the class.

### PARAM, AVAL AND AVALC

There has been some confusion about the purpose and use of the BDS variables PARAM (Parameter), AVAL (Analysis Value), and AVALC (Analysis Value (C)). People used to SDTM, and who think of ADaM as a mapping exercise from SDTM, are particularly prone to misunderstanding and misusing these variables.

Section 3.3.4 of ADaMIG1.1 includes several new paragraphs whose goal is to clarify these variables and their relationships. Paraphrasing, the paragraphs state that:

- In contrast to SDTM Findings --TEST, ADaM BDS PARAM has no qualifiers.
- PARAM is created to meet an analysis need, not just because something is collected.
- A given PARAM may be highly derived data from several inputs, including any combination of SDTM domains across multiple classes and any ADaM datasets.
- PARAM describes what is in AVAL or AVALC.
- For most parameters, either AVAL or AVALC will be populated, not both.
- AVAL and AVALC are both populated only when AVAL and AVALC are a 1:1 map in the parameter on all rows on which both are populated. Two examples of when it is appropriate to populate both are described.

To further assist in this clarification, CDISC Core has now been specified separately for AVAL and AVALC, and has changed from Req (required) to Cond (conditionally required) for both; also, it has been specified that at least one of the two must be present in the dataset.

The paper “A Guide to the ADaM Basic Data Structure for Dataset Designers” (Barrick and Troxell 2014) discusses the use of these and other variables, including AVISIT, DTYPE, BASETYPE, PARAMTYP and PARCATy.

## **PARCATy IS NOT A QUALIFIER OF PARAM**

As mentioned, ADaMIG1.0 clearly states that PARAM has no qualifiers.

Nevertheless, there has been considerable misuse of the variable PARCATy (Parameter Category y) as a qualifier of PARAM.

This confusion stems in large part, the author believes, from the fact that in SDTM, --CAT can vary within --TEST. Once again, it seems as if SDTM-tinted goggles have lead some users astray.

In ADaMIG1.1, language has been added to the CDISC Notes for PARCATy to clarify that it is not a qualifier of PARAM, but rather a grouping of PARAMs; PARAM to PARCATy is a many-to-one map.

In retrospect, it might have saved a lot of pain for SDTM goggle-wearers if PARCATy had been named PARGRy or PARAMGy. In fact, in ADaM, there is no real reason to carry around two different fragments, for category and group. Group alone would have sufficed. That there are two is an unfortunate accident of history and language usage in clinical biostatistics.

## **SCOPE OF STATEMENTS**

In the text of ADaMIG1.0, many statements in CDISC Notes and elsewhere were made without a clearly defined scope of applicability. Thus, for example, there has been confusion about whether statements applied always, or within a study, or within a dataset, or to a particular parameter within a dataset, or only to certain rows within a parameter.

Importantly, throughout ADaMIG1.0, a great deal of attention has been paid to clarifying the precise scope of statements.

## **SOME OTHER CLARIFICATIONS**

Here is a partial list of some other changes that improve clarity:

- Added examples
- Added tables of variable name fragments
- Added example diagram of use of Phase, Period and Subperiod
- Added text to clarify DTYPE and visit windowing variables
- Created new section about variables useful for datapoint traceability
- Clarified text regarding parameter invariance.
- Clarified wording about ADaM data sources
- Added new section about size of ADaM datasets
- Added new section about traceability when the multiple imputation method is used
- Added new section about copying values onto a new record

## **CONCLUSION**

In the author’s opinion, ADaM is best understood on its own terms, and not merely as a destination into which to push the SDTM data. Version 1.1 of the CDISC ADaM Implementation Guide contains many clarifications that are relevant to users of both versions of the ADaMIG.

## REFERENCES

M. Barrick and J. Troxell. 2014. “A Guide to the ADaM Basic Data Structure for Dataset Designers”. Proceedings of the PharmaSUG Conference.

J. Troxell. 2015. “What is the “ADAM OTHER” Class of Datasets, and When Should it be Used?”. Proceedings of the PharmaSUG Conference.

## ACKNOWLEDGMENTS

The entire CDISC ADaM Team deserves praise for their efforts over many years in developing ADaMIG Versions 1.0 and 1.1, as well as the other ADaM documents.

Relevant to the particular subject of clarifications in ADaMIG1.1, many people contributed. The contributions of Cathy Barrows deserve special mention.

## CONTACT INFORMATION

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

John Troxell  
Accenture  
908-612-1991  
[john.troxell@accenture.com](mailto:john.troxell@accenture.com)