Analysis Data Model Implementation Guide

Version 1.1

Prepared by the

CDISC Analysis Data Model Team

Notes to Readers

This Implementation Guide is Version 1.1 and corresponds to Version 2.1 of the CDISC Analysis Data Model.

Revision History

Date	Version	Summary of Changes
2016-02-12	1.1 Final	Released version reflecting all changes and corrections identified during comment period.
2014-05-23	1.1 Draft	Draft for Public Comment
2009-12-17	1.0 Final	

Note: Please see Appendix C for Representations and Warranties; Limitations of Liability, and Disclaimers.

CONTENTS

1 Introduction

1.1 Purpose

This document comprises the Clinical Data Interchange Standards Consortium (CDISC) Version 1.1 of the Analysis Data Model Implementation Guide (ADaMIG), which has been prepared by the Analysis Data Model (ADaM) Team of CDISC. The ADaMIG specifies ADaM standard dataset structures and variables, including naming conventions. It also specifies standard solutions to implementation issues.

The ADaMIG must be used in close concert with the current version of the document titled “Analysis Data Model (ADaM)” which is available for download at http://www.cdisc.org/adam. That document explains the purpose of the Analysis Data Model. It describes fundamental principles that apply to all analysis datasets, with the driving principle being that the design of ADaM datasets and associated metadata facilitate explicit communication of the content of, input to, and purpose of submitted ADaM datasets. The Analysis Data Model supports efficient generation, replication, and review of analysis results.

Note that in the remainder of the ADaMIG, the document titled “Analysis Data Model (ADaM)” is referred to as the ADaM model document.

1.2 Background

Readers of this implementation guide should be familiar with the CDISC Study Data Tabulation Model (SDTM) and the Study Data Tabulation Model Implementation Guide (SDTMIG), both of which are available at http://www.cdisc.org/sdtm, since SDTM is the source for ADaM data.

Both the SDTM and ADaM standards were designed to support submission to a regulatory agency such as the United States Food and Drug Administration (FDA). Since inception, the CDISC ADaM Team has been encouraged and informed by FDA statistical and medical reviewers who participate in ADaM meetings as observers, and who have participated in CDISC-FDA pilots. The origin of the fundamental principles of ADaM is the need for transparency of communication with and scientifically valid review by regulatory agencies. The ADaM standard has been developed to meet the needs of the FDA and industry. ADaM is applicable to a wide range of drug development activities in addition to FDA regulatory submissions. It provides a standard for transferring datasets between sponsors and contract research organizations (CROs), development partners, and independent data monitoring committees. As adoption of the ADaM model becomes more widespread, the use of this common model will support more efficient data-sharing among pharmaceutical sponsors, contract research organizations, and any partners involved in in–licensing, out–licensing, or mergers.

In addition, readers of the ADaMIG should be aware of information provided by the FDA. Specifically, the FDA website has a central location for the posting of FDA regulations and guidance documents that relate to data standards. The main page, entitled ‘Study Data Standards Resources’ contains links to important documents, both published and draft, for CDER, CBER, and CDRH (http://www.fda.gov/forindustry/datastandards/studydatastandards/default.htm). One such document is the Study Data Technical Conformance Guide, which is referred to in this implementation guide as the FDA TCG.

1.3 What Is Covered in the ADaMIG

This document describes two ADaM standard data structures: the Subject-Level Analysis Dataset (ADSL) and the Basic Data Structure (BDS). The third ADaM standard data structure, the Occurrence Data Structure (OCCDS), is described in the document “ADaM Structure for Occurrence Data (OCCDS) v1.0”.

The ADSL dataset contains one record per subject. It contains variables such as subject-level population flags, planned and actual treatment variables for each period, demographic information, stratification and subgrouping variables, important dates, etc. ADSL contains required variables (as specified in this document) plus other subject-level variables that are important in describing a subject's experience in the trial. ADSL and its related metadata are required in a CDISC-based submission of data from a clinical trial, even if no other ADaM datasets are submitted. Note that this ADaM requirement is also discussed in the FDA TCG v2.3.

A BDS dataset contains one or more records per subject, per analysis parameter, per analysis timepoint. Analysis timepoint is not required; it is dependent on the analysis. In situations where there is no analysis timepoint, the structure is one or more records per subject per analysis parameter. This structure contains a central set of variables that represent the actual data being analyzed. The BDS supports parametric and nonparametric analyses such as analysis of variance (ANOVA), analysis of covariance (ANCOVA), categorical analysis, logistic regression, Cochran-Mantel-Haenszel, Wilcoxon rank-sum, time-to-event analysis, etc.

Though the BDS supports most statistical analyses, it does not support all statistical analyses. For example, it does not support simultaneous analysis of multiple dependent (response/outcome) variables or a correlation analysis across a range of response variables.

The BDS was not designed to support analysis of incidence of adverse events or other occurrence data. Analysis of such data is supported in the OCCDS.

This version of the implementation guide does not fully cover dose-escalation trials or integration of multiple studies.

1.3.1 Other ADaM-Related CDISC Documents

Other CDISC documents relevant to the ADaM standard have been produced by the ADaM Team, or by the ADaM Team in concert with other CDISC teams. The most current versions of these documents can be found on the CDISC website.

Table 1.3.1.1 lists these documents and describes their applicability to ADaMIG versions 1.0 and 1.1.

Table 1.3.1.1 Other CDISC Documents and their Applicability to ADaMIG Versions

Document	ADaMIG v1.0	ADaMIG v1.1
Analysis Data Model (ADaM) v2.1, December 2009	Foundation document for ADaMIG v1.0	Still applicable
ADaM Examples in Commonly Used Statistical Analysis Methods v1.0, December 2011	Written for ADaMIG v1.0	Still applicable
The ADaM Basic Data Structure for Time-to-Event Analyses v1.0, May 2012	Written for ADaMIG v1.0	Still applicable
Update to the first CDISC SDTM/ADaM Pilot Project, January 2013	Written for ADaMIG v1.0	Still applicable
ADaM Data Structure for Adverse Event Analysis v1.0, May 2012	Written for ADaMIG v1.0	Superseded by OCCDS v1.0
ADaM Structure for Occurrence Data (OCCDS) v1.0, February, 2016	Not written for ADaMIG v1.0	Written for ADaMIG v1.1
CDISC ADaM Validation Checks v1.3, March 2015	Written for ADaMIG v1.0	Mostly applicable; v1.4 will be written for ADaMIG v1.1
Define-XML v2.0, March 2013	Applicable	Applicable
Analysis Results Metadata Specification for Define-XML Version 2 v1.0, January 2015	Applicable	Applicable

As shown in Table 1.3.1.1, for a particular version of the ADaMIG, documents with different release dates are often used together. If there are conflicts among the documents applicable to a particular version of the ADaMIG, the best practice is to use the content from the document with the latest date of final publication.

The ADaM Team is currently working on additional ADaM standard documents addressing the following topics:

- Integration of multiple studies
- Pharmacokinetics
- Oncology
- Questionnaires, Ratings, and Scales

Some CDISC Therapeutic Area User Guides (TAUGs) are released in draft, provisional, or final versions that discuss analysis and propose approaches to ADaM implementation. These documents are useful to implementers of the ADaM standard, but do not officially comprise part of the ADaM standard until such time as the ADaM Team incorporates their recommendations in a release of the foundational standard.

1.4 Organization of this Document

This document is organized into the following sections:

- Section 1, Introduction, provides an overall introduction to the importance of the ADaM standard and how it relates to other CDISC data standards.
- Section 2, Fundamentals of the ADaM Standard, provides a review of the fundamental principles that apply to all analysis datasets and introduces two standard structures that are flexible enough to represent the great majority of analysis situations.
- Section 3, Standard ADaM Variables, defines standard variables that commonly will be used in the ADaM standard data structures.
- Section 4, Implementation Issues, Standard Solutions, and Examples, presents standard solutions for implementation issues, illustrated with examples.
- Appendices provide additional background material and describe other supplemental material relevant to implementation.

Throughout this document the terms “producer” and “consumer” are used to refer to the originator/sender/owner/sponsor of the data and the reviewer/user/recipient of the data, respectively. These terms are used to simplify the document and to avoid any implication that the statements made in the document only apply to ADaM datasets in the context of electronic submissions to regulatory agencies.

1.5 Definitions

1.5.1 General ADaM Definitions

Analysis-enabling – Required for analysis. A column or row is analysis-enabling if it is required to perform the analysis. Examples: a hypertension category column added to the ADaM dataset to enable subgroup analysis; an age covariate added to enable the analysis to be age-adjusted; a center stratification factor in a multicenter study.

Traceability – The property that enables the understanding of the data’s lineage and/or the relationship between an element and its predecessor(s). Traceability facilitates transparency, which is an essential component in building confidence in a result or conclusion. Ultimately, traceability in ADaM permits the understanding of the relationship between the analysis results, the ADaM datasets, the SDTM datasets, and the data collection instrument. Traceability is built by clearly establishing the path between an element and its immediate predecessor. The full path is traced by going from one element to its predecessors, then on to their predecessors, and so on, back to the SDTM datasets, and ultimately to the data collection instrument.

Supportive – A column or row is supportive if it is not required in order to perform an analysis but is included in order to facilitate traceability or review. Example: the LBSEQ and VISIT columns were carried over from SDTM in order to promote understanding of how the ADaM dataset rows relate to the study tabulation dataset.

Record – A row in a dataset. A record is also referred to as an observation within this document.

Variable – A column in a dataset.

1.5.2 Basic Data Structure Definitions

Analysis parameter – A row identifier used to uniquely characterize a group of values that share a common definition. Note that the ADaM analysis parameter contains all of the information needed to uniquely identify a group of related analysis values. In contrast, the SDTM --TEST column may need to be combined with qualifier columns such as --POS, --LOC, --SPEC, etc., in order to identify a group of related values. Example: The primary efficacy analysis parameter is "3-Minute Sitting Systolic Blood Pressure (mmHg)." In this document the word "parameter" is used as a synonym for "analysis parameter."

Analysis timepoint – A row identifier used to classify values within an analysis parameter into temporal or conceptual groups used for analyses. These groupings may be observed, planned or derived. Example: The primary efficacy analysis was performed at the Week 2, Week 6, and Endpoint analysis timepoints.

Analysis value – (1) The numeric (AVAL) or character (AVALC) value described by the analysis parameter. The analysis value may be present in the input data, a categorization of an input data value, or derived. Example: The analysis value of the parameter "Average Heart Rate (bpm)" was derived as the average of the three heart rate values measured at each visit. (2) In addition, values of certain functions are considered to be analysis values. Examples: baseline value (BASE), change from baseline (CHG).

Parameter-variant – A column that is derived as a function of AVAL (or AVALC) is parameter-variant if it is calculated differently for some parameters for which the variable is populated in a dataset.

Parameter-invariant – A column that is derived as a function of AVAL (or AVALC) is parameter-invariant if it is calculated the same way for all parameters for which the variable is populated in a dataset. Thus, a column is parameter-invariant if how it is derived does not depend on which parameter is on the row. The parameter-invariant derivation remains the same across all parameters, though it may be left null for parameters where it does not apply. For example, the derivation for the change from baseline variable is CHG=AVAL-BASE, an equation that is the same for all parameters. CHG is therefore a parameter-invariant variable. The concept of parameter invariance is essential to the integrity of the BDS because it is an integral component in the rules defined in Section 4.2 that prohibit "horizontalization" (creation of new columns when the model dictates that a new row is required instead) by producers.

1.6 Analysis Datasets and ADaM Datasets

The section compares and contrasts analysis datasets in general versus ADaM datasets in particular.

Analysis dataset – An analysis dataset is defined as a dataset used for analysis and reporting.

ADaM dataset – An ADaM dataset is a particular type of analysis dataset that either:
(1) is compliant with one of the ADaM defined structures and follows the ADaM fundamental principles; or
(2) follows the ADaM fundamental principles defined in the ADaM model document and adheres as closely as possible to the ADaMIG variable naming and other conventions.

Non-ADaM analysis dataset – A non-ADaM analysis dataset is an analysis dataset that is not an ADaM dataset. Examples of non-ADaM analysis datasets include:

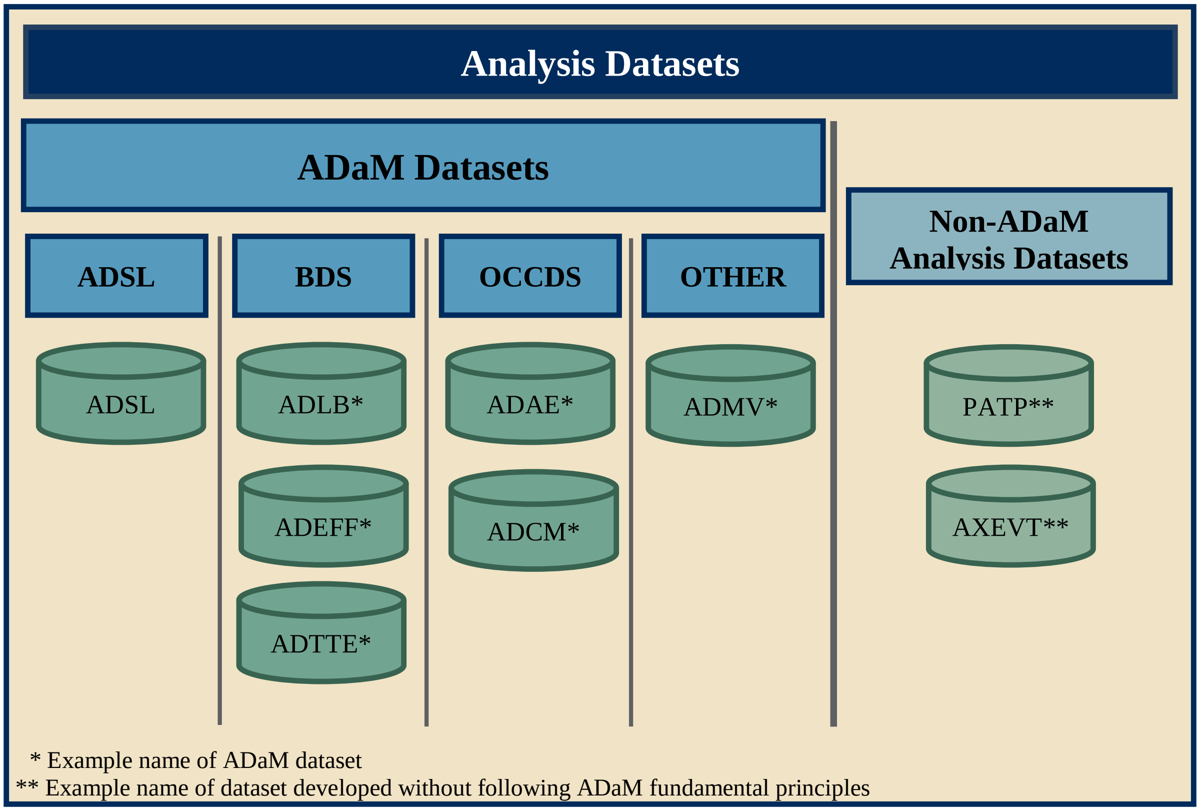
- an analysis dataset created according to a legacy company standard
- an analysis dataset that does not follow the ADaM fundamental principles.

It is important not to refer to non-ADaM analysis datasets as ADaM datasets.

To prevent confusion, non-ADaM analysis dataset names should not start with the prefix AD. It is good practice to start the names of non-ADaM analysis datasets with the two-letter prefix "AX".

Currently ADaM has three structures: ADSL (Subject-Level Analysis Dataset), BDS (Basic Data Structure), and OCCDS (Occurrence Data Structure). These three structures correspond to the SUBJECT LEVEL ANALYSIS DATASET, BASIC DATA STRUCTURE, and OCCURRENCE DATA STRUCTURE classes of ADaM datasets. Analysis datasets that follow the ADaM fundamental principles and other ADaM conventions, but that do not follow one of the three defined structures (ADSL, BDS, OCCDS), are considered to be ADaM datasets with a class of ADAM OTHER. Controlled terminology for the class element of the analysis dataset metadata can be downloaded at http://www.cdisc.org/terminology.

In the ADaM model, it is assumed that the original data sources for ADaM datasets are SDTM datasets, even when ADaM datasets are derived from other ADaM datasets. ADaM has features that enable traceability from analysis results to ADaM datasets and from ADaM datasets to SDTM datasets.



* Example name of ADaM dataset
** Example name of dataset developed without following ADaM fundamental principles

Figure 1.6.1 Categories of Analysis Datasets

Analysis Datasets within the eCTD Folder Structure

The specification for organizing datasets and their associated files in folders within a submission is summarized in the following figure, as noted in the FDA TCG v2.3. For ease of use with the define file and in the eCTD (electronic Common Technical Document) folder structure, all analysis datasets should be kept in one folder. If a set of analysis datasets includes an ADaM-compliant ADSL dataset (as required for a CDISC-conformant submission), then the whole set of analysis datasets should be placed into the **adam** folder. If not, the whole set of analysis datasets should be placed into the **legacy** folder.

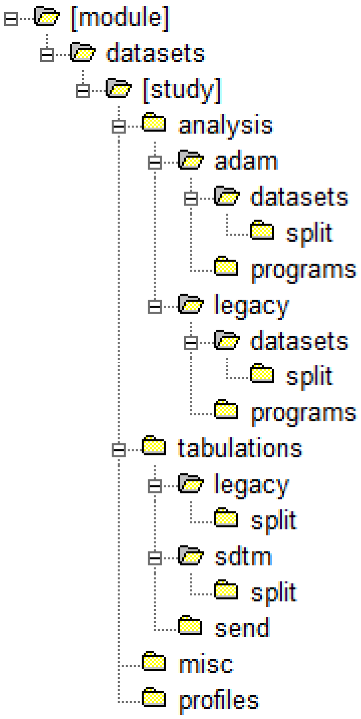


Figure 1.6.2 Analysis Data in the eCRT Structure

2 Fundamentals of the ADaM Standard

2.1 Fundamental Principles

ADaM datasets must adhere to certain fundamental principles as described in the ADaM model document:

- ADaM datasets and associated metadata must clearly and unambiguously communicate the content and source of the datasets supporting the statistical analyses performed in a clinical study.
- ADaM datasets and associated metadata must provide traceability to show the source or derivation of a value or a variable (i.e., the data's lineage or relationship between a value and its predecessor(s)). The metadata must identify when and how analysis data have been derived or imputed.
- ADaM datasets must be readily usable with commonly available software tools.
- ADaM datasets must be associated with metadata to facilitate clear and unambiguous communication. Ideally the metadata are machine-readable.
- ADaM datasets should have a structure and content that allow statistical analyses to be performed with minimal programming. Such datasets are described as "analysis-ready." ADaM datasets contain the data needed for the review and re-creation of specific statistical analyses. It is not necessary to collate data into "analysis-ready" datasets solely to support data listings or other non-analytical displays

Refer to the ADaM model document at http://www.cdisc.org/adam for more details.

2.2 Traceability

To assist review, ADaM datasets and metadata must clearly communicate how the ADaM datasets were created. The verification of derivations in an ADaM dataset requires having at hand the input data used to create the ADaM dataset. A CDISC-conformant submission includes both SDTM and ADaM datasets; therefore, it follows that the relationship between SDTM and ADaM must be clear. This requirement highlights the importance of traceability between the analyzed data (ADaM) and its input data (SDTM).

Traceability is built by clearly establishing the path between an element and its immediate predecessor. As described in Section 1.5.1, the full path is traced by going from one element to its predecessors, then on to their predecessors, and so on, back to the SDTM datasets, and ultimately to the data collection instrument.

Note that the CDISC Clinical Data Acquisition Standards Harmonization (CDASH) standard is harmonized with SDTM, and therefore assists in assuring end-to-end traceability.

Traceability establishes across-dataset relationships as well as within-dataset relationships. For example, the metadata for supportive variables within the ADaM dataset facilitates the understanding of how (and perhaps why) derived records were created.

There are two levels of traceability:

- Metadata traceability facilitates the understanding of the relationship of the analysis variable to its source dataset(s) and variable(s) and is required for ADaM compliance. This traceability is established by describing (via metadata) the algorithm used or steps taken to derive or populate an analysis variable from its immediate predecessor. Metadata traceability is also used to establish the relationship between an analysis result and ADaM dataset(s).
- Datapoint traceability points directly to the specific predecessor record(s) and should be implemented if practical and feasible. This level of traceability can be very helpful when trying to trace a complex data manipulation path. This traceability is established by providing clear links in the data (e.g., use of --SEQ variable) to the specific data values used as input for an analysis value. The BDS and OCCDS structures were designed to enable datapoint traceability back to predecessor data.

It may not always be practical or feasible to provide datapoint traceability via record identifier variables from the source dataset(s). However metadata traceability must always clearly explain how an analysis variable was populated regardless of whether datapoint traceability is also provided.

Very complex derivations may require the creation of intermediate analysis datasets. In these situations, traceability may be accomplished by submitting those intermediate analysis datasets along with their associated metadata. Traceability would then involve several steps. The analysis results would be linked by appropriate metadata to the data which supports the analytical procedure; those data would be linked to the intermediate analysis data; the intermediate data would in turn be linked to the source SDTM data.

When traceability is successfully implemented, it is possible to identify:

- Information that exists in the submitted SDTM data
- Information that is derived or imputed within the ADaM dataset
- The method used to create derived or imputed data
- Information used for analyses, in contrast to information that is not used for analyses yet is included to support traceability or future analysis

2.3 The ADaM Data Structures

A fundamental principle of ADaM datasets is clear communication. Given that ADaM datasets contain both source and derived data, a central issue becomes communicating how the variables and observations were derived and how observations are used to produce analysis results. The consumer of an ADaM dataset must be able to identify clearly the data inputs and the algorithms used to create the derived information. If this information is communicated in a predictable manner through the use of a standard data structure and metadata, the consumer of an ADaM dataset should be able to understand how to use the ADaM dataset to replicate results or to explore alternative analyses.

Many types of statistical analyses do not require a specialized structure. In other words, the structure of an ADaM dataset does not necessarily limit the type of analysis that can be done, nor should it limit the communication about the dataset itself. Instead, if a predictable structure can be used for the majority of ADaM datasets, communication will be enhanced.

A predictable structure has other advantages in addition to supporting clear communication. First, a predictable structure eases the burden of the management of dataset metadata because there is less variability in the types of observations and variables that are included. Second, software tools can be developed to support metadata management and data review, including tools to restructure the data (e.g., transposing) based on known key variables. Finally, a predictable structure allows an ADaM dataset to be checked for conformance with ADaM standards, using a set of known conventions which can be verified.

As described in Section 1, the ADaMIG describes two ADaM standard data structures: the subject-level analysis dataset (ADSL) and the Basic Data Structure (BDS). Standard ADaM variables are described in Section 3. Implementation issues, solutions, and examples are presented in Section 4. Together, Sections 3 and 4 fully specify these standard data structures. A description of OCCDS (the third ADaM standard data structure) can be found in the document titled “ADaM Structure for Occurrence Data” as noted in Section 1.3.

2.3.1 The ADaM Subject-Level Analysis Dataset (ADSL)

ADSL contains one record per subject, regardless of the type of clinical trial design. The label of the ADSL dataset is “Subject-Level Analysis Dataset”.

ADSL contains variables such as subject-level population flags, planned and actual treatment variables, demographic information, randomization factors, subgrouping variables, and important dates. ADSL contains required variables (as specified in Section 3.2) plus other subject-level variables that are important in describing a subject’s experience in the trial. This structure allows merging with any other dataset, including ADaM and SDTM datasets. ADSL is a source for subject-level variables used in other ADaM datasets, such as population flags and treatment variables.

It should be noted that though ADSL is a source for subject-level variables used in other datasets, there is no requirement that every ADSL variable be copied into a BDS dataset. However, at a minimum, all ADSL variables needed to enable analysis of the BDS dataset, such as statistical model covariates, population flags, subgrouping variables, etc., should be copied from ADSL into the BDS dataset. In addition, it is desirable to copy ADSL variables that are useful for traceability or supportive of review. Note that FDA TCG v2.3 requests that "core" subject-level variables be present in all analysis datasets. Refer to current regulatory agency requirements (see Section 1.2) and discuss with the reviewing agency.

Although it would be technically feasible to take every single data value in a study and include them all as variables in a subject-level dataset, such as ADSL, that is not the intent or the purpose of ADSL. ADSL is used to provide key facts about the subject that are analysis-enabling or facilitate interpretation of analysis. ADSL is not the correct location for key endpoints and data that vary over time during the course of a study.

There is only one ADSL per study. ADSL and its related metadata are required in a CDISC-based submission of data from a clinical trial even if no other ADaM datasets are submitted.

2.3.2 The ADaM Basic Data Structure (BDS)

A BDS dataset contains one or more records per subject, per analysis parameter, per analysis timepoint. Analysis timepoint is conditionally required, depending on the analysis. In situations where there is no analysis timepoint, the structure is one or more records per subject per analysis parameter. As defined in Section 1.5.2, analysis timepoint here represents one or more actual variables, such as Analysis Visit (AVISIT), Analysis Timepoint (ATPT), or other timing variables. In addition, other variables may be needed to describe more completely the structure of a given BDS dataset. For example, Baseline Type (BASETYPE) is needed when there is more than one definition of baseline for a given Analysis Parameter (PARAM) in the same dataset. This structure contains a central set of variables that represent the data being analyzed. These variables include the value being analyzed (e.g., AVAL) and the description of the value being analyzed (e.g., PARAM). Other variables in the dataset provide more information about the value being analyzed (e.g., the subject identification), describe and trace the derivation of it (e.g., DTYPE), or enable the analysis of it (e.g., treatment variables, covariates). It should be noted that though ADSL is a source for subject-level variables used in BDS datasets, this does not mean that every ADSL variable should be included in the BDS dataset.

Readers are cautioned that ADaM dataset structures do not have counterparts in SDTM. Because the BDS tends toward a vertical design, some might perceive it as similar to the SDTM Findings class. However, BDS datasets may be derived from Findings, Events, Interventions and Special-Purpose SDTM domains, other ADaM datasets, or any combination thereof. Furthermore, in contrast to SDTM Findings class datasets, BDS datasets provide robust and flexible support for the performance and review of most statistical analyses.

A record in an ADaM dataset can represent an observed, derived, or imputed value required for analysis. For example, it may be a time to an event, such as the time to when a score became greater than a threshold value or the time to discontinuation, or it may be a highly derived quantity such as a surrogate for tumor growth rate derived by fitting a regression model to laboratory data. A data value may be derived from any combination of SDTM and/or ADaM datasets.

The BDS is flexible in that additional rows and columns can be added to support the analyses and provide traceability, according to the rules described in Section 4.2. However, it should be stressed that in a study there is often more than one ADaM dataset that follows the BDS. The capability of adding rows and columns does not mean that everything should be forced into a single ADaM dataset. The optimum number of ADaM datasets should be designed for a study, as discussed in the ADaM model document.

3 Standard ADaM Variables

This section defines the required characteristics of standard variables (columns) that are frequently needed in ADaM datasets. The ADaM standard requires that these variable names be used when a variable that contains the content defined in Section 3 is included in an ADaM dataset. It requires these ADaM standard variables be used for the purposes indicated, even if the content of an ADaM variable is a copy of the content of an SDTM variable.

This section also defines standard naming fragments (with position within the variable name included as the part of the definition in some instances) to be used in creating new variable names. In the variable name fragments below, a ‘*’ is used to indicate that one or more letters can be added to create a producer-specific variable name. If a fragment is defined for a specific concept (Section 3.1.5, Variable Naming Fragments), it is best practice that any variable related to the concept contain the defined fragment in its name. Specific fragments, described in Table 3.1.5.1, are required to be used whenever the concept applies and are reserved to be used only for the corresponding concept. For example, the fragment “DTF” is defined as a suffix for date imputation flag variables; therefore, a variable that indicates whether or not a date has been imputed contains “DTF” as the last three characters in the variable name. In addition, Table 3.1.5.2 and Table 3.3.3.3 list fragments that can be used when naming variables in ADaM datasets. These lists of fragments are provided as a guide when naming variables in ADaM datasets, and are to be used in addition to the fragments defined in the SDTMIG. Section 3.1 defines ADaM Variable Conventions that apply to all ADaM variables, including the standard ADaM variables specified in Sections 3.2 and 3.3, as well as when defining new ADaM variables. Section 3.2 describes variables in ADSL. Section 3.3 describes variables in the BDS. Section 3.4 describes variables that are not specific to the ADSL or BDS structures.

In this section, ADaM variables are described in tabular format. The two rightmost columns, “Core” and “CDISC Notes” provide information about the variables to assist producers in preparing their datasets. These columns are not meant to be metadata submitted in define.xml. The “Core” column describes whether a variable is required, conditionally required, or permissible. The “CDISC Notes” column provides more information about the variable. In addition, the “Type” column specifies whether the variable being described is character or numeric, though more specific datatype information will be provided in metadata (e.g., text, integer, float).

Values of ADaM “Core” Attribute

Req = Required. The variable must be included in the dataset.
Cond = Conditionally required. The variable must be included in the dataset in certain circumstances.
Perm = Permissible. The variable may be included in the dataset, but is not required.

Unless otherwise specified, all ADaM variables are populated as appropriate, meaning nulls are allowed.

3.1 ADaM Variable Conventions

3.1.1 General Variable Conventions

1. To ensure compliance with SAS Version 5 transport file format and Oracle constraints, all ADaM variable names must be no more than 8 characters in length, start with a letter (not underscore), and be comprised only of letters (A-Z), underscore (_), and numerals (0-9). All ADaM variable labels must be no more than 40 characters in length. All ADaM character variables must be no more than 200 characters in length.
2. The lower case letters "*w*", "*xx*", "*y*", and "*zz*" that appear in a variable name or label in this document must be replaced in the actual variable name or label using the following conventions.
    a. The lower-case letter "*w*" in a variable name (e.g., PH*w*SDT, P*xx*S*w*SDT) is an index for the *w*th variable where "*w*" is replaced with a single digit [1-9].
    b. The letters "*xx*" in a variable name (e.g., TRT*xx*P, AP*xx*SDT) refer to a specific period where "*xx*" is replaced with a zero-padded two-digit integer [01-99]. The use of '*xx*' within a variable name is restricted to the concept of a period.
    c. The lower-case letter "*y*" in a variable name (e.g., SITEGR*y*) refers to a grouping or other categorization scheme, an analysis criterion, or an analysis range, and is replaced with an integer [1-99, not zero-padded]. Truncation of the original variable name may be necessary in rare situations when a two digit index is needed and causes the length of the variable name to exceed 8 characters. In these situations, it is recommended that the same truncation be used for both the character and numeric versions of the variables in a variable pair.
    d. The lower-case letters "*zz*" in a variable name (e.g., ANL*zz*FL) are an index for the *zz*th variable where "*zz*" is replaced with a zero-padded two-digit integer [01-99]. Note that the '*zz*' convention represents a simple counter, while the '*xx*' convention represents a specific period.
    e. If an indexed variable is included in a dataset, there is no requirement that the preceding variable(s) in the sequence be included. For example, a dataset might include ANL02FL but not ANL01FL.
3. Any variable in an ADaM dataset whose name is the same as an SDTM variable must be a copy of the SDTM variable, and its label, meaning, and values must not be modified. ADaM adheres to a principle of harmonization known as "same name, same meaning, same values." However, to optimize file size, it is permissible that the length of the variables differ (e.g., trailing blanks may be removed). In many cases it makes sense to copy over an SDTM variable. For example, the SDTM variable --SEQ may be useful for traceability. However, in other cases, it is also perfectly acceptable, and might be much better, to create an ADaM variable with a meaningful variable name and clear and unambiguous metadata. An SDTM variable may be somewhat meaningless when removed from its SDTM context. For example, the meaning of the SDTM variable DSDECOD may depend on other SDTM variables such as DSCAT and DSSCAT, and ultimately on how the data were collected and mapped to SDTM in a particular study; thus it may be better to create a clearly-defined ADaM variable. In any case, whenever values are modified in any way, it is mandatory to do so in an ADaM variable, and it is prohibited to do so in a variable whose name is that of an SDTM variable.
4. When an ADaM standard variable name has been defined for a specific concept, the ADaM standard variable name must be used, even if the content of an ADaM variable is a direct copy of the content of an SDTM variable. For example, in the creation of ADLB, even if AVAL is just a copy of LBSTRESN the dataset must contain AVAL.
5. For variable pairs designated as having a one-to-one mapping within a specified scope (e.g., within a parameter, within a study), if both variables are present in the dataset and there exists a row in that scope on which both variables are populated, then there must be a one-to-one mapping between the two variables on all rows within the scope on which both variables are populated. The scopes noted in this document should be considered the minimum level for the mapping; it does not preclude the producer from using a broader level of scope. For example, if a one-to-one mapping is specified as within

a PARAM, the producer may elect to use the same one-to-one mapping across all PARAMs within the dataset or study. In addition, note that “within a parameter” means “within a parameter within a dataset.”

6. In a pair of corresponding variables (e.g., TRTP and TRTPN), the primary or most commonly used variable does not have the suffix or extension (e.g., N for Numeric or C for Character). The relevant suffix is used only on the name of the secondary member of the variable pair. For example, in the (TRTP, TRTPN) pair, the primary variable, TRTP is character; but it is not named TRTPC. Similarly in the (APERIOD, APERIODC) pair, the primary variable, APERIOD, is numeric; but it is not named APERIODN. When the secondary variable is numeric, it can only be included if the primary variable is also present in the dataset. If both variables of a variable pair are present, there must be a one-to-one mapping between the values of the two variables, as described in Item 5 above.
7. In general, if SDTM character variables are converted to numeric variables in ADaM datasets, then they should be named as they are in the SDTM with an “N” suffix added. For example, the numeric version of the DM SEX variable is SEXN in an ADaM dataset, and a numeric version of RACE is RACEN. As stated previously, the secondary variable of the variable pair cannot be present in the dataset unless the primary variable is also present. Applying that to the variable pairs being described in this item, the numeric equivalent of the variable cannot be present in the dataset unless the character version is also present. If necessary to keep within the 8-character variable name length limit, the last character may be removed prior to appending the N. Note that this naming scheme applies only to numeric variables whose values map one-to-one to the values of the equivalent character variables. Note also that this convention does not apply to SDTM date/time ISO8601-formatted character variables converted to ADaM numeric *DT, *TM, and *DTM variables.
8. Variables whose names end in FL are character flag (or indicator) variables with at most two possible non-missing values, Y or N (i.e., yes or no). The name of the corresponding numeric flag (or indicator) variable ends in FN. If the flag is included in an ADaM dataset, the character version (*FL) is required but the corresponding numeric version (*FN) can also be included. If both versions of the flag are included, there must be a one-to-one mapping between the values of the two variables, as described in Section 3.1.4.
9. Variables whose names end in GR*y*, G*y*, or CAT*y* are grouping variables, where y refers to the grouping scheme or algorithm (not the category within the grouping). For example, SITEGR3 is the name of a variable containing site group (pooled site) names, where the grouping has been done according to the third site-grouping algorithm; SITEGR3 does not mean the third group of sites. Within this document, CAT*y* is the suffix used for categorization of ADaM-specified analysis variables (e.g., CHGCAT*y* categorizes CHG).
10. It is recommended that producer-defined grouping or categorization variables begin with the name of the variable being grouped and end in GR*y* (e.g., variable ABCGR*y* is a character description of a grouping or categorization of the values from the ABC variable for analysis purposes). If any grouping of values from an SDTM variable is done, the name of the derived ADaM character grouping variable should begin with the SDTM variable name and end in GR*y* (GR*y*N for the numeric equivalent) where y is an integer [1-99, not zero-padded] representing a grouping scheme. For example, if a character analysis variable is created to contain values of Caucasian and Non-Caucasian from the SDTM RACE variable, then it should be named RACEGR*y* and its numeric equivalent should be named RACEGR*y*N (e.g., RACEGR1, RACEGR1N). As described in Table 3.1.5.1, G*y* can be used as an abbreviated form of GR*y* when the use of GR*y* would create a variable name longer than 8 characters. Truncation of the original variable name may be necessary when appending suffix fragments GR*y*, GR*y*N, G*y*, or G*y*N.

3.1.2 Timing Variable Conventions

1. Numeric dates, times and datetimes should be formatted, so as to be human-readable with no loss of precision.
2. Variables whose names end in DT are numeric dates.
3. Variables whose names end in DTM are numeric datetimes.

4. Variables whose names end in TM are numeric times.
5. If a *DTM and associated *TM variable exist, then the *TM value must match the time part of the *DTM value when the *DTM variable is populated. If a *DTM and associated *DT variable exist, then the *DT value must match the date part of the *DTM value when the *DTM variable is populated.
6. Names of timing start variables end with an S followed by the characters indicating the type of timing (i.e., SDT, STM, SDTM), unless otherwise specified elsewhere in Section 3.
7. Names of timing end variables end with an E followed by the characters indicating the type of timing (i.e., EDT, ETM, EDTM), unless otherwise specified elsewhere in Section 3.
8. Variables whose names end in DY are relative day variables. In ADaM as in the SDTM, there is no day 0. If there is a need to create a relative day variable that includes day 0, then its name must not end in DY.
9. ADaM relative day variables need not be anchored by SDTM RFSTDTC. The anchor (i.e., reference) date variable must be indicated in the variable-level metadata for the relative day variable. The anchor date variable should also be included in ADSL or the current ADaM dataset to facilitate traceability. Similarly, anchor time variables used to calculate values for ADaM relative time variables must be indicated in the variable-level metadata for the relative time variable, and must be included in ADSL or the current ADaM dataset. Note that it is possible to have different definitions for a relative day (or time) variable (e.g., ADY) in separate datasets, using different anchor dates (or times). For example, the derivation of ADY for efficacy datasets might be different from that for safety datasets.
10. Table 3.3.3.3 presents standard suffix naming conventions for producer-defined supportive variables containing numeric dates, times, datetimes, and relative days, as well as date and time imputation flags. These conventions are applicable to all ADaM datasets. The asterisk that appears in a variable name in the table must be replaced by a suitable character string, so that the actual variable name is meaningful and complies with the restrictions noted in Section 3.1.1.
11. The reader is cautioned that the root or prefix (represented by *) of such producer-specified supportive ADaM date, time and datetime variable names must be chosen with care, to prevent unintended conflicts among other such names and standard numeric versions of possible SDTM variable names. In particular, **potentially problematic values for producer-defined roots/prefixes (*) include:**
    a. One-letter prefixes.
       For an example of the problem, if * is Q, then a date *DT would be QDT; however, a starting date *SDT would be QSDT, which would potentially be confusing if the producer intended QSDT to be something other than the numeric date version of the SDTM variable QSDTC.
    b. Two-letter prefixes, except when intentionally chosen to refer explicitly to a specific SDTM domain and its --DTC, --STDTC, and/or --ENDTC variables.
       For an example of an appropriate intentional use of a two-letter prefix, if * is LB, then *DT is LBDT, the numeric date version of SDTM LBDTC.
       For an example of the problem, if * is QQ, then a date *DT would be QQDT, which would potentially be confusing if the producer intended QQDT to be something other than the numeric date version of a potential SDTM variable QQDTC.
    c. Three-letter prefixes ending in S or E.
       For an example of the problem, if * is QQS, then a date *DT would be QQSDT, which would potentially be confusing if the producer intended QQSDT to be something other than the numeric date version of a potential SDTM variable QQSTDTC.
12. In general, all three of *DT, *TM, *DTM are not required. Include only the *DT, *TM, and *DTM variables needed for analysis or review. However, when a *DTM variable exists, it is good practice to include a corresponding *DT variable.

For more information regarding date and time variable conventions, refer to Table 3.3.3.3.

3.1.3 Date and Time Imputation Flag Variables

When a date or time is imputed, it is required that the variable containing the imputed value be accompanied by a date or time imputation flag variable. The variable fragments to be used for these variables are DTF and TMF, as defined in Table 3.1.5.1. DF and TF can be used as abbreviated forms of DTF and TMF, respectively, when the use of DTF or TMF would create a variable name longer than 8 characters. These additional imputation flag variables are conditionally required. The root, identified by *, of the names of each pair of variables, *DT and *DTF (or *DF), should be identical. The same is true for the corresponding time and imputation flag variables *TM and *TMF (or *TF). Thus it is good practice to limit roots to 5 characters in length.

It should be noted that in many instances in Section 3, specific DTF and TMF flags are defined within sets of timing variables. However, imputation flags should be created for all date or time variables when imputation has been performed, even if there is not a specific variable mentioned in Section 3 (e.g., for EOSDT).

1. As described in Table 3.1.5.1, variables whose names end in DTF are date imputation flags. *DTF variables represent the highest level of imputation of the *DT variable based on the source SDTM DTC variable. *DTF = Y if the year is imputed. *DTF = M if year is present and month is imputed. *DTF = D if only day is imputed. *DTF = null if *DT equals the SDTM DTC variable date part equivalent. If a date was imputed, *DTF must be populated and is required. Both *DTF and *TMF may be needed to describe the level of imputation in *DTM if imputation was done. Note that the list of examples in Table 3.1.3.1 is not exhaustive.

Table 3.1.3.1 Some Examples of Setting of Date Imputation Flag

Missing Elements	**SDTM --DTC String**	**ADaM Date Value (*DT Variable)[1,2]** **(## indicates imputed portion)**	**Imputation flag (*DTF variable)**
None	YYYY-MM-DD	YYYY-MM-DD	Blank
Day	YYYY-MM	YYYY-MM-##	D
Month	YYYY---DD	YYYY-##-DD	M
Month and Day	YYYY	YYYY-##-##	M
Year	--MM-DD	####-MM-DD	Y
Year and Month	----DD	####-##-DD	Y
Year and Month and Day		####-##-##	Y

[1] The ISO formats used in the ADaM Date Value column are for the purposes of illustration, and are not intended to imply any type of display standard or requirement. The DT variable is numeric and the producer will determine the appropriate display format.

[2] The indication of imputed values is not intended to imply an imputation rule or standard. For example, if the month is missing, imputation rules might specify that the collected day value be ignored so that both month and day are imputed.

2. As described in Table 3.1.5.1, variables whose names end in TMF are time imputation flags. *TMF variables represent the level of imputation of the *TM (and *DTM) variable based on the source SDTM DTC variable. *TMF = H if the entire time is imputed. *TMF = M if minutes and seconds are imputed. *TMF = S if only seconds are imputed. *TMF = null if *TM equals the SDTM DTC variable time part equivalent. For a given SDTM DTC variable, if only hours and minutes are ever collected, and seconds are imputed in *DTM as 00, then it is not necessary to set *TMF to “S”. However if seconds are generally collected but are missing in a given value of the DTC variable and imputed as 00, or if a collected value of seconds is changed in the creation of *DTM, then *TMF should be set to “S”. If a time was imputed *TMF must be populated and is required. Both *DTF and *TMF may be needed to describe the level of imputation in *DTM if imputation was done.

Note that using SDTM --DTC variables for comparison purposes in analysis algorithms may be problematic in the presence of missing date or time elements. SDTM --DTC variables containing date, time, and datetime values are character strings that, in the presence of missing elements (year, month, day, hour, minute, or second), sort or compare in a manner that may be equivalent to imputation of missing elements with the lowest possible value. For example, if in a given --DTC variable in a dataset, dates are present on all records, but time is missing on some records, then within any given date, the records with missing time may sort or compare before the records that contain a value of time. Thus the --DTC variable would sort or compare in a manner that is equivalent to imputing midnight when time is missing. The sort or comparison may work mechanically, but imputing midnight may not be the most appropriate thing to do for statistical analysis. Furthermore, the effective imputation of midnight would be hidden and not made explicit. It is important to consider the implications of implicit or explicit imputation whenever dates, times, or datetimes are compared or sorted.

3.1.4 Flag Variable Conventions

1. The terms “flag” and “indicator” are used interchangeably within this document, and “flag variables” are sometimes referred to simply as “flags.”
2. Population flags must be included in a dataset if the dataset is analyzed by the given population. At least one population flag is required for datasets used for analysis. A character indicator variable is required for every population that is defined in the statistical analysis plan. All applicable subject-level population flags must be present in ADSL.
3. Character and numeric subject-level population flag names end in FL and FN, respectively. Similarly, parameter-level population flag names end in PFL and PFN, and record-level population flag names end in RFL and RFN. Please also refer to Item 8 in Section 3.1.1.
4. For subject-level character population flag variables: N = no (not included in the population), Y = yes (included). Null values are not allowed.
5. For subject-level numeric population flag variables: 0 = no (not included), 1 = yes (included). Null values are not allowed.
6. For parameter-level and record-level character population flag variables: Y = yes (included). Null values are allowed. Note that the controlled terminology is not the same for these population flag variables as for subject-level population flag variables. Depending on how validation checks are written, this difference could cause an issue for a producer-defined subject-level flag variable with a name that ends in “RFL” or “PFL” if it is copied into a BDS dataset.
7. For parameter-level and record-level numeric population flag variables: 1 = yes (included). Null values are allowed. Depending on how validation checks are written, this difference could cause an issue for a producer-defined subject-level flag variable with a name that ends in “RFN” or “PFN” if it is copied into a BDS dataset.
8. In addition to the population flag variables defined in Section 3, other population flag variables may be added to ADaM datasets as needed, and must comply with these conventions.
9. For character flags with variable names that end in FL and that are not population flags, a scheme of Y/N/null, or Y/null may be specified. As indicated in Table 3.3.4.2, and Table 3.3.8.1, some common character flags use the scheme Y/null. Corresponding 1/0/null and 1/null schemes apply to numeric flags with variable names that end in FN and that are not population indicators.
10. Additional flags may be added if their names and values comply with these conventions.

3.1.5 Variable Naming Fragments

Table 3.1.5.1 contains a list of standard suffix fragments (i.e., variable name fragments used as the last part of a variable name) that are required when naming variables in ADaM datasets, as defined in Section 3.1. For these fragments, it is a requirement that the appropriate fragment be used whenever the concept applies and that the fragment is reserved to be used only for the corresponding concept. For example, a variable whose name ends in DT must contain a numeric date, and a variable created to contain a numeric date must have a name ending in DT.

Table 3.1.5.1 Required Suffix Fragments for Use in Naming ADaM Variables

Fragment	CDISC Notes
GRy	Suffix used in names of grouping variables, where y refers to the grouping scheme or algorithm (not the category within the grouping). Note that GRy can be abbreviated to Gy when necessary to comply with the variable name length limit of 8 characters. The corresponding numeric version of the variable will use the suffix GRyN (or GyN if the Gy abbreviation is used). For more information on grouping variables see Section 3.1.1. See Table 3.2.2 for examples of grouping variables.
FL	Suffix used in names of character flag variables, when the valid values of the variable are Y/Null or Y/N/Null. The corresponding numeric version of the variable will use the suffix FN. For more information on flag variables, see Section 3.1.1 and Section 3.1.4. See Table 3.2.3, Table 3.3.4.2, and Table 3.3.8.1 for examples of flag variables.
DT	Suffix used in names of numeric date variables. For more information on timing variables, see Section 3.1.2. See Section 3.3.3 for examples of timing variables.
TM	Suffix used in names of numeric time variables. For more information on timing variables, see Section 3.1.2. See Section 3.3.3 for examples of timing variables. Note that although ADaM variable ARELTM ends in TM, it is an exception, and is not a numeric time variable. In addition, the SDTM variables --ELTM are not numeric time variables.
DTM	Suffix used in names of numeric datetime variables. For more information on timing variables, see Section 3.1.2. See Section 3.3.3 for examples of timing variables.
DTF	Suffix used in names of date imputation flag variables. Note that DTF can be abbreviated to DF to comply with the variable name length limit of 8 characters. For more information, see Section 3.1.3. See Section 3.3.3 for examples of timing imputation variables.
TMF	Suffix used in names of time imputation flag variables. Note that TMF can be abbreviated to TF to comply with the variable name length limit of 8 characters. For more information, see Section 3.1.3. See Section 3.3.3 for examples of timing imputation variables.
DY	Suffix used in names of relative day variables that do not include day 0. For more information on timing variables, see Section 3.1.2. See Section 3.3.3 for examples of timing variables.

Table 3.1.5.2 contains a list of additional standard reserved fragments to use as a guide when naming variables in ADaM datasets. This list should be used in addition to the list of timing fragments defined in Table 3.3.3.3 and to the fragments defined in the SDTMIG. It should be noted that some concepts have slightly different fragments in ADaM than in the SDTMIG; the ADaM fragment takes precedence when creating an ADaM variable. When using fragments, the general rule is to use the fragment(s) that best conveys the meaning of the variable within the 8-character limit. The list of fragments is provided as a guideline, not as a requirement.

Table 3.1.5.2 Additional Fragments That May Be Used in Naming ADaM Variables

Fragment	CDISC Notes
BL	Baseline, position relative to type of variable. Not to be used to support more than one baseline definition for AVAL in BDS datasets. See note below.
CHG	Change, position relative to type of variable. Not to be used to support change from more than one baseline for AVAL in BDS datasets. See note below.
FU	Follow-up, position relative to type of variable.
OT	On treatment, position relative to type of variable.
RU	Run-in, position relative to type of variable.
SC	Screening, position relative to type of variable.

Fragment	CDISC Notes
TA	Taper, position relative to type of variable.
TI	Titer, position relative to type of variable.
U	Units, suffix To identify the units for a variable, a separate variable can be created, using the name of the original variable with a “U” suffix added. To keep within the 8-character variable name length limit, some truncation may be necessary prior to appending the U. In situations where the units do not vary within the ADaM dataset, it may be preferable to simply include the units in the variable’s label and metadata. The approach taken will be determined by the producer, based on the requirements of the analysis and review of the dataset. Note that there is no separate units variable for BDS variables PARAM or AVAL, since the units of AVAL will be included in the value of PARAM.
WA	Washout, position relative to type of variable.

Note that in BDS datasets, there is only one baseline variable, BASE, and only one change from baseline variable, CHG. The BL and CHG fragments must not be used to create BDS variables containing alternative baselines and changes from baseline for AVAL. Additional definitions of baseline relevant to a given parameter must be accommodated by the addition of rows rather than addition of variables. See Section 3.3.4 and Section 4.2, Rule 6. However, if the baseline or change from baseline of a different parameter is needed in the analysis of a given parameter, for example, as a covariate in an analysis of covariance, that analysis-enabling variable may be added and its name should contain the fragment BL or CHG.

There are two main categories of variable names relative to timepoints: content at a particular timepoint (e.g., weight at baseline) and timepoint timing (e.g., screening date). Assembly of these types of variable names using timing fragments defined in Table 3.1.5.2 is described below:

Content at a timepoint: Because the timing of a variable qualifies the content of the variable, timing fragments are used as the variable name suffix. The complete variable naming convention is *(xx)FF, where * represents the content of the variable (up to 4 characters) and FF represents the timing fragment. For any timing fragments that are repeated for multiple periods, the period number (xx) should be placed before the suffix. If period numbers are not needed, the variable will be of the form *FF, with * representing the content of the variable (up to 6 characters)

Timepoint timing: If timepoint variables are needed that would use these timing fragments, the timing fragments will become the prefix of the variable name. For dates, then, the structure of the variable name is FF(xx)*, where * represents the date fragment (e.g., DT, SDT, EDTM, etc.) (up to 4 characters) and FF represents the timing fragment. This is consistent with ADSL timepoint variables such as RANDDT, TR01SDT, etc.

Some examples of variable names that follow these guidelines are:

- SBP01BL, SBP02BL – Period-level baseline variables for systolic blood pressure. Other abbreviations for systolic blood pressure are also acceptable.
- WEIGHTSC or WTSC – Screening weight. Other abbreviations of weight are also acceptable.
- RUSDT – Run-in start date, using the timing fragment as the prefix in a timing variable as defined in Table 3.3.3.3.
- WA01SDT, WA01EDT, WA02SDT and WA02EDT – Washout start and end dates for two periods, using the timing fragment as the prefix in a timing variable as defined in Table 3.3.3.3.

3.1.6 Additional Information about Section 3

In general, the variable labels specified in the tables in Section 3 are required. There are only two exceptions to this rule:

1. Descriptive text is allowed at the end of the labels of variables whose names contain indexes “y” or “zz”; and
2. Variable labels containing a word or phrase in brackets, e.g. {Time}, should be replaced by the producer with appropriate text that contains the bracketed word or phrase somewhere in the text (e.g., the label for a *TM variable is indicated as {Time} in this document) indicating any producer-defined label is permitted as long as the word Time is incorporated in it.

It is important to note that the standard variable labels by no means imply the use of standard derivation algorithms across studies and/or producers.

It should be noted that when the CDISC Notes for a variable refer to another variable, it is understood that this means “on the same record or row”. For example, the CDISC notes for TRTPN state “The numeric code for TRTP [on the same record]” where the text in brackets is understood.

Controlled terminology has been developed for the values of certain ADaM variables. The most current CDISC terminology sets can be accessed via the CDISC website (http://www.cdisc.org/terminology). In the tables in Section 3, the parenthesized external codelist name appears in the column labeled “Codelist/ Controlled Terms” where relevant. Where examples of controlled terms appear in this document, they should be considered examples only; the official source is the latest CDISC set available through the website.

Note that CDISC controlled terminology sets cannot represent null (absence of a value) in the list of valid terms since null isn’t a term. However, unless specified in the definition for a specific variable below, null is allowed.

Additional variables not defined in Section 3 may be necessary to enable the analysis or to support traceability and may therefore be added to ADaM datasets, providing that they adhere to the ADaM naming conventions and rules as defined in this document.

3.2 ADSL Variables

In the ADaM model document, it is noted that an ADaM-compliant ADSL dataset and its related metadata are required in a CDISC-based submission of data from a clinical trial even if no other ADaM datasets are submitted. The structure of ADSL is one record per subject, regardless of the type of clinical trial design.

This section lists standard ADSL variables. Section 2.3.1 describes the content of ADSL and addresses the kinds of variables that are and are not appropriate for inclusion in ADSL. Within a given study, USUBJID is the key variable that links ADSL to other datasets (both SDTM and ADaM).

For ADSL variables, the scope is “within the study.” For example, the definition of SITEGR1 is consistent for all datasets within a study. It is acknowledged that the scope of USUBJID extends beyond the study, as defined in the SDTM Implementation Guide.

Table 3.2.1 ADSL Identifier Variables

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
STUDYID	Study Identifier	Char		Req	DM.STUDYID
USUBJID	Unique Subject Identifier	Char		Req	DM.USUBJID
SUBJID	Subject Identifier for the Study	Char		Req	DM.SUBJID. SUBJID is required in ADSL, but permissible in other datasets.
SITEID	Study Site Identifier	Char		Req	DM.SITEID. SITEID is required in ADSL, but permissible in other datasets.
SITEGRy	Pooled Site Group y	Char		Perm	Character description of a grouping or pooling of clinical sites for analysis purposes. For example, SITEGR3 is the name of a variable containing site group (pooled site) names, where the grouping has been done according to the third site grouping algorithm, defined in variable metadata; SITEGR3 does not mean the third group of sites.
SITEGRyN	Pooled Site Group y (N)	Num		Perm	The numeric code for SITEGRy. One-to-one mapping to SITEGRy within a study.
REGIONy	Geographic Region y	Char		Perm	Character description of geographical region. For example, REGION1 might have values of ‘Asia’, ‘Europe’, ‘North America’, ‘Rest of World’; REGION2 might have values of ‘United States’, ‘Rest of World’.
REGIONyN	Geographic Region y (N)	Num		Perm	The numeric code for REGIONy. Orders REGIONy for analysis and reporting. One-to-one mapping to REGIONy within a study.

Table 3.2.2 ADSL Subject Demographics Variables

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
AGE	Age	Num		Req	DM.AGE. If analysis needs require a derived age that does not match DM.AGE, then AAGE must be added
AGEU	Age Units	Char	(AGEU)	Req	DM.AGEU
AGEGRy	Pooled Age Group y	Char		Perm	Character description of a grouping or pooling of the subject’s age for analysis purposes. For example, AGEGR1 might have values of “<18”, “18-65”, and “>65”; AGEGR2 might have values of “Less than 35 y old” and “At least 35 y old”.
AGEGRyN	Pooled Age Group y (N)	Num		Perm	The numeric code for AGEGRy. Orders the grouping or pooling of subject age for analysis and reporting. One-to-one mapping to AGEGRy within a study.
AAGE	Analysis Age	Num		Cond	Age used for analysis that is derived differently from DM.AGE. AAGE is required if age is calculated differently than in SDTM.
SEX	Sex	Char	(SEX)	Req	The sex of the subject is a required variable in ADSL; must be identical to DM.SEX.
RACE	Race	Char	(RACE)	Req	The race of the subject is a required variable in ADSL; must be identical to DM.RACE.
RACEGRy	Pooled Race Group y	Char		Perm	Character description of a grouping or pooling of the subject’s race for analysis purposes.
RACEGRyN	Pooled Race Group y (N)	Num		Perm	The numeric code for RACEGRy. Orders the grouping or pooling of subject race for analysis and reporting. One-to-one mapping to RACEGRy within a study.

Population flags are required by ADaM. Table 3.2.3 describes ADaM population flags, though the list is not meant to be all-inclusive. See Section 3.5 for details on the differences between SDTM- and ADaM-defined population flags.

Table 3.2.3 ADSL Population Indicator Variables

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
FASFL	Full Analysis Set Population Flag	Char	Y, N	Cond	These flags identify whether or not the subject is included in the specified population. A minimum of one subject-level population flag variable is required in ADSL. Not all of the indicators listed here need to be included in ADSL. As stated in Section 3.1.4, Item 2, only those indicators corresponding to populations defined in the statistical analysis plan or populations used as a basis for analysis need be included in ADSL. This list of flags is not meant to be all-inclusive. Additional population flags may be added. The values of subject-level population flags cannot be blank. If a flag is used, the corresponding numeric version (*FN, where 0=no and 1=yes) of the population flag can also be included. Please also refer to Section 3.1.4.
SAFFL	Safety Population Flag	Char	Y, N	Cond	
ITTFL	Intent-To-Treat Population Flag	Char	Y, N	Cond	
PPROTFL	Per-Protocol Population Flag	Char	Y, N	Cond	
COMPLFL	Completers Population Flag	Char	Y, N	Cond	
RANDFL	Randomized Population Flag	Char	Y, N	Cond	
ENRLFL	Enrolled Population Flag	Char	Y, N	Cond	

Table 3.2.4 ADSL Treatment Variables

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
ARM	Description of Planned Arm	Char		Req	DM.ARM
ACTARM	Description of Actual Arm	Char		Perm	DM.ACTARM
TRTxxP	Planned Treatment for Period xx	Char		Req	Subject-level identifier that represents the planned treatment for period xx. In a one-period randomized trial, TRT01P would be the treatment to which the subject was randomized. TRTxxP might be derived from the SDTM DM variable ARM. At least TRT01P is required.
TRTxxPN	Planned Treatment for Period xx (N)	Num		Perm	The numeric code variable for TRTxxP. One-to-one mapping to TRTxxP within a study.
TRTxxA	Actual Treatment for Period xx	Char		Cond	Subject-level identifier that represents the actual treatment for the subject for period xx. Required when actual treatment does not match planned and there is an analysis of the data as treated.
TRTxxAN	Actual Treatment for Period xx (N)	Num		Perm	The numeric code variable for TRTxxA. One-to-one mapping to TRTxxA within a study.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
TRTSEQP	Planned Sequence of Treatments	Char		Cond	Required when there is an analysis based on the sequence of treatments, for example in a crossover design. TRTSEQP is not necessarily equal to ARM, for example if ARM contains elements that are not relevant to analysis of treatments or ARM is not fully descriptive (e.g., “GROUP 1,” “GROUP 2”). When analyzing based on the sequence of treatments, TRTSEQP is required even if identical to ARM.
TRTSEQPN	Planned Sequence of Treatments (N)	Num		Perm	Numeric version of TRTSEQP. One-to-one mapping to TRTSEQP within a study.
TRTSEQA	Actual Sequence of Treatments	Char		Cond	TRTSEQA is required if a situation occurred in the conduct of the trial where a subject received a sequence of treatments other than what was planned and there is an analysis based on the sequence of treatments.
TRTSEQAN	Actual Sequence of Treatments (N)	Num		Perm	Numeric version of TRTSEQA. One-to-one mapping to TRTSEQA within a study.
TRxxPGy	Planned Pooled Treatment y for Period xx	Char		Perm	Planned pooled treatment y for period xx. Useful when planned treatments (TRTxxP) in the specified period xx are pooled together for analysis according to pooling algorithm y. For example when in period 2 the first pooling algorithm dictates that all doses of Drug A (TR02PG1=“All doses of Drug A”) are pooled together for comparison to all doses of Drug B (TR02PG1=“All doses of Drug B”). Each value of TRTxxP is pooled within at most one value of TRxxPGy.
TRxxPGyN	Planned Pooled Trt y for Period xx (N)	Num *		Perm	The numeric code for TRxxPGy. One-to-one mapping to TRxxPGy within a study.
TRxxAGy	Actual Pooled Treatment y for Period xx	Char		Cond	Actual pooled treatment y for period xx. Required when TRxxPGy is present and TRTxxA is present.
TRxxAGyN	Actual Pooled Trt y for Period xx (N)	Num *		Perm	The numeric code for TRxxAGy. One-to-one mapping to TRxxAGy within a study.
TSEQPGy	Planned Pooled Treatment Sequence y	Char		Perm	Planned pooled treatment sequence y. Useful when planned treatment sequences (TRTSEQP) are pooled together for analysis according to pooling algorithm y. For example, this might be used in an analysis of an extension study when the analysis is based on what the subject received in the parent study as well as in the extension study.
TSEQPGyN	Planned Pooled Treatment Sequence y (N)	Num		Perm	Numeric version of TSEQPGy. One-to-one mapping to TSEQPGy within a study.
TSEQAGy	Actual Pooled Treatment Sequence y	Char		Cond	Actual pooled treatment sequence y. Required when TSEQPGy is present and TRTSEQA is present.
TSEQAGyN	Actual Pooled Treatment Sequence y (N)	Num		Perm	Numeric version of TSEQAGy. One-to-one mapping to TSEQAGy within a study.
** TRxxPGyN and TRxxAGyN were mistakenly indicated as character variables in ADaMIG v1.0. The error is corrected above.*					

Table 3.2.5 describes ADSL dose variables, which are used to describe dosage amount. These variables can only be used in addition to, not instead of, the ADSL treatment variables in Table 3.2.4. It is permitted to include dosing information in both the treatment variables and in the dosing variables.

Table 3.2.5 ADSL Dose Variables

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
DOSExxP	Planned Treatment Dose for Period xx	Num		Perm	Subject-level identifier that represents the planned treatment dosage for period xx.
DOSExxA	Actual Treatment Dose for Period xx	Num		Perm	Subject-level identifier that represents the actual treatment dosage for period xx.
DOSExxU	Units for Dose for Period xx	Char		Perm	The units for DOSExxP and DOSExxA. It is permissible to use suffixes such as "P" and "A" if needed, with labels modified accordingly.

Table 3.2.6 ADSL Treatment Timing Variables

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
TRTSDT	Date of First Exposure to Treatment	Num		Cond	Date of first exposure to treatment for a subject in a study. TRTSDT and/or TRTSDTM are required if there is an investigational product. Note that TRTSDT is not required to have the same value as the SDTM DM variable RFXSTDTC. While both of these dates reflect the concept of first exposure, the ADaM date may be derived to support the analysis which may not necessarily be the very first date in the SDTM EX domain.
TRTSTM	Time of First Exposure to Treatment	Num		Perm	Time of first exposure to treatment for a subject in a study.
TRTSDTM	Datetime of First Exposure to Treatment	Num		Cond	Datetime of first exposure to treatment for a subject in a study. TRTSDT and/or TRTSDTM are required if there is an investigational product.
TRTSDTF	Date of First Exposure Imput. Flag	Char	(DATEFL)	Cond	The level of imputation of date of first exposure to treatment. If TRTSDT (or the date part of TRTSDTM) was imputed, TRTSDTF must be populated and is required. See Section 3.1.3.
TRTSTMF	Time of First Exposure Imput. Flag	Char	(TIMEFL)	Cond	The level of imputation of time of first exposure to treatment. If TRTSTM (or the time part of TRTSDTM) was imputed, TRTSTMF must be populated and is required. See Section 3.1.3.
TRTEDT	Date of Last Exposure to Treatment	Num		Cond	Date of last exposure to treatment for a subject in a study. TRTEDT and/or TRTEDTM are required if there is an investigational product. Note that TRTEDT is not required to have the same value as the SDTM DM variable RFXENDTC. While both of these dates reflect the concept of last exposure, the ADaM date may be derived to support the analysis which may not necessarily be the very last date in the SDTM EX domain.
TRTETM	Time of Last Exposure to Treatment	Num		Perm	Time of last exposure to treatment for a subject in a study.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
TRTEDTM	Datetime of Last Exposure to Treatment	Num		Cond	Datetime of last exposure to treatment for a subject in a study. TRTEDT and/or TRTEDTM are required if there is an investigational product.
TRTEDTF	Date of Last Exposure Imput. Flag	Char	(DATEFL)	Cond	The level of imputation of date of last exposure to treatment. If TRTEDT (or the date part of TRTEDTM) was imputed, TRTEDTF must be populated and is required. See Section 3.1.3.
TRTETMF	Time of Last Exposure Imput. Flag	Char	(TIMEFL)	Cond	The level of imputation of time of last exposure to treatment. If TRTETM (or the time part of TRTEDTM) was imputed, TRTETMF must be populated and is required. See Section 3.1.3.
TRxxSDT	Date of First Exposure in Period xx	Num		Cond	Date of first exposure to treatment in period xx. TRxxSDT and/or TRxxSDTM are only required in trial designs where multiple treatments are given to the same subject, such as a crossover design, but are permissible for other trial designs. Also useful in designs where multiple periods exist for the same treatment (i.e., multiple cycles of the same study treatment).
TRxxSTM	Time of First Exposure in Period xx	Num		Cond	The starting time of exposure to treatment in period xx. TRxxSTM and/or TRxxSDTM are only required in trial designs where multiple treatments are given to the same subject, such as a crossover design (but are permissible for other trial designs), and time is important to the analysis.
TRxxSDTM	Datetime of First Exposure in Period xx	Num		Cond	Datetime of first exposure to treatment in period xx. TRxxSDT and/or TRxxSDTM are only required in trial designs where multiple treatments are given to the same subject, such as a crossover design, but are permissible for other trial designs.
TRxxSDTF	Date 1st Exposure Period xx Imput. Flag	Char	(DATEFL)	Cond	The level of imputation of date of first exposure to treatment in period xx. If TRxxSDT (or the date part of TRxxSDTM) was imputed, TRxxSDTF must be populated and is required. See Section 3.1.3.
TRxxSTMF	Time 1st Exposure Period xx Imput. Flag	Char	(TIMEFL)	Cond	The level of imputation of time of first exposure to treatment in period xx. If TRxxSTM (or the time part of TRxxSDTM) was imputed, TRxxSTMF must be populated and is required. See Section 3.1.3.
TRxxEDT	Date of Last Exposure in Period xx	Num		Cond	Date of last exposure to treatment in period xx. TRxxEDT and/or TRxxEDTM are only required in trial designs where multiple treatments are given to the same subject, such as a crossover design, but are permissible for other trial designs.
TRxxETM	Time of Last Exposure in Period xx	Num		Cond	The ending time of exposure to treatment in period xx. TRxxETM and/or TRxxEDTM are only required in trial designs where multiple treatments are given to the same subject, such as a crossover design, and ending time is important to the analysis, but are permissible for other trial designs.
TRxxEDTM	Datetime of Last Exposure in Period xx	Num		Cond	The datetime of last exposure to treatment in period xx. TRxxEDT and/or TRxxEDTM are only required in trial designs where multiple treatments are given to the same subject, such as a crossover design, but are permissible for other trial designs.
TRxxEDTF	Date Last Exposure Period xx Imput. Flag	Char	(DATEFL)	Cond	The level of imputation of date of last exposure to treatment in period xx. If TRxxEDT (or the date part of TRxxEDTM) was imputed, TRxxEDTF must be populated and is required. See Section 3.1.3.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
TRxxETMF	Time Last Exposure Period xx Imput. Flag	Char	(TIMEFL)	Cond	The level of imputation of time of last exposure to treatment in period xx. If TRxxETM (or the time part of TRxxEDTM) was imputed, TRxxETMF must be populated and is required. See Section 3.1.3.

Additional timing variables can be included for phase, period, and subperiod (APHASE, APERIOD, and ASPER are defined in Table 3.3.3.1). Table 3.2.7 provides the subject-level variables for these timing elements.

The following provisions apply to the inclusion or exclusion of sets of pairs of subject-level timing variables in ADSL (i.e., the pair of start and end variables for each of the timing elements in the study (e.g. APxxSDT and APxxEDT for each period in the study)). A set of timing variables for a specific timing element (i.e., phase, period, or subperiod) includes only those variables from Table 3.2.7 that are applicable to the study. For example, although the period start time is defined in the table below, it should be included in the set of period timing variables only if needed for the study.

- A set of timing variables can be included in ADSL only if the definitions for all of the variables in the set are fixed across the study (i.e., the definitions of the start and end of each timing element for a given subject do not change based on endpoint or data type).
- If any of the definitions of the variables in the set do vary, for example, when analysis period start and stop date definitions differ for safety and efficacy analyses, then none of the variables in the set can be included in ADSL.
- If none of the variable definitions in the set vary, then the full set of variables can be included in ADSL (i.e., either the full set is included or none of the variables in the set are included).

Table 3.2.7 Subject-Level Period, Subperiod, and Phase Timing Variables

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
APxxSDT	Period xx Start Date	Num		Perm	The starting date of period xx.
APxxSTM	Period xx Start Time	Num		Perm	The starting time of period xx.
APxxSDTM	Period xx Start Datetime	Num		Perm	The starting datetime of period xx.
APxxSDTF	Period xx Start Date Imput. Flag	Char	(DATEFL)	Cond	The level of imputation of period xx start date. See Section 3.1.3.
APxxSTMF	Period xx Start Time Imput. Flag	Char	(TIMEFL)	Cond	The level of imputation of period xx start time. See Section 3.1.3.
APxxEDT	Period xx End Date	Num		Perm	The ending date of period xx.
APxxETM	Period xx End Time	Num		Perm	The ending time of period xx.
APxxEDTM	Period xx End Datetime	Num		Perm	The ending datetime of period xx.
APxxEDTF	Period xx End Date Imput. Flag	Char	(DATEFL)	Cond	The level of imputation of period xx end date. See Section 3.1.3.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
APxxETMF	Period xx End Time Imput. Flag	Char	(TIMEFL)	Cond	The level of imputation of period xx end time. See Section 3.1.3.
PxxSw	Description of Period xx Subperiod w	Char		Perm	Description of analysis subperiod w within period xx.
PxxSwSDT	Period xx Subperiod w Start Date	Num		Perm	The starting date of subperiod w within period xx.
PxxSwSTM	Period xx Subperiod w Start Time	Num		Perm	The starting time of subperiod w within period xx.
PxxSwSDM	Period xx Subperiod w Start Datetime	Num		Perm	The starting datetime of subperiod w within period xx.
PxxSwSDF	Period xx Subper w Start Date Imput Flag	Char	(DATEFL)	Cond	The level of imputation of the start date for subperiod w within period xx. See Section 3.1.3.
PxxSwSTF	Period xx Subper w Start Time Imput Flag	Char	(TIMEFL)	Cond	The level of imputation of the start time for subperiod w within period xx. See Section 3.1.3.
PxxSwEDT	Period xx Subperiod w End Date	Num		Perm	The ending date of subperiod w within period xx.
PxxSwETM	Period xx Subperiod w End Time	Num		Perm	The ending time of subperiod w within period xx.
PxxSwEDM	Period xx Subperiod w End Datetime	Num		Perm	The ending datetime of subperiod w within period xx.
PxxSwEDF	Period xx Subper w End Date Imput Flag	Char	(DATEFL)	Cond	The level of imputation of the end date for subperiod w within period xx. See Section 3.1.3.
PxxSwETF	Period xx Subper w End Time Imput Flag	Char	(TIMEFL)	Cond	The level of imputation of the end time for subperiod w within period xx. See Section 3.1.3.
APHASEw	Description of Phase w	Char		Perm	Description of analysis phase w.
PHwSDT	Phase w Start Date	Num		Perm	The starting date of phase w.
PHwSTM	Phase w Start Time	Num		Perm	The starting time of phase w.
PHwSDTM	Phase w Start Datetime	Num		Perm	The starting datetime of phase w.
PHwSDTF	Phase w Start Date Imputation Flag	Char	(DATEFL)	Cond	The level of imputation of the start date for phase w. See Section 3.1.3.
PHwSTMF	Phase w Start Time Imputation Flag	Char	(TIMEFL)	Cond	The level of imputation of the start time for phase w. See Section 3.1.3.
PHwEDT	Phase w End Date	Num		Perm	The ending date of phase w.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
PHwETM	Phase w End Time	Num		Perm	The ending time of phase w.
PHwEDTM	Phase w End Datetime	Num		Perm	The ending datetime of phase w.
PHwEDTF	Phase w End Date Imputation Flag	Char	(DATEFL)	Cond	The level of imputation of the end date for phase w. See Section 3.1.3.
PHwETMF	Phase w End Time Imputation Flag	Char	(TIMEFL)	Cond	The level of imputation of the end time for phase w. See Section 3.1.3.

Table 3.2.8 ADSL Subject-Level Trial Experience Variables

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
EOSSTT	End of Study Status	Char		Perm	The subject’s status as of the end of study or data cutoff. Examples: COMPLETED, DISCONTINUED, ONGOING.
EOSDT	End of Study Date	Num		Perm	Date subject ended the study – either date of completion or date of discontinuation or data cutoff date for interim analyses.
DCSREAS	Reason for Discontinuation from Study	Char		Perm	Reason for subject’s discontinuation from study. The source would most likely be the SDTM DS dataset. Null for subjects who completed the study.
DCSREASP	Reason Spec for Discont from Study	Char		Perm	Additional detail regarding subject’s discontinuation from study (e.g., description of “other”).
EOTSTT	End of Treatment Status	Char		Perm	The subject’s status as of the end of treatment or data cutoff. Examples: COMPLETED, DISCONTINUED, ONGOING.
DCTREAS	Reason for Discontinuation of Treatment	Char		Perm	If a subject discontinued treatment in the study, then this variable indicates the reason for discontinuation. This is for discontinuation of treatment in the overall study and not to be used for discontinuation reason within individual treatment periods.
DCTREASP	Reason Specify for Discont of Treatment	Char		Perm	Additional detail regarding subject’s discontinuation from treatment (e.g., description of “other”).
EOTxxSTT	End of Treatment Status in Period xx	Char		Perm	The subject’s treatment status as of the end of period xx, or data cutoff if within period xx. Examples: COMPLETED, DISCONTINUED, ONGOING.
DCTxxRS	Reason for Discont of Treat in Period xx	Char		Perm	Reason for discontinuing treatment in period xx.
DCTxxRSP	Reason Spec for Disc of Trt in Period xx	Char		Perm	Additional detail regarding subject’s discontinuation of treatment in period xx (e.g., description of “other”).
EOPxxSTT	End of Period xx Status	Char		Perm	The subject’s status as of the end of period xx, or data cutoff if within period xx. Examples: COMPLETED, DISCONTINUED, ONGOING.
DCPxxRS	Reason for Discont from Period xx	Char		Perm	Reason for discontinuing analysis period xx.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
DCPxxRSP	Reason Spec for Discont from Period xx	Char		Perm	Additional detail regarding subject’s discontinuation from period xx (e.g., description of “other”).
RFICDT	Date of Informed Consent	Num		Perm	Date subject gave informed consent. Generally equivalent to DM.RFICDTC.
ENRLDT	Date of Enrollment	Num		Perm	Date of subject’s enrollment into trial.
RANDDT	Date of Randomization	Num		Cond	Required in randomized trials.
RFICyDT	Date of Informed Consent y	Num		Perm	This variable may be used in the case where there are multiple consent dates within a study. This date does not need to repeat the date in RFICDT. ‘y’ can start with 1 but it is not required to start with 1.
ENRLyDT	Date of Enrollment y	Num		Perm	This variable may be used in the case where there are multiple enrollment dates within a study. This date does not need to repeat the date in ENRLDT. ‘y’ can start with 1 but it is not required to start with 1.
RANDyDT	Date of Randomization y	Num		Perm	This variable may be used in the case where there are multiple randomization dates within a study. This date does not need to repeat the date in RANDDT. ‘y’ can start with 1 but it is not required to start with 1.
LSTALVDT	Date Last Known Alive	Num		Perm	If this variable is included in ADSL, the best practice is to populate it for everyone. If the derivation for subjects who died differs from the derivation for subjects who are not known to have died, the differences should be noted in metadata.
TRCMP	Treatment Compliance (%)	Num		Perm	Overall percent compliance with treatment in the trial. TRCMP may be useful for inclusion in ADSL for reasons such as defining subgroups and/or populations.
TRCMPGy	Treatment Compliance (%) Group y	Char		Perm	Grouping ‘y’ of TRCMP, treatment compliance percentage.
TRCMPGyN	Treatment Compliance (%) Group y (N)	Num		Perm	Numeric version of treatment compliance (%) grouping ‘y’. Must have a one-to-one mapping to TRCMPGy.
TRxxDURD	Treatment Duration in Period xx (Days)	Num		Perm	Treatment duration for period xx as measured in days. More than one of TRxxDURD, TRxxDURM, and TRxxDURY can be populated, but each represents the entire duration in its respective units.
TRxxDURM	Treatment Duration in Period xx (Months)	Num		Perm	Treatment duration for period xx, as measure in months. More than one of TRxxDURD, TRxxDURM, and TRxxDURY can be populated, but each represents the entire duration in its respective units.
TRxxDURY	Treatment Duration in Period xx (Years)	Num		Perm	Treatment duration for period xx, as measured in years. More than one of TRxxDURD, TRxxDURM, and TRxxDURY can be populated, but each represents the entire duration in its respective units.
TRTDURD	Total Treatment Duration (Days)	Num		Perm	Total treatment duration, as measured in days. More than one of TRTDURD, TRTDURM, and TRTDURY can be populated, but each represents the entire duration in its respective units.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
TRTDURM	Total Treatment Duration (Months)	Num		Perm	Total treatment duration, as measured in months. More than one of TRTDURD, TRTDURM, and TRTDURY can be populated, but each represents the entire duration in its respective units.
TRTDURY	Total Treatment Duration (Years)	Num		Perm	Total treatment duration, as measured in years. More than one of TRTDURD, TRTDURM, and TRTDURY can be populated, but each represents the entire duration in its respective units.
DTHDT	Date of Death	Num		Perm	Date of subject's death. Derived from DM.DTHDTC.
DTHDTF	Date of Death Imputation Flag	Char		Cond	Imputation flag for date of subject's death. If DTHDT was imputed, DTHDTF must be populated and is required. See Section 3.1.3.
DTHCAUS	Cause of Death	Char		Perm	Cause of Death.
DTHCAUSN	Cause of Death (N)	Num		Perm	Numeric representation of cause of death. Must have a one-to-one mapping to DTHCAUS.
DTHCGRy	Cause of Death Group y	Char		Perm	Grouping 'y' of DTHCAUS, the subject's cause of death.
DTHCGRyN	Cause of Death Group y (N)	Num		Perm	Numeric version of grouping 'y' of the subject's cause of death. Must have a one-to-one mapping to DTHCGRy.

3.3 ADaM Basic Data Structure (BDS) Variables

The ADaM model document introduces the ADaM Basic Data Structure. A BDS dataset contains one or more records per subject, per analysis parameter, per analysis timepoint. Analysis timepoint is conditionally required, depending on the analysis. In situations where there is no analysis timepoint, the structure is one or more records per subject per analysis parameter. Typically there are several BDS datasets in a study. This section of the ADaMIG defines the standard variables used in BDS datasets. See Section 3.2 for ADSL variables, any of which may be copied to BDS datasets to support traceability or enable analysis.

Within Section 3.3, "within a given study, subject, and dataset" is implied, unless otherwise stated. For example, the description of ABLFL defines it as a variable that indicates baseline record for each parameter, or if there is more than one baseline definition, for each parameter and baseline type (BASETYPE). It should be understood that the baseline record is for the subject identified by USUBJID. In addition, note that "within a parameter" means "within a parameter within a dataset."

3.3.1 Identifier Variables for BDS Datasets

Table 3.3.1.1 Identifier Variables for BDS Datasets

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
STUDYID	Study Identifier	Char		Req	DM.STUDYID
USUBJID	Unique Subject Identifier	Char		Req	DM.USUBJID

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
SUBJID	Subject Identifier for the Study	Char		Perm	DM.SUBJID. SUBJID is required in ADSL, but permissible in other datasets.
SITEID	Study Site Identifier	Char		Perm	DM.SITEID. SITEID is required in ADSL, but permissible in other datasets.
ASEQ	Analysis Sequence Number	Num		Perm	Sequence number given to ensure uniqueness of subject records within an ADaM dataset. As long as values are unique within a subject within the dataset, any valid number can be used for ASEQ. ASEQ uniquely indexes records within a subject within an ADaM dataset. ASEQ is useful for traceability when the dataset is used as input to another ADaM dataset. To refer to a record in a predecessor ADaM dataset, set SRCDOM to the name of the predecessor dataset, and set SRCSEQ to the value of ASEQ in the predecessor dataset.

3.3.2 Record-Level Treatment and Dose Variables for BDS Datasets

At least one treatment variable is required in a BDS dataset. This requirement is satisfied by any of the subject-level or record-level treatment variables (e.g. TRTxxP or TRTP). One is allowed to use any treatment variable in analysis of BDS. Any subject-level treatment variable may be copied into the BDS dataset from ADSL. In addition, record-level treatment variables, as defined in this section, may be used for analysis. See Section 4.1 for examples of treatment variables.

All treatment variables defined in Table 3.3.2.1 are record-level. This means that the values contained in the variable may vary by record within a subject.

Table 3.3.2.1 Record-Level Treatment Variables for BDS Datasets

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
TRTP	Planned Treatment	Char		Cond	TRTP is a record-level identifier that represents the planned treatment attributed to a record for analysis purposes. TRTP indicates how treatment varies by record within a subject and enables analysis of crossover and other designs. Though there is no requirement that TRTP will correspond to the TRTxxP as defined by the record's value of APERIOD, if populated, TRTP must match at least one value of the character planned treatment variables in ADSL (e.g., TRTxxP, TRTSEQP, TRxxPGy). As noted previously, at least one treatment variable is required even in non-randomized trials. This requirement is satisfied by any subject-level or record-level treatment variables (e.g., TRTxxP, TRTP, TRTA). Even if not used for analysis, any ADSL treatment variable may be included in the BDS dataset.
TRTPN	Planned Treatment (N)	Num		Perm	The numeric code for TRTP. One-to-one mapping within a study to TRTP.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
TRTA	Actual Treatment	Char		Cond	TRTA is a record-level identifier that represents the actual treatment attributed to a record for analysis purposes. TRTA indicates how treatment varies by record within a subject and enables analysis of crossover and other multi-period designs. Though there is no requirement that TRTA will correspond to the TRTxxA as defined by the record's value of APERIOD, TRTA must match at least one value of the character actual treatment variables in ADSL (e.g., TRTxxA, TRTSEQA, TRxxAGy). As noted previously, at least one treatment variable is required. This requirement is satisfied by any subject-level or record-level treatment variables (e.g., TRTxxP, TRTP, TRTA). Even if not used for analysis, any ADSL treatment variable may be included in the BDS dataset.
TRTAN	Actual Treatment (N)	Num		Perm	The numeric code for TRTA. One-to-one mapping within a study to TRTA.
TRTPGy	Planned Pooled Treatment y	Char		Perm	TRTPGy is the planned pooled treatment y attributed to a record for analysis purposes. "y" represents an integer [1-99, not zero-padded] corresponding to a particular pooling scheme. Useful when planned treatments (TRTP) are pooled together for analysis, for example when all doses of Drug A (TRTPG1=All doses of Drug A) are compared to all doses of Drug B (TRTPG1=All doses of Drug B). Each value of TRTP is pooled within at most one value of TRTPGy.
TRTPGyN	Planned Pooled Treatment y (N)	Num		Perm	The numeric code for TRTPGy. One-to-one mapping within a study to TRTPGy.
TRTAGy	Actual Pooled Treatment y	Char		Cond	TRTAGy is the actual pooled treatment y attributed to a record for analysis purposes. "y" represents an integer [1-99, not zero-padded] corresponding to a particular pooling scheme. Required when TRTPGy is present and TRTA is present.
TRTAGyN	Actual Pooled Treatment y (N)	Num		Perm	The numeric code for TRTAGy. One-to-one mapping within a study to TRTAGy.

All dose variables defined in Table 3.3.2.2 are record-level. This means that the values contained in the variable may vary by record within a subject. These record-level dose variables, plus subject-level dose variables copied from ADSL, can be used in addition to, but not instead of, treatment variables.

Table 3.3.2.2 Record-Level Dose Variables for BDS Datasets

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
DOSEP	Planned Treatment Dose	Num		Perm	DOSEP represents the planned treatment dosage associated with the record.
DOSCUMP	Cumulative Planned Treatment Dose	Num		Perm	Cumulative planned dosage of treatment for the subject at the point in time of the record (e.g., ADT).
DOSEA	Actual Treatment Dose	Num		Perm	DOSEA represents the actual treatment dosage associated with the record.
DOSCUMA	Cumulative Actual Treatment Dose	Num		Perm	Cumulative actual dosage of treatment for the subject at the point in time of the record (e.g., ADT).

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
DOSEU	Treatment Dose Units	Char		Perm	The units for DOSEP, DOSCUMP, DOSEA, and DOSCUMA. It is permissible to use suffixes such as “P” and “A” if needed, with labels modified accordingly.

3.3.3 Timing Variables for BDS Datasets

Any SDTM timing variables (including, but not limited to, EPOCH, --DTC, --DY, VISITNUM, VISIT, and VISITDY) may be copied into ADaM datasets if they would help to support data traceability and/or show how ADaM timing variables contrast with the SDTM data.

Table 3.3.3.1 defines analysis timing variables for BDS datasets. The timing variables whose names start with the letter “A” are the timing variables directly associated with the AVAL and AVALC variables in the ADaM dataset.

Timing variables (e.g., *DT) not directly characterizing AVAL should be prefixed by a character string instead of the placeholder asterisk shown in Table 3.3.3.1, so that their actual names comply with the variable naming conventions described in Section 3.1. In many cases, the prefix for these date and time variables would match that of an SDTM --DTC, --STDTC or --ENDTC variable name. For example, if a numeric date variable were created from --STDTC, then it would be named --SDT. However, if --DTC or --STDTC is the date that is associated with AVAL and AVALC, its numeric equivalent should be named ADT or ASTDT, as appropriate. The Timing Variable Conventions documented in Section 3.1.2 apply here as well.

Table 3.3.3.1 Timing Variables for BDS Datasets

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
ADT	Analysis Date	Num		Perm	The date associated with AVAL and/or AVALC in numeric format.
ATM	Analysis Time	Num		Perm	The time associated with AVAL and/or AVALC in numeric format.
ADTM	Analysis Datetime	Num		Perm	The datetime associated with AVAL and/or AVALC in numeric format.
ADY	Analysis Relative Day	Num		Perm	The relative day of AVAL and/or AVALC. The number of days from an anchor date (not necessarily DM.RFSTDTC) to ADT. See Section 3.1.2.
ADTF	Analysis Date Imputation Flag	Char	(DATEFL)	Cond	The level of imputation of analysis date. If ADT (or the date part of ADTM) was imputed, ADTF must be populated and is required. See Section 3.1.3.
ATMF	Analysis Time Imputation Flag	Char	(TIMEFL)	Cond	The level of imputation of analysis time. If ATM (or the time part of ADTM) was imputed, ATMF must be populated and is required. See Section 3.1.3.
ASTDT	Analysis Start Date	Num		Perm	The start date associated with AVAL and/or AVALC. ASTDT and AENDT may be useful for traceability when AVAL summarizes data collected over an interval of time, or when AVAL is a duration.
ASTTM	Analysis Start Time	Num		Perm	The start time associated with AVAL and/or AVALC. ASTTM and AENTM may be useful for traceability when AVAL summarizes data collected over an interval of time, or when AVAL is a duration.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
ASTDTM	Analysis Start Datetime	Num		Perm	The start datetime associated with AVAL and/or AVALC. ASTDTM and AENDTM may be useful for traceability when AVAL summarizes data collected over an interval of time, or when AVAL is a duration.
ASTDY	Analysis Start Relative Day	Num		Perm	The number of days from an anchor date (not necessarily DM.RFSTDTC) to ASTDT. See Section 3.1.2.
ASTDTF	Analysis Start Date Imputation Flag	Char	(DATEFL)	Cond	The level of imputation of analysis start date. If ASTDT (or the date part of ASTDTM) was imputed, ASTDTF must be populated and is required. See Section 3.1.3.
ASTTMF	Analysis Start Time Imputation Flag	Char	(TIMEFL)	Cond	The level of imputation of analysis start time. If ASTTM (or the time part of ASTDTM) was imputed, ASTTMF must be populated and is required. See Section 3.1.3.
AENDT	Analysis End Date	Num		Perm	The end date associated with AVAL and/or AVALC. See also ASTDT.
AENTM	Analysis End Time	Num		Perm	The end time associated with AVAL and/or AVALC. See also ASTTM.
AENDTM	Analysis End Datetime	Num		Perm	The end datetime associated with AVAL and/or AVALC. See also ASTDTM.
AENDY	Analysis End Relative Day	Num		Perm	The number of days from an anchor date (not necessarily DM.RFSTDTC) to AENDT. See Section 3.1.2.
AENDTF	Analysis End Date Imputation Flag	Char	(DATEFL)	Cond	The level of imputation of analysis end date. If AENDT (or the date part of AENDTM) was imputed, AENDTF must be populated and is required. See Section 3.1.3.
AENTMF	Analysis End Time Imputation Flag	Char	(TIMEFL)	Cond	The level of imputation of analysis end time. If AENTM (or the time part of AENDTM) was imputed, AENTMF must be populated and is required. See Section 3.1.3.
AVISIT	Analysis Visit	Char		Cond	The analysis visit description; required if an analysis is done by nominal, assigned or analysis visit. AVISIT may contain the visit names as observed (i.e., from SDTM VISIT), derived visit names, time window names, conceptual descriptions (such as Average, Endpoint, etc.), or a combination of any of these. AVISIT is a derived field and does not have to map to VISIT from the SDTM. 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. See Section 3.3.8 for information about flag variables. AVISIT should be unique for a given analysis visit window. In the event that a record does not fall within any predefined analysis timepoint window, AVISIT can be populated in any way that the producer chooses to indicate this fact (i.e., blank or "Not Windowed"). The way that AVISIT is calculated, including the variables used in its derivation, should be indicated in the variable metadata for AVISIT. The values and the rules for deriving AVISIT may be different for different parameters within the same dataset. Values of AVISIT are producer-defined, and are often directly usable in Clinical Study Report displays.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
AVISITN	Analysis Visit (N)	Num		Perm	A numeric representation of AVISIT. Since study visits are usually defined by certain timepoints, defining AVISITN so that it represents the timepoint associated with the visit can facilitate plotting and interpretation of the values. Alternatively, AVISITN may be a protocol visit number, a cycle number, an analysis visit number, or any other number logically related to AVISIT or useful for sorting that is needed for analysis. Within a parameter, there is a one-to-one mapping between AVISITN and AVISIT so that AVISITN has the same value for each distinct AVISIT. (Best practice would dictate that the mapping would be one-to-one within a study, but that is not an ADaM requirement.) In the event that a record does not fall within any predefined analysis timepoint window, AVISITN can be populated in any way that the producer chooses to indicate this fact (e.g., may be null). Values of AVISITN are producer-defined.
ATPT	Analysis Timepoint	Char		Cond	The analysis timepoint description; required if an analysis is done by nominal, assigned or analysis timepoint (instead of or in addition to by-visit). Timepoints are relative to ATPTREF. ATPT may contain the timepoint names as observed (i.e., from SDTM --TPT), derived timepoint names, time window names, conceptual descriptions (such as Average, Endpoint, etc.), or a combination of any of these. This variable is often used in conjunction with AVISIT. ATPT represents the analysis timepoint of the record. ATPT can be within an analysis visit (e.g., blood pressure assessments at 10 min, 20 min, and 30 min post-dose at AVISIT=Week 1) or can be unrelated to AVISIT (e.g., migraine symptoms 30 min, 60 min, and 120 min post-dose for attack 1). The way that ATPT is calculated, including the variables used in its derivation, should be indicated in the variable metadata for ATPT. The values and the rules for deriving ATPT may be different for different parameters within the same dataset. Values of ATPT are producer-defined, and are often directly usable in Clinical Study Report displays.
ATPTN	Analysis Timepoint (N)	Num		Perm	ATPTN provides a numeric representation of ATPT. Defining ATPTN so that its values represent the planned timepoints (e.g., minutes or hours after dosing) is not required but can facilitate plotting and interpretation of the values. Within the same parameter, there is a one-to-one mapping between ATPT and ATPTN. (Best practice would dictate that the mapping would be one-to-one within a study, but that is not an ADaM requirement.)
ATPTREF	Analysis Timepoint Reference	Char		Perm	Description of the fixed reference point referred to by ATPT/ATPTN (e.g., time of dose).
APHASE	Phase	Char		Perm	APHASE is a categorization of timing within a study, for example a higher-level categorization of APERIOD or an analysis epoch. For example, APHASE could describe spans of time for SCREENING, ON TREATMENT, and FOLLOW-UP.
APHASEN	Phase (N)	Num		Perm	APHASEN provides a numeric representation of APHASE. Within a study, there is a one-to-one mapping between APHASE and APHASEN.
APERIOD	Period	Num		Perm	APERIOD is a record-level timing variable that represents the analysis period within the study associated with the record for analysis purposes. The value of APERIOD (if populated) must be one of the xx values found in the ADSL TRTxxP variables.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
APERIODC	Period (C)	Char		Perm	Text characterizing to which analysis period the record belongs. One-to-one mapping within a dataset to APERIOD.
ASPER	Subperiod within Period	Num		Perm	The numeric value characterizing a sublevel within APERIOD to which the record belongs. Within each APERIOD, the first ASPER is 1 (i.e., it resets to 1 when the APERIOD value changes).
ASPERC	Subperiod within Period (C)	Char		Perm	Text characterizing to which subperiod the record belongs. One-to-one mapping within a period to ASPER.
ARELTM	Analysis Relative Time	Num		Perm	The time relative to an anchor time. The amount of time from an anchor time to ATM. When ARELTM is present, the anchor time variable and ARELTMU must also be included in the dataset, and the anchor time variable must be identified in the metadata for ARELTM.
ARELTMU	Analysis Relative Time Unit	Char		Perm	The units of ARELTM. For example, “HOURS” or “MINUTES.” ARELTMU is required if ARELTM is present.

Additional timing variables can be included for phase, period, and subperiod. Table 3.3.3.2 provides the record-level variables for these timing elements.

Table 3.3.3.2 Period, Subperiod, and Phase Start and End Timing Variables

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
APERSDT	Period Start Date	Num		Perm	The starting date for the period defined by APERIOD.
APERSTM	Period Start Time	Num		Perm	The starting time for the period defined by APERIOD.
APERSDTM	Period Start Datetime	Num		Perm	The starting datetime for the period defined by APERIOD.
APERSDTF	Period Start Date Imput. Flag	Char	(DATEFL)	Cond	The level of imputation of period start date. If APERSDT (or the date part of APERSDTM) was imputed, APERSDTF must be populated and is required. See Section 3.1.3.
APERSTMF	Period Start Time Imput. Flag	Char	(TIMEFL)	Cond	The level of imputation of period start time. If APERSTM (or the time part of APERSDTM) was imputed, APERSTMF must be populated and is required. See Section 3.1.3.
APEREDT	Period End Date	Num		Perm	The ending date for the period defined by APERIOD.
APERETM	Period End Time	Num		Perm	The ending time for the period defined by APERIOD.
APEREDTM	Period End Datetime	Num		Perm	The ending datetime for the period defined by APERIOD.
APEREDTF	Period End Date Imput. Flag	Char	(DATEFL)	Cond	The level of imputation of period end date. If APEREDT (or the date part of APEREDTM) was imputed, APEREDTF must be populated and is required. See Section 3.1.3.
APERETMF	Period End Time Imput. Flag	Char	(TIMEFL)	Cond	The level of imputation of period end time. If APERETM (or the time part of APEREDTM) was imputed, APERETMF must be populated and is required. See Section 3.1.3.
ASPRSDT	Subperiod Start Date	Num		Perm	The starting date for the subperiod defined by ASPER.
ASPRSTM	Subperiod Start Time	Num		Perm	The starting time for the subperiod defined by ASPER.
ASPRSDTM	Subperiod Start Datetime	Num		Perm	The starting datetime for the subperiod defined by ASPER.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
ASPRSDTF	Subperiod Start Date Imput. Flag	Char	(DATEFL)	Cond	The level of imputation of subperiod start date. If ASPRSDT (or the date part of ASPRSDTM) was imputed, ASPRSDTF must be populated and is required. See Section 3.1.3.
ASPRSTMF	Subperiod Start Time Imput. Flag	Char	(TIMEFL)	Cond	The level of imputation of subperiod start time. If ASPRSTM (or the time part of ASPRSDTM) was imputed, ASPRSTMF must be populated and is required. See Section 3.1.3.
ASPREDT	Subperiod End Date	Num		Perm	The ending date for the subperiod defined by ASPER.
ASPRETM	Subperiod End Time	Num		Perm	The ending time for the subperiod defined by ASPER.
ASPREDTM	Subperiod End Datetime	Num		Perm	The ending datetime for the subperiod defined by ASPER.
ASPREDTF	Subperiod End Date Imput. Flag	Char	(DATEFL)	Cond	The level of imputation of subperiod end date. If ASPREDT (or the date part of ASPREDTM) was imputed, ASPREDTF must be populated and is required. See Section 3.1.3.
ASPRETMF	Subperiod End Time Imput. Flag	Char	(TIMEFL)	Cond	The level of imputation of subperiod end time. If ASPRETM (or the time part of ASPREDTM) was imputed, ASPRETMF must be populated and is required. See Section 3.1.3.
PHSDT	Phase Start Date	Num		Perm	The starting date for the phase defined by APHASE.
PHSTM	Phase Start Time	Num		Perm	The starting time for the phase defined by APHASE.
PHSDTM	Phase Start Datetime	Num		Perm	The starting datetime for the phase defined by APHASE.
PHSDTF	Phase Start Date Imput. Flag	Char	(DATEFL)	Cond	The level of imputation of phase start date. If PHSDT (or the date part of PHSDTM) was imputed, PHSDTF must be populated and is required. See Section 3.1.3.
PHSTMF	Phase Start Time Imput. Flag	Char	(TIMEFL)	Cond	The level of imputation of phase start time. If PHSTM (or the time part of PHSDTM) was imputed, PHSTMF must be populated and is required. See Section 3.1.3.
PHEDT	Phase End Date	Num		Perm	The ending date for the phase defined by APHASE.
PHETM	Phase End Time	Num		Perm	The ending time for the phase defined by APHASE.
PHEDTM	Phase End Datetime	Num		Perm	The ending datetime for the phase defined by APHASE.
PHEDTF	Phase End Date Imput. Flag	Char	(DATEFL)	Cond	The level of imputation of phase end date. If PHEDT (or the date part of PHEDTM) was imputed, PHEDTF must be populated and is required. See Section 3.1.3.
PHETMF	Phase End Time Imput. Flag	Char	(TIMEFL)	Cond	The level of imputation of phase end time. If PHETM (or the time part of PHEDTM) was imputed, PHETMF must be populated and is required. See Section 3.1.3.

Table 3.3.3.3 lists suffixes that can be used for timing variables that are not directly descriptive of the analysis value (AVAL and/or AVALC) but may be included for support of review. There may be a number of “sets” of these variables as indicated by the “*” prefix. **See Timing Variable Conventions Item 11 for important cautions regarding the “*” prefix.** See Section 3.1.6 regarding labels for variables where bracketed words or phrases have been specified.

Table 3.3.3.3 Suffixes for User-Defined Timing Variables in BDS Datasets

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
*DT	{Date}	Num		Perm	Analysis date not directly characterizing AVAL and/or AVALC in numeric format.
*TM	{Time}	Num		Perm	Analysis time not directly characterizing AVAL and/or AVALC in numeric format.
*DTM	{Datetime}	Num		Perm	Analysis datetime not directly characterizing AVAL and/or AVALC in numeric format.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
*ADY	{Relative Day}	Num		Perm	Analysis relative day not directly characterizing AVAL and/or AVALC.
*DTF	{Date Imputation Flag}	Char	(DATEFL)	Cond	The level of imputation of *DT. If *DT (or the date part of *DTM) was imputed, *DTF must be populated and is required. See Section 3.1.3.
*TMF	{Time Imputation Flag}	Char	(TIMEFL)	Cond	The level of imputation of *TM. If *TM (or the time part of *DTM) was imputed, *TMF must be populated and is required. See Section 3.1.3.
*SDT	{Start Date}	Num		Perm	Starting analysis date not directly characterizing AVAL and/or AVALC in numeric format.
*STM	{Start Time}	Num		Perm	Starting analysis time not directly characterizing AVAL and/or AVALC in numeric format.
*SDTM	{Start Datetime}	Num		Perm	Starting analysis datetime not directly characterizing AVAL and/or AVALC in numeric format.
*SDY	{Relative Start Day}	Num		Perm	Starting analysis relative day not directly characterizing AVAL and/or AVALC.
*SDTF	{Start Date Imputation Flag}	Char	(DATEFL)	Cond	The level of imputation of *SDT. If *SDT (or the date part of *SDTM) was imputed, *SDTF must be populated and is required. See Section 3.1.3.
*STMF	{Start Time Imputation Flag}	Char	(TIMEFL)	Cond	The level of imputation of *STM. If *STM (or the time part of *SDTM) was imputed, *STMF must be populated and is required. See Section 3.1.3.
*EDT	{End Date}	Num		Perm	Ending analysis date not directly characterizing AVAL and/or AVALC in numeric format.
*ETM	{End Time}	Num		Perm	Ending analysis time not directly characterizing AVAL and/or AVALC in numeric format.
*EDTM	{End Datetime}	Num		Perm	Ending analysis datetime not directly characterizing AVAL and/or AVALC in numeric format.
*EDY	{Relative End Day}	Num		Perm	Ending analysis relative day not directly characterizing AVAL and/or AVALC.
*EDTF	{End Date Imputation Flag}	Char	(DATEFL)	Cond	The level of imputation of *EDT. If *EDT (or the date part of *EDTM) was imputed, *EDTF must be populated and is required. See Section 3.1.3.
*ETMF	{End Time Imputation Flag}	Char	(TIMEFL)	Cond	The level of imputation of *ETM. If *ETM (or the time part of *EDTM) was imputed, *ETMF must be populated and is required. See Section 3.1.3.

3.3.4 Analysis Parameter Variables for BDS Datasets

Table 3.3.4.1 Analysis Parameter Variables for BDS Datasets

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
PARAM	Parameter	Char		Req	The description of the analysis parameter. Examples include: “Supine Systolic Blood Pressure (mm Hg)”, “Log10 (Weight (kg))”, “Time to First Hypertension Event (Days)”, “Estimated Tumor Growth Rate”, etc. PARAM should be sufficient to describe unambiguously the contents of AVAL and/or AVALC. PARAM must include all descriptive and qualifying information relevant to the analysis purpose of the parameter. Examples of qualifying information that might be relevant to analysis and therefore are candidates for inclusion in PARAM are units, specimen type, location, position, machine type, and transformation function. There is no need to include qualifiers that are not relevant to the analysis of PARAM. In contrast to SDTM --TEST, no additional variable is needed to further qualify PARAM. PARAM may be longer than 40 characters in length but is restricted to a maximum of 200 characters. PARAM is often directly usable in Clinical Study Report displays. Note that in the ADaMIG, “parameter” is a synonym of “analysis parameter.”
PARAMCD	Parameter Code	Char		Req	The short name of the analysis parameter in PARAM. Values of PARAMCD should follow the SAS Version 5 transport file format and Oracle constraints as noted under General Variable Conventions in Section 3.1.1. There must be a one-to-one mapping to PARAM within a dataset.
PARAMN	Parameter (N)	Num		Perm	A numeric representation of PARAM. Useful for ordering and programmatic manipulation. There must be a one-to-one mapping to PARAM within a dataset.
PARAMTYP	Parameter Type	Char	(PARAMTYP)	Perm	Indicator of whether the parameter is derived as a function of one or more other parameters. This variable will be retired from the ADaMIG in the next version because it was confused with the concept of DTYPE and therefore was being misused. The variable metadata should be adequate to indicate when a parameter is wholly derived.
PARCATy	Parameter Category y	Char		Perm	A categorization of PARAM within a dataset. For example, values of PARCAT1 might group the parameters having to do with a particular questionnaire, lab specimen type, or area of investigation. Note that PARCATy is not a qualifier for PARAM. PARAM to PARCATy is a many-to-one mapping; any given PARAM may be associated with at most one level of PARCATy (e.g., one level of PARCAT1 and one level of PARCAT2).
PARCATyN	Parameter Category y (N)	Num		Perm	A numeric representation of PARCATy. This can be used for operations on PARCATy. There must be a one-to-one relationship within a dataset between PARCATy and PARCATyN.
AVAL	Analysis Value	Num		Cond	Numeric analysis value described by PARAM. On a given record, it is permissible for AVAL, AVALC, or both to be null. AVAL is required if AVALC is not present, since either AVAL or AVALC must be present in the dataset.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
AVALC	Analysis Value (C)	Char		Cond	Character analysis value described by PARAM. AVALC can be a character string mapping to AVAL, but if so there must be a one-to-one map between AVAL and AVALC within a given PARAM. AVALC should not be used to categorize the values of AVAL. Within a given parameter, if there exists a row on which both AVALC and AVAL are populated, then there must be a one-to-one mapping between AVALC and AVAL on all rows on which both variables are populated. (In other words, there is no requirement that records with a null value in either AVAL or AVALC be included when determining whether the one-to-one mapping requirement is satisfied.) On a given record, it is permissible for AVAL, AVALC, or both to be null. AVALC is required if AVAL is not present, since either AVAL or AVALC must be present in the dataset.
AVALCATy	Analysis Value Category y	Char		Perm	A categorization of AVAL or AVALC within a parameter. Not necessarily a one-to-one mapping to AVAL and/or AVALC. For example, if PARAM is “Headache Severity” and AVAL has values 0, 1, 2, or 3, AVALCAT1 can categorize AVAL into “None or Mild” (for AVAL 0 or 1) and “Moderate or Severe” (for AVAL 2 or 3).
AVALCAyN	Analysis Value Category y (N)	Num		Perm	A numeric representation of AVALCATy. This can be used for ordering of values of AVALCATy or for other purposes. There must be a one-to-one relationship within a parameter between AVALCATy and AVALCAyN.
BASE	Baseline Value	Num		Cond	The subject’s baseline analysis value for a parameter and baseline definition (i.e. BASETYPE) if present. BASE contains the value of AVAL copied from a record within the parameter on which ABLFL = “Y”. Required if dataset supports analysis or review of numeric baseline value or functions of numeric baseline value. If BASE is populated for a parameter, and BASE is non-null for a subject for that parameter, then there must be a record flagged by ABLFL for that subject and parameter. Note that a baseline record may be derived (e.g., it may be an average) in which case DTYPE must be populated on the baseline record.
BASEC	Baseline Value (C)	Char		Perm	The subject’s baseline value of AVALC for a parameter and baseline definition (i.e. BASETYPE) if present. May be needed when AVALC is of interest. BASEC contains the value of AVALC copied from a record within the parameter on which ABLFL = “Y”. If both AVAL and AVALC are populated within a parameter, the baseline record for AVALC must be the same record as that for AVAL. Within a given parameter, if there exists a row on which both BASEC and BASE are populated, then there must be a one-to-one mapping between BASEC and BASE on all rows on which both variables are populated. (In other words, there is no requirement that records with a null value in either BASE or BASEC be included when determining whether the one-to-one mapping requirement is satisfied.) On a given record, it is permissible for BASE, BASEC, or both to be null.
BASECATy	Baseline Category y	Char		Perm	A categorization of BASE or BASEC within a parameter. Not necessarily a one-to-one mapping to BASE or BASEC. For example, if PARAM is “Headache Severity” and AVAL has values 0, 1, 2, or 3, BASECAT1 can categorize BASE into “None or Mild” (for BASE 0 or 1) and “Moderate or Severe” (for BASE 2 or 3).

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
BASECAyN	Baseline Category y (N)	Num		Perm	A numeric representation of BASECATy. This can be used for ordering of values of BASECATy or for other purposes. There must be a one-to-one relationship within a parameter between BASECATy and BASECAyN.
BASETYPE	Baseline Type	Char		Cond	Producer-defined text describing the definition of baseline relevant to the value of BASE on the current record. Required when there are multiple ways that baseline is defined. If used for any PARAM within a dataset, should be non-null for all records of that dataset. Refer to Section 4.2.1, Rule 6, for an example.
CHG	Change from Baseline	Num		Perm	Change from baseline analysis value. Equal to AVAL-BASE. If used for a given PARAM, should be populated for all post-baseline records of that PARAM regardless of whether that record is used for analysis. The decision on how to populate pre-baseline and baseline values of CHG is left to producer choice.
CHGCATy	Change from Baseline Category y	Char		Perm	A categorization of CHG within a parameter. Not necessarily a one-to-one mapping to CHG. The definition of CHGCATy may vary by PARAM. For example, CHGCAT1 may be used to categorize CHG with respect to ranges of change in SYSBP; "-10 to -5 mm Hg", "-5 to 0 mm Hg" categories.
CHGCATyN	Change from Baseline Category y (N)	Num		Perm	A numeric representation of CHGCATy. This can be used for ordering of values of CHGCATy or for other purposes. There must be a one-to-one relationship within a parameter between CHGCATy and CHGCATyN.
PCHG	Percent Change from Baseline	Num		Perm	Percent change from baseline analysis value. Equal to ((AVAL-BASE)/BASE)*100. If used for a given PARAM, should be populated (when calculable) for all post-baseline records of that PARAM regardless of whether that record is used for analysis. The decision on how to populate pre-baseline and baseline values of PCHG is left to producer choice.
PCHGCATy	Percent Chg from Baseline Category y	Char		Perm	A categorization of PCHG within a parameter. Not necessarily a one-to-one mapping to PCHG. The definition of PCHGCATy may vary by PARAM. For example, PCHGCAT1 may be used to categorize PCHG with respect to ranges of change in SYSBP; ">5%", ">10%" categories.
PCHGCAyN	Percent Chg from Baseline Category y (N)	Num		Perm	A numeric representation of PCHGCATy. This can be used for ordering of values of PCHGCATy or for other purposes. There must be a one-to-one relationship within a parameter between PCHGCATy and PCHGCAyN.
R2BASE	Ratio to Baseline	Num		Perm	Ratio to the baseline value. Equal to AVAL / BASE. If used for a given PARAM, should be populated for all post-baseline records of that PARAM regardless of whether that record is used for analysis. The decision on how to populate pre-baseline and baseline values of R2BASE is left to producer choice.
R2AyLO	Ratio to Analysis Range y Lower Limit	Num		Perm	Ratio to the lower limit of the analysis range y. Equal to AVAL / AyLO. AyLO must exist in the ADaM dataset. If used for a given PARAM, should be populated for all post-baseline records of that PARAM regardless of whether that record is used for analysis. The decision on how to populate pre-baseline and baseline values of R2AyLO is left to producer choice.
R2AyHI	Ratio to Analysis Range y Upper Limit	Num		Perm	Ratio to the upper limit of the analysis range y. Equal to AVAL / AyHI. AyHI must exist in the ADaM dataset. If used for a given PARAM, should be populated for all post-baseline records of that PARAM regardless of whether that record is used for analysis. The decision on how to populate pre-baseline and baseline values of R2AyHI is left to producer choice.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
SHIFTy	Shift y	Char		Perm	A shift in values depending on the defined pairing for group y. SHIFTy can be based on the change in value of any of the following pairs (BASECATy, AVALCATy), (BNRIND, ANRIND), (ByIND, AyIND), (BTOXGR, ATOXGR), (BASE, AVAL) or (BASEC, AVALC). Useful for shift tables. For example, "NORMAL to HIGH". If used for a given PARAM, should be populated (when calculable) for all post-baseline records of that PARAM regardless of whether that record is used for analysis. The decision on how to populate baseline and pre-baseline values of SHIFTy is left to producer choice.
SHIFTyN	Shift y (N)	Num		Perm	Numeric version of SHIFTy. SHIFTyN maps one-to-one to SHIFTy within a parameter. If used for a given PARAM, should be populated (when calculable) for all post-baseline records of that PARAM regardless of whether that record is used for analysis. The decision on how to populate baseline and pre-baseline values of SHIFTyN is left to producer choice.

Users may create additional variables that are parameter-invariant functions of AVAL and BASE on the same row. Refer to Section 4.2 for the rules governing when derivations are added as rows, and when they are added as columns.

PARAM, AVAL, and AVALC

It is important to understand a key difference in approach between the SDTM Findings class variable --TEST and the ADaM BDS variable PARAM. SDTM --TEST is designed to work in conjunction with other variables called qualifiers, such as specimen type, machine type, body position, etc., in order to describe the collected result. In contrast, the ADaM BDS variable PARAM does not have any accompanying qualifier variables. PARAM is the only variable that describes AVAL or AVALC. Qualifiers are not allowed.

PARAM is created to meet an analysis need, not just because something was collected. PARAM may describe an analysis value that is highly derived from subject data from any combination of SDTM domains of any class or classes, and/or any ADaM dataset. PARAM describes what is in AVAL or AVALC.

For most parameters, only AVAL or AVALC will be populated, not both. That both --STRESC and --STRESN are present and populated in SDTM Findings class domains does not imply that both AVAL and AVALC must be present and populated in BDS datasets. AVAL and AVALC have a different purpose than --STRESN and --STRESC. For example, for parameters corresponding to numeric tests in SDTM Findings class domains, it is not recommended to copy SDTM --STRESC into AVALC, because there is no analysis need for a character value. Furthermore, doing so may result in breaking the one-to-one mapping requirement in some cases. If it is desired for traceability or listing purposes to bring the value of --STRESC into the ADaM dataset, the variable --STRESC may be copied as is without renaming it.

AVAL and AVALC are both populated only when there is a one-to-one mapping that may be useful, for example:

- When PARAM describes the numeric score of an individual question from a questionnaire, AVAL contains the score, and AVALC can be populated with the question answer text. Populating AVALC with the question answer text is supportive of review, and may help the recipient understand the meaning of the numeric score that is the subject of the parameter. Within the parameter, there is a one-to-one relationship between AVAL and AVALC on the rows on which both are populated.

- When PARAM describes a character-valued response from a set of possible values, the result is contained in AVALC. If desired for ordering or other reasons, AVAL can be also be populated, as long as the result of populating both AVAL and AVALC for the parameter is that they are a one-to-one map on the rows on which both are populated.

Table 3.3.4.2 Analysis Parameter Criteria Variables for BDS Datasets

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
CRITy	Analysis Criterion y	Char		Perm	A text string identifying a pre-specified criterion within a parameter, for example SYSBP > 90. Required if CRITyFL exists. In some cases, the presence of the text string indicates that the criterion is satisfied on this record and CRITyFL is set to Y, while a null value indicates that the criterion is not satisfied or is not evaluable and is accompanied by a null value in CRITyFL. In other cases, the text string identifies the criterion being evaluated and is populated on every row for the parameter, but whether or not the criterion is satisfied is indicated by the value of the variable CRITyFL. See CRITyFL and CRITyFN. Refer to Section 4.7 for additional discussion of CRITy, CRITyFL and CRITyFN.
CRITyFL	Criterion y Evaluation Result Flag	Char	Y *or* Y, N	Cond	Character flag variable indicating whether the criterion defined in CRITy was met by the data on the record. See CRITy for more information regarding how to use CRITy and CRITyFL to indicate whether a criterion is met. Required if CRITy exists. Refer to Section 4.7 for additional discussion.
CRITyFN	Criterion y Evaluation Result Flag (N)	Num	1 *or* 1, 0	Perm	Numeric representation of CRITyFL. There is a one-to-one mapping between CRITyFL and CRITyFN. CRITyFN can be included only if CRITyFL is also included.
MCRITy	Analysis Multi-Response Criterion y	Char		Perm	A text string identifying a pre-specified criterion within a parameter, where the criterion can have multiple responses (as opposed to CRITy which has binary responses). Required if MCRITyML exists. For example, the grade of a lab analyte is compared to the baseline grade, with the possible conditions being 0 to 1, 0 to 2, etc. The text string identifies the criterion being evaluated (for example, “Grade increase”) and is populated on every row for the parameter; which level of the criterion is satisfied is indicated by the value of the variable MCRITyML (for example “0 to 1”, “0 to 2”, etc.) See MCRITyML and MCRITyMN below, and refer to Section 4.7 for additional discussion of MCRITy, MCRITyML, and MCRITyMN.
MCRITyML	Multi-Response Criterion y Evaluation	Char		Cond	Character variable indicating which level of the criterion defined in MCRITy was met by the data on the record. See MCRITy for more information regarding how to use MCRITy and MCRITyML to indicate whether a criterion was met. Content is sponsor-defined. Required if MCRITy exists.
MCRITyMN	Multi-Response Criterion y Eval (N)	Num		Perm	Numeric representation of MCRITyML. There is a one-to-one mapping between MCRITyML and MCRITyMN. Content is sponsor-defined. MCRITyMN can be included only if MCRITyML is also included.

3.3.5 Analysis Descriptor Variables for BDS Datasets

For a given parameter within a BDS dataset, it is important to 1) be able to distinguish analysis values that are special cases of AVAL/AVALC (those for which the value is determined differently than the other analysis values within the parameter), and 2) understand what method or algorithm was used to populate each special case. The variable DTYPE (Table 3.3.5.1) is to be used to identify records within a given parameter that contain these special-case analysis values. The value of DTYPE indicates the method used for populating the analysis value; a null value of DTYPE indicates the analysis value was not a special case. The metadata for AVAL (or AVALC) will give further information about the details of any algorithm or statistical method used to derive or impute these values.

As an example, consider a situation where the analysis value for a parameter is populated by copying a value from an SDTM dataset, unless that value is missing. If the value is missing, then the analysis value is populated using a specific imputation method. It is helpful to be able to identify the “special case” instances when the analysis value is imputed rather than copied from SDTM, as well as knowing what imputation method was used.

As another example, consider a situation where the analysis value for a parameter is populated based on the subject’s corresponding value in another parameter or dataset, unless the value is outside a specified range. If the value is outside the range, then the analysis value will instead be populated with a pre-specified constant. It is helpful to be able to identify the “special case” instances where the analysis value was out of range and therefore replaced with a constant.

As yet another example, consider a situation where in addition to a subject’s analysis value for each visit, an additional timepoint is to be identified called “POST-BASELINE” with the analysis value populated with an average of the analysis values from the subject’s on-treatment visits. Though it is possible to identify the “special case” analysis values by looking at the value of AVISIT, DTYPE also facilitates the identification as well as providing information about the algorithm used.

See the CDISC Notes for DTYPE for a list of situations in which DTYPE should be populated. Examples of these instances are:

- A new record has been created within a parameter to facilitate a cross-timepoint derivation such as endpoint, minimum, maximum and average post-baseline, with the analysis value calculated according to the derivation algorithm (See Section 4.5.3);
- A new record is created within a parameter to represent a missing timepoint for a subject, with an imputed analysis value (see Section 4.5.1);
- The analysis value on an existing record is modified according to a pre-specified algorithm (e.g., setting AVAL to a pre-specified constant for results outside of a pre-specified range, replacing a missing AVAL based on a pre-specified algorithm).

In short, when the analysis value on a record within a parameter has been imputed or modified, DTYPE will indicate the method used to populate the analysis value.

DTYPE would be used if there are special cases within the new parameter that should be identified. If a parameter is wholly derived, such as a Time-to-Event parameter, then it is a misapplication to populate DTYPE for all records in that parameter because, by definition, all records are derived using the same method.

Table 3.3.5.1 Analysis Descriptor Variables for BDS Datasets

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
DTYPE	Derivation Type	Char	(DTYPE)	Cond	Analysis value derivation method. DTYPE is used to denote, and is required to be populated, when the value of AVAL or AVALC has been imputed or derived differently than the other analysis values within the parameter. DTYPE is required to be populated even if AVAL and AVALC are null on the derived record. 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 or AVALC) on an existing record is being replaced with a value based on a pre-specified algorithm. DTYPE is used to denote analysis values that are “special cases” within a parameter. For each value of DTYPE, the precise derivation algorithm must be defined in analysis variable metadata, even for DTYPE values in the controlled terminology. The controlled terminology for DTYPE is extensible. See Section 4 for examples of the use of DTYPE. Examples of DTYPE values: LOCF = last observation carried forward. WOCF = worst observation carried forward. AVERAGE = average of values.

If analysis timepoints are defined by relative day or hour windows, then the variables in Table 3.3.5.2 may be used along with ADY or ARELTM to clarify how the record representing each analysis timepoint was chosen from among the possible candidates. The record chosen is indicated by the analyzed record flag ANLzzFL (see Table 3.3.8.1). Note that the variables in Table 3.3.5.2 may not be applicable in all situations and are presented as an option.

Table 3.3.5.2 Analysis Visit Windowing Variables for BDS Datasets

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
AWRANGE	Analysis Window Valid Relative Range	Char		Perm	The range of values that are valid for a given analysis timepoint (a given value of AVISIT). For example, “5-9 DAYS”.
AWTARGET	Analysis Window Target	Num		Perm	The target or most desired analysis relative day (ADY) value or analysis relative time (ARELTM) value for a given value of AVISIT.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
AWTDIFF	Analysis Window Diff from Target	Num		Perm	Absolute difference between ADY or ARELTM and AWTARGET. It will be necessary to adjust for the fact that there is no day 0 in the event that ADY and AWTARGET are not of the same sign. If the sign of the difference is important, then AWTDIFF might have to be used in conjunction with ADY or ARELTM and possibly AWTARGET when choosing among records.
AWLO	Analysis Window Beginning Timepoint	Num		Perm	The value of the beginning timepoint (inclusive) needs to be used in conjunction to AWRANGE. For example, if AWRANGE is "5-9 DAYS", then AWLO is "5".
AWHI	Analysis Window Ending Timepoint	Num		Perm	The value of the ending timepoint (inclusive) needs to be used in conjunction to AWRANGE. For example, if AWRANGE is "5-9 DAYS", then AWHI is "9".
AWU	Analysis Window Unit	Char		Perm	Unit used for AWLO and AWHI. Examples: DAYS, HOURS.

3.3.6 Time-to-Event Variables for BDS Datasets

Table 3.3.6.1 describes variables useful for time-to-event analysis. Please refer to the document titled "The ADaM Basic Data Structure for Time-to-Event Analyses" for discussion and examples of the use of these variables and other ADaM variables to support time-to-event analyses. For example, the document describes using AVAL for the length of time from the start of the at-risk period to the event of interest, ADT for the date of event or censoring, and AVISIT for the analysis visit where event or censoring occurred.

Table 3.3.6.1 Time-to-Event Variables for BDS Datasets

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
STARTDT	Time-to-Event Origin Date for Subject	Num		Perm	The original date of risk for the time-to-event analysis. This is generally the time at which a subject is first at risk for the event of interest evaluation (as defined in the Protocol or Statistical Analysis Plan). For example, this may be the randomization date or the date of first study therapy exposure.
STARTDTM	Time-to-Event Origin Datetime	Num		Perm	The datetime associated with STARTDT in numeric format.
STARTDTF	Origin Date Imputation Flag	Char	(DATEFL)	Cond	The level of imputation of the start date. See Section 3.1.3.
STARTTMF	Origin Time Imputation Flag	Char	(TIMEFL)	Cond	The level of imputation of the start time. See Section 3.1.3.
CNSR	Censor	Num		Cond	Defines whether the event was censored for the subject within the parameter (period of observation truncated prior to event being observed). It is strongly recommended to use 0 as an event indicator and positive integers as censoring indicators. It is also recommended that unique positive integers be used to indicate coded descriptions of censoring reasons. CNSR is required for time-to-event parameters.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
EVNTDESC	Event or Censoring Description	Char		Perm	Description of the event of interest or censoring reason for the subject within the parameter.
CNSDTDSC	Censor Date Description	Char		Perm	Describes the circumstance represented by the censoring date if different from the event date that warrants censoring.

3.3.7 Toxicity and Range Variables for BDS Datasets

Table 3.3.7.1 Toxicity and Range Variables for BDS Datasets

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
ATOXGR	Analysis Toxicity Grade	Char		Perm	Toxicity grade of AVAL or AVALC for analysis; may be based on SDTM --TOXGR or an imputed or assigned value.
BTOXGR	Baseline Toxicity Grade	Char		Perm	ATOXGR of the baseline record identified by ABLFL.
ANRIND	Analysis Reference Range Indicator	Char		Perm	Indicates where AVAL or AVALC falls with respect to the normal reference range for analysis; may be based on SDTM --NRIND or an imputed or assigned value.
BNRIND	Baseline Reference Range Indicator	Char		Perm	ANRIND of the baseline record identified by ABLFL.
ANRLO	Analysis Normal Range Lower Limit	Num *		Perm	Normal range lower limit for analysis; may be based on SDTM --NRLO or an imputed or assigned value.
ANRLOC	Analysis Normal Range Lower Limit (C)	Char		Perm	Character analysis normal range lower limit. ANRLOC can be a character string mapping to ANRLO, but if so there must be a one-to-one map between ANRLO and ANRLOC within a given PARAM. ANRLOC should not be used to categorize the values of ANRLO. Within a given parameter, if there exists a row on which both ANRLOC and ANRLO are populated, then there must be a one-to-one mapping between ANRLOC and ANRLO on all rows on which both variables are populated. (In other words, there is no requirement that records with a null value in either ANRLO or ANRLOC be included when determining whether the one-to-one mapping requirement is satisfied.) On a given record, it is permissible for ANRLO, ANRLOC, or both to be null.
ANRHI	Analysis Normal Range Upper Limit	Num *		Perm	Normal range upper limit for analysis; may be based on SDTM --NRHI or an imputed or assigned value.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
ANRHIC	Analysis Normal Range Upper Limit (C)	Char		Perm	Character analysis normal range upper limit. ANRHIC can be a character string mapping to ANRHI, but if so there must be a one-to-one map between ANRHI and ANRHIC within a given PARAM. ANRHIC should not be used to categorize the values of ANRHI. Within a given parameter, if there exists a row on which both ANRHIC and ANRHI are populated, then there must be a one-to-one mapping between ANRHIC and ANRHI on all rows on which both variables are populated. (In other words, there is no requirement that records with a null value in either ANRHI or ANRHIC be included when determining whether the one-to-one mapping requirement is satisfied.) On a given record, it is permissible for ANRHI, ANRHIC, or both to be null.
AyLO	Analysis Range y Lower Limit	Num *		Cond	AyLO and/or AyHI are used for analysis ranges other than the normal range. AyLO and/or AyHI are created to capture the different levels of cutoff values used to determine whether an analysis is within a clinically acceptable value range or outside that value range. AyLO and/or AyHI are usually but not necessarily constants, parameter-specific constants, or subject-specific constants. AyLO must be included if R2AyLO is included in the dataset.
AyLOC	Analysis Range y Lower Limit (C)	Char		Perm	Character analysis range y lower limit. AyLOC can be a character string mapping to AyLO, but if so there must be a one-to-one map between AyLO and AyLOC within a given PARAM. AyLOC should not be used to categorize the values of AyLO. Within a given parameter, if there exists a row on which both AyLOC and AyLO are populated, then there must be a one-to-one mapping between AyLOC and AyLO on all rows on which both variables are populated. (In other words, there is no requirement that records with a null value in either AyLO or AyLOC be included when determining whether the one-to-one mapping requirement is satisfied.) On a given record, it is permissible for AyLO, AyLOC, or both to be null.
AyHI	Analysis Range y Upper Limit	Num *		Cond	See AyLO. For example, if ECG QTc values are summarized based on values >450, values >480, and values >500, there is a need for 3 "hi value" range variables against which to compare values: A1HI=450, A2HI=480, A3HI=500. AyHI must be included if R2AyHI is included in the dataset.
AyHIC	Analysis Range y Upper Limit (C)	Char		Perm	Character analysis range y upper limit. AyHIC can be a character string mapping to AyHI, but if so there must be a one-to-one map between AyHI and AyHIC within a given PARAM. AyHIC should not be used to categorize the values of AyHI. Within a given parameter, if there exists a row on which both AyHIC and AyHI are populated, then there must be a one-to-one mapping between AyHIC and AyHI on all rows on which both variables are populated. (In other words, there is no requirement that records with a null value in either AyHI or AyHIC be included when determining whether the one-to-one mapping requirement is satisfied.) On a given record, it is permissible for AyHI, AyHIC, or both to be null.
AyIND	Analysis Range y Indicator	Char		Perm	Indicates relationship of AVAL to the analysis range variables AyLO and/or AyHI, or the relationship of AVALC to the analysis range variables AyLOC and/or AyHIC.
ByIND	Baseline Analysis Range y Indicator	Char		Perm	AyIND of the baseline record identified by ABLFL.

** ANRLO, ANRHI, AyLO, and AyHI were mistakenly indicated as character variables in ADaMIG v1.0. The error is corrected above, and the character versions of the variables added.*

3.3.8 Indicator Variables for BDS Datasets

Refer to Section 3.1.4 for important points about the use of flag variables. See Section 3.5 for a discussion of the differences between ADaM population and baseline flags and the flags in the SDTMIG, and for a discussion of parameter-level and record-level population flags.

Table 3.3.8.1 Flag Variables for BDS Datasets

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
ABLFL	Baseline Record Flag	Char	Y	Cond	Character indicator to identify the baseline record for each subject, parameter, and baseline type (BASETYPE) combination. See BASETYPE in Table 3.3.4.1. ABLFL is required if BASE is present in the dataset. A baseline record may be derived (e.g., it may be an average), in which case DTYPE must also be populated. If BASE is populated for a parameter, and BASE is non-null for a subject for that parameter, then there must be a record flagged by ABLFL for that subject and parameter.
ABLFN	Baseline Record Flag (N)	Num	1	Perm	A numeric representation of ABLFL. ABLFN has a one-to-one mapping with ABLFL. As described in Section 3.1.1 Item 8, ABLFN can be included only if ABLFL is also included.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
ANLzzFL	Analysis Flag zz	Char	Y	Cond	ANLzzFL is a conditionally required flag to be used in addition to other selection variables when the other selection variables in combination are insufficient to identify the exact set of records used for one or more analyses. Often one ANLzzFL will serve to support the accurate selection of records for more than one analysis. Note that it is allowable to add additional descriptive text to the label (see Section 3.1.6, Item 1). When one is defining the set of records used in a particular analysis or family of analyses, ANLzzFL is supplemental to, and is intended to be used in conjunction with, other selection variables, such as subject-level, parameter-level and record-level population flags, AVISIT, DTYPE, grouping variables such as SITEGRy, and others. The lower-case letter "*zz*" in the variable name is an index for the *zz*th record selection algorithm where "*zz*" is replaced with a zero-padded two-digit integer [01-99]. Every record selection algorithm "*zz*" (i.e., every algorithm for populating an ANLzzFL) must be defined in variable metadata. When the set of records that the algorithm "*zz*" operates on is pre-filtered by application of other criteria, such as a record-level population flag, then the selection algorithm definition in the metadata must so specify. Note that the ANLzzFL value of Y indicates that the record fulfilled the requirements of the algorithm, but does not necessarily imply that the record was actually used in one or more analyses, as whether or not a record is used also depends on the other selection variables applied. The ANLzzFL flag is useful in many circumstances; an example is when there is more than one record for an analysis timepoint within a subject and parameter, as it can be used to identify the record chosen to represent the timepoint for an analysis. "zz" is an index for a record selection algorithm, such as "record closest to target relative day for the AVISIT, with ties broken by the latest record, for each AVISIT within <list of AVISITS>." Note that it is not required that a specific ANLzzFL variable has the same definition across a project or even across datasets within a study. There is also no requirement that the ANLzzFL variables in a dataset or study be used in numerical order; e.g. ANL02FL might occur in a dataset or study without ANL01FL present in the same dataset or study.
ANLzzFN	Analysis Flag zz (N)	Num	1	Perm	Numeric version of ANLzzFL. There is a one-to-one mapping between ANLzzFL and ANLzzFN within a dataset. As described in Section 3.1.1 Item 8, ANLzzFN can be included only if ANLzzFL is also included.
ONTRTFL	On Treatment Record Flag	Char	Y	Perm	Character indicator of whether the observation occurred while the subject was on treatment. ONTRTFL is producer-defined, and its definition may vary across datasets in a study based on analysis needs.
ONTRTFN	On Treatment Record Flag (N)	Num	1	Perm	Numeric representation of ONTRTFL. There is a one-to-one mapping between ONTRTFL and ONTRTFN. As described in Section 3.1.1 Item 8, ONTRTFN can be included only if ONTRTFL is also included.
LVOTFL	Last Value On Treatment Record Flag	Char	Y	Perm	Character indicator of the subject's last non-missing value on treatment for each parameter.

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
LVOTFN	Last Value On Treatment Record Flag (N)	Num	1	Perm	Numeric representation of LVOTFL. There is a one-to-one mapping between LVOTFL and LVOTFN. As described in Section 3.1.1 Item 8, LVOTFN can be included only if LVOTFL is also included.

Table 3.3.8.2 BDS Population Indicators

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
ITTRFL	Intent-To-Treat Record-Level Flag	Char	Y	Perm	These indicators identify whether or not the subject was in the specified analysis for the specific record. Useful when the subject is included in the subject-level population, but there are records for the subject that do not satisfy requirements for the population. The valid values of these record-level population indicators are Y or Null. If a flag is used, the corresponding numeric version (*FN, where 1=yes) of the flag can also be included. As described in Section 3.1.1 Item 8, the *FN version of the variable can be included only if the corresponding *FL is also included. Additional indicators may also be used; refer to Section 3.1.4.
SAFRFL	Safety Analysis Record-Level Flag	Char	Y	Perm	
FASRFL	Full Analysis Set Record-Level Flag	Char	Y	Perm	
PPROTRFL	Per-Protocol Record-Level Flag	Char	Y	Perm	
COMPLRFL	Completers Record-Level Flag	Char	Y	Perm	
ITTPFL	Intent-To-Treat Parameter-Level Flag	Char	Y	Perm	These indicators identify whether or not the subject was in the specified analysis for the specific parameter. Useful when the subject is included in the subject-level population, but there are parameters for which the subject does not satisfy requirements for the population. The valid values of these parameter-level population indicators are Y or Null. If a flag is used, the corresponding numeric version (*FN, where 1=yes) of the flag can also be included. As described in Section 3.1.1 Item 8, the *FN version of the variable can be included only if the corresponding *FL is also included. Additional indicators may also be used; refer to Section 3.1.4, “Flag Variable Conventions.”
SAFPFL	Safety Analysis Parameter-Level Flag	Char	Y	Perm	
FASPFL	Full Analysis Set Parameter-Level Flag	Char	Y	Perm	
PPROTPFL	Per-Protocol Parameter-Level Flag	Char	Y	Perm	
COMPLPFL	Completers Parameter-Level Flag	Char	Y	Perm	

3.3.9 Datapoint Traceability Variables

Variables to support datapoint traceability should be included whenever practical. The dataset content that serves as primary candidates for datapoint traceability when used in conjunction with USUBJID are the dataset or domain name, the name of the source variable, and the relevant sequence number (SDTM domain

--SEQ value or the ADaM ASEQ value). The ADaM ASEQ variable (Table 3.3.1.1) facilitates datapoint traceability by providing sequence numbers that are unique within a subject within an ADaM dataset, ensuring uniqueness of a record when used in combination with USUBJID.

Table 3.3.9.1 defines additional variables useful in certain situations to facilitate datapoint traceability. They are useful in situations where a single ADaM dataset or multiple SDTM datasets and/or ADaM datasets were used to create one ADaM dataset. Section 4.4 contains an example of how to use these variables.

If the value of AVAL or AVALC in the ADaM dataset is taken from another ADaM dataset, SRCDOM, SRCVAR, and SRCSEQ will contain the name of the source ADaM dataset, the variable name and the ASEQ value of the row where the source datapoint is located, respectively.

If all values of AVAL or AVALC in the ADaM dataset are taken from a single SDTM domain, the SRC variables defined here can be used. However, all records in the ADaM dataset would have the same value for SRCDOM and SRCVAR. In this case, the producer may elect to simply include the --SEQ variable from the source SDTM domain since it would be sufficient to provide the needed traceability within a subject. In the event that the source SDTM dataset is a supplemental qualifier, the two-letter domain prefix of --SEQ in the ADaM dataset would be the related domain abbreviation (the value of RDOMAIN in SUPP-- or SUPPQUAL).

Variables used for datapoint traceability may also include any other variables that facilitate transparency and clarity of derivations and analysis.

Table 3.3.9.1 Datapoint Traceability Variables

Variable Name	Variable Label	Type	Codelist/ Controlled Terms	Core	CDISC Notes
SRCDOM	Source Data	Char		Perm	The SDTM domain name or ADaM dataset name that relates to the analysis value (e.g., AVAL or AVALC in a BDS dataset). If the source data is a supplemental qualifier in SDTM, this variable will contain the value of RDOMAIN in SUPP-- or SUPPQUAL.
SRCVAR	Source Variable	Char		Perm	The name of the column (in the domain or dataset identified by SRCDOM) that relates to the analysis value (e.g., AVAL or AVALC in a BDS dataset). In the event that SRCDOM is a SUPPQUAL, then SRCVAR will be populated with the value of the related QNAM.
SRCSEQ	Source Sequence Number	Num		Perm	The sequence number --SEQ or ASEQ of the row (in the domain or dataset identified by SRCDOM) that relates to the analysis value (e.g., AVAL or AVALC in a BDS dataset). In the event that SRCDOM is a SUPPQUAL, then this variable will contain the sequence number of the relevant related domain record.

3.4 Analysis-Enabling Variables

There is a class of variables that enable one or more of the analyses that the dataset was designed to support. See the definition of analysis-enabling in Section 1.5.1. Often, these enabling variables would include the indicator variables and analysis descriptor variables described above, which are often needed to make the ADaM dataset one statistical procedure away from analysis results. Enabling variables may also include stratification and subgrouping variables, model covariates and any other variables required to be present in order to perform an analysis.

3.5 Differences between SDTM and ADaM Population and Baseline Flags

The SDTM Implementation Guide includes controlled terminology for some Supplemental Qualifier values for subject-level population flags. The conceptual mapping from those terms to ADaM indicator variables is presented in Table 3.5.1.

Table 3.5.1 ADaM Subject-Level Population Flags Corresponding to SDTM Supplemental Qualifiers

SDTM QNAM	SDTM QLABEL	ADaM Subject-Level Population Flags
COMPLT	Completers Population Flag	COMPLFL
FULLSET	Full Analysis Set Flag	FASFL
ITT	Intent to Treat Population Flag	ITTFL
PPROT	Per Protocol Set Flag	PPROTFL
SAFETY	Safety Population Flag	SAFFL

It is possible that the ADaM subject-level population flags might not match their conceptual counterparts in SDTM. For example, the SDTM ITT Supplemental Qualifier may not match the ADaM ITTFL indicator variable for a given subject. These population indicators may not match because of operational issues. It is entirely possible that a company could inherit a SDTM database that for various reasons cannot be changed. It is not incumbent on those creating ADaM datasets to go back and “fix” the SDTM population supplemental qualifiers and there may be good reason not to do so. The ADaM Team agrees that it would be best if the SDTM subject-level population Supplemental Qualifiers are in harmony with the ADaM population indicator variables, but it is important to recognize that there may be situations where they differ. There are additional ADaM subject-level population flags that do not have counterparts in SDTM. ADaM also supports parameter-level and record-level population flags, which do not exist in SDTM.

Similarly, a baseline record identified in SDTM may not be the record identified in an ADaM dataset and there are many reasons why this may occur. There are ADaM parameters that are highly derived and do not have simple counterparts in a Findings domain. An ADaM parameter may be derived from SDTM data spanning multiple domains and classes. Such a parameter would not exist in the SDTM and so its baseline could only exist in the ADaM dataset. Also, it may be necessary to have separate baselines for different periods within the study, for example to support analyses of change from screening baseline, double-blind treatment baseline, and open-label extension baseline (see Section 4.2, Rule 6). When there is record-level population flagging, it may be necessary to have different baselines for two different analysis populations. Lastly, it may be desired to conduct analyses for different definitions of baseline. The ADaM baseline flag ABLFL, coupled with the BASE and BASETYPE columns, plus population flags, can handle all of these practical scenarios.

For analysis purposes, the values of population and baseline flags used for analysis are found in the ADaM datasets. ADaM flags should be described in ADaM metadata.

4 Implementation Issues, Standard Solutions, and Examples

The ADaM standard variables (columns) are described in Section 3. However, there is more to the ADaM model than just using the ADaM standard variables. The purpose of Section 4 is to provide additional guidance on how to implement ADaM standard datasets correctly.

Section 4.1 provides examples of treatment variables for common trial designs.

Sections 4.2-4.9 are concerned with the BDS. These sections provide standard solutions to BDS implementation issues, illustrated with examples. The focus of Sections 4.2-4.7 is the building of a BDS dataset. Section 4.2 focuses on assembling the rows and columns of the dataset. Sections 4.3 and 4.4 discuss issues around the inclusion/exclusion of rows not used in an analysis. Sections 4.5, 4.6, and 4.7 discuss issues around identification of rows for analysis. Section 4.8 contains an example of the use of the BDS variables for phase, period, and subperiod. Section 4.9 presents some comments on additional issues to consider in building ADaM datasets.

For examples of the OCCDS, refer to the separate document "ADaM Structure for Occurrence Data".

For space reasons, the examples do not show complete datasets with all of the required and permissible variables. Rather, only those variables needed to illustrate the point being discussed are shown.

4.1 Examples of Treatment Variables for Common Trial Designs

Examples 1-4 in this section illustrate the concepts related to treatment variables in ADSL for several different trial designs, including a parallel design, a crossover design, and an open-label extension of a parallel design study. Note that only selected variables are illustrated; these examples are not intended to imply that these are the only variables in ADSL. Examples 5 and 6 illustrate concepts related to treatment variables in BDS.

Example 1

In the first example (Table 4.1.1), the treatment variables for three subjects in a parallel design study (one treatment period) are illustrated. Note that the third subject was randomized to active treatment yet received placebo instead. TR01SDT and TR01EDT are not required variables in trial designs that do not involve multiple treatment periods.

Table 4.1.1 Randomized Parallel Design

Row	USUBJID	ARM	ACTARM	TRT01P	TRT01A	TRTSDT	TRTEDT
1	1001	Drug X 5 mg	Drug X 5 mg	Drug X 5 mg	Drug X 5 mg	23OCT2007	17DEC2007
2	1002	Placebo	Placebo	Placebo	Placebo	19JUL2006	20SEP2007
3	1003	Drug X 5 mg	Placebo	Drug X 5 mg	Placebo	01NOV2007	20NOV2007

Example 2

The second example (Table 4.1.2) illustrates the treatment variables for three subjects in a two-period crossover design. It should be noted that TRTSDT and TRTEDT are not displayed, but TRTSDT=TR01SDT and TRTEDT is the maximum of TR01EDT and TR02EDT as some subjects may have discontinued before receiving TRT02P. Note that subjects 1002 and 1003 (in rows 2 and 3) were each exposed to placebo for both trial periods.

Table 4.1.2 Two-Period Crossover Design

Row	USUBJID	TRTSEQP	TRT01P	TRT02P	TRTSEQA	TRT01A	TRT02A	TR01SDT	TR01EDT	TR02SDT	TR02EDT
1	1001	Placebo – Drug X	Placebo	Drug X	Placebo – Drug X	Placebo	Drug X	15FEB2006	03MAY2006	10MAY2006	15AUG2006
2	1002	Placebo – Drug X	Placebo	Drug X	Placebo – Placebo	Placebo	Placebo	01MAR2006	12JUN2006	20JUN2006	23SEP2006
3	1003	Drug X – Placebo	Drug X	Placebo	Placebo – Placebo	Placebo	Placebo	03FEB2006	25APR2006	01MAY2006	04AUG2006

Example 3

The third example (Table 4.1.3) illustrates the treatment variables for three subjects in a three-period crossover design. It should be noted that TRTSDT and TRTEDT are not displayed, but TRTSDT=TR01SDT and TRTEDT is the maximum of TR01EDT, TR02EDT, and TR03EDT as some subjects may have discontinued before receiving TRT03P. In this trial, all subjects received the planned treatment at each period so the TRTxxA variables are not needed.

Table 4.1.3 Three-Period Crossover Design

Row	USUBJID	TRTSEQP	TRT01P	TRT02P	TRT03P	TR01SDT	TR01EDT	TR02SDT	TR02EDT	TR03SDT	TR03EDT
1	1001	Placebo – Drug X – Drug Y	Placebo	Drug X	Drug Y	15FEB2006	03MAY2006	10MAY2006	15AUG2006	23AUG2006	14NOV2006
2	1002	Drug Y – Placebo – Drug X	Drug Y	Placebo	Drug X	01MAR2006	12JUN2006	20JUN2006	23SEP2006	01OCT2006	05DEC2006
3	1003	Drug X – Drug Y – Placebo	Drug X	Drug Y	Placebo	03FEB2006	25APR2006	01MAY2006	04AUG2006	12AUG2006	15OCT2006

Example 4

The fourth example (Table 4.1.4) illustrates the treatment variables for two subjects in an open-label extension from a parallel design study. For open-label studies, the variable TRT01P is used for the treatment to which the subject was randomized in the double-blinded trial. TRT02P is used for the open-label treatment.

Table 4.1.4 Open-Label Extension of a Parallel Design – ADSL Dataset

Row	USUBJID	TRTSEQP	TRT01P	TRT02P	TR01SDT	TR01EDT	TR02SDT	TR02EDT
1	1001	Drug X 5 mg - Drug X 5 mg	Drug X 5 mg	Drug X 5 mg	14AUG2007	20SEP2007	21SEP2007	15MAR2008
2	1002	Placebo - Drug X 5 mg	Placebo	Drug X 5 mg	05JUL2007	15AUG2007	17AUG2007	04FEB2008

Examples 5 and 6 build on the ADSL dataset illustrated in Table 4.1.4.

Example 5

As stated in Section 3.3.2, at least one treatment variable is required in a BDS dataset. This requirement is satisfied by any of the subject-level or record-level treatment variables, e.g. TRTxxP, TRTP. The following two examples illustrate some possible approaches for BDS treatment variables. These examples are not meant to imply a standard or best practice; they are for illustration purposes only. Please refer to Section 3.3.2 for important additional information.

In the first illustration of a BDS dataset (Table 4.1.5), the ADSL treatment variables have been copied into the BDS dataset. In addition, TRTP contains the treatment assigned at the time of the assessment (i.e., at ADT). This allows this dataset to support multiple analysis strategies. If the data are analyzed using the randomized treatment from the double-blinded trial, then TRT01P can be used as the treatment variable in the analysis. If the data are analyzed using the treatment assigned at the time of the assessment, then TRTP can be used as the treatment variable in the analysis. In this example, TRTP is blank for assessments that are not on-treatment.

Table 4.1.5 Open-Label Extension of a Parallel Design – BDS Dataset, Illustration 1

Row	USUBJID	APERIOD	ADT	TRTP	TRT01P	TRT02P
1	1001		10AUG2007		Drug X 5 mg	Drug X 5 mg
2	1001	1	14AUG2007	Drug X 5 mg	Drug X 5 mg	Drug X 5 mg
3	1001	2	21SEP2007	Drug X 5 mg	Drug X 5 mg	Drug X 5 mg
4	1002		01JUL2007		Placebo	Drug X 5 mg
5	1002	1	05JUL2007	Placebo	Placebo	Drug X 5 mg
6	1002	2	17AUG2007	Drug X 5 mg	Placebo	Drug X 5 mg

Example 6

A different approach is illustrated in the second illustration of a BDS dataset (Table 4.1.6), TRTP contains the treatment being used for the analysis of that record. Note that the assessment occurring prior to period 1 have TRTP populated, even though no treatment is actually administered on those dates, as seen by ADSL in Table 4.1.4.

Table 4.1.6 Open-Label Extension of a Parallel Design – BDS Dataset, Illustration 2

Row	USUBJID	APERIOD	ADT	TRTP
1	1001		10AUG2007	Drug X 5 mg
2	1001	1	14AUG2007	Drug X 5 mg
3	1001	2	21SEP2007	Drug X 5 mg
4	1002		01JUL2007	Placebo
5	1002	1	05JUL2007	Placebo
6	1002	2	17AUG2007	Drug X 5 mg

4.2 Creation of Derived Columns versus Creation of Derived Rows

This section provides specific rules to use in building a BDS dataset. These rules are essential, because they ensure the BDS dataset is analysis-focused, with all analysis-enabling variables and supportive variables included in a predictable structure, while preventing a “horizontalization” of the dataset.

The rows (i.e., records) in the ADaM BDS represent subject data for analysis parameters and timepoints (as applicable). There may be multiple rows within a given combination of subject, parameter and timepoint, depending on the number of observations collected or derived, baseline definition, etc.

The ADaM BDS structure contains a central set of columns (i.e., variables) that represent the data being analyzed. These variables include the value being analyzed (e.g., AVAL) and the description of the value being analyzed (e.g., PARAM). Other columns in the dataset provide more information about the value being analyzed (e.g., the subject identification) or describe and trace the derivation of it (e.g., DTYPE) or support the analysis of it (e.g., treatment variables, covariates). Standard columns exist for a variety of purposes, such as SDTM record identifiers for traceability, population and other record selection flags, analysis values, and some standard functions of analysis values. Permissible columns are not limited to those whose variable names are specified in Section 3, and may include study-specific analysis model covariates, subgrouping variables, variables supportive of traceability, as well as other variables needed for analysis or useful for review.

The BDS is flexible in that derived data can be added to the collected data as additional rows and columns that support the analyses and provide traceability. However, there are some constraints on how to incorporate derived data in the BDS dataset. Specifically, the subject of Section 4.2 is to address when derived data that are functions of analysis values should be added as additional columns, and when they should be added as additional rows instead.

The precise sequence of steps involved in creating a BDS ADaM dataset varies according to operational and study-specific needs. For the purposes of this discussion, it is useful to think of two fundamental steps.

1. Create an initial dataset from the source datasets: The first step is to create a set of rows and columns more or less directly derived from or loaded from input datasets (primarily SDTM datasets and other ADaM datasets) into their appropriate places. This step will include creation and population of columns containing analysis parameters (PARAM etc.), analysis timepoint (AVISIT etc.) and analysis values (AVAL, AVALC, etc.). It would also include addition of columns containing identifiers (STUDYID, SITEID, USUBJID, SUBJID) and other SDTM variables for traceability (VISIT, --SEQ, etc.).
2. Add additional derived data as needed for the analysis: The second step consists of adding derived rows and columns based on the initial set of ADaM dataset records and columns. The rules that govern this step are:

   Rule 1: A parameter-invariant function of AVAL and BASE on the same row that does not involve a transform of BASE should be added as a new column.

   Rule 2: A transformation of AVAL that does not meet the conditions of Rule 1 should be added as a new parameter, and AVAL should contain the transformed value.

   Rule 3: A function of one or more rows within the same parameter for the purpose of creating an analysis timepoint should be added as a new row for the same parameter.

   Rule 4: A function of multiple rows within a parameter should be added as a new parameter.

   Rule 5: A function of more than one parameter should be added as a new parameter.

   Rule 6: When there is more than one definition of baseline, each additional definition of baseline requires the creation of its own set of rows.

These rules are further described and illustrated in the remainder of this section.

It is important to understand that the rules outlined here are specific to rows and columns that are created based on data already present in the ADaM dataset. The rules do not apply to data that are copied or derived directly from other datasets (either SDTM or ADaM or both). For example, how to include a transformation of AVAL within the same dataset is governed by the rules, but the inclusion of a covariate derived from another dataset (e.g., inclusion of a variable from ADSL) is not governed by these rules.

4.2.1 Rules for the Creation of Rows and Columns

To preserve the BDS, it is necessary to place constraints on when one is allowed to create derived columns. Rule 1 describes when derived data belongs in columns. Rules 2-6 describe situations in which one should derive data in new rows, whether as entire new parameters, or as additional rows in existing parameters. In the sections and examples below, there is some text that is **bolded**. The use of the bold font is to emphasize to the reader the importance of the concept or example that is being discussed.

Rule 1. A parameter-invariant function of AVAL and BASE on the same row that does not involve a transform of BASE should be added as a new column.

The three conditions of Rule 1 for when a function of AVAL and BASE should be added as a column (i.e., a function column) are:

1. The function is of AVAL and, optionally, BASE, on the same row; and
2. The function is parameter-invariant; and
3. The function does not involve a transform of BASE.

The remainder of the discussion of this rule is devoted to explaining these conditions.

PARAM uniquely describes the contents of AVAL or AVALC. Often, AVAL itself is not the value that is needed for analysis. For example, in a change from baseline analysis, it is the change from baseline CHG that is analyzed. The change from baseline column CHG should be created according to Rule 1 because it satisfies the three conditions:

1. CHG is derived from AVAL and BASE on the same row;
2. The same calculation applies on all rows in the dataset on which CHG is populated (the function CHG=AVAL-BASE does not vary according to PARAM). This second condition is known as the property of "*parameter-invariance*"; **unless listed in Section 3**, a function of AVAL (and optionally BASE) may not be derived as a column if' it is parameter-variant (i.e., is calculated differently for different parameters) .
3. In the function CHG=AVAL-BASE, BASE is not transformed.

Table 4.2.1.1 illustrates the CHG column. Note that the producer elected not to populate CHG on the screening or run-in rows, as they are pre-baseline. The baseline flag column ABLFL identifies the row that was used to populate the BASE column.

Table 4.2.1.1 Illustration of Rule 1: Creation of a Column Containing a Same-Row Parameter-Invariant Function of AVAL and BASE

Row	PARAM	PARAMCD	AVISIT	ABLFL	AVAL	BASE	CHG
1	Weight (kg)	WEIGHT	Screening		99	100	**.**
2	Weight (kg)	WEIGHT	Run-In		101	100	**.**
3	Weight (kg)	WEIGHT	Baseline	Y	100	100	**0**
4	Weight (kg)	WEIGHT	Week 24		94	100	**-6**
5	Weight (kg)	WEIGHT	Week 48		92	100	**-8**
6	Weight (kg)	WEIGHT	Week 52		95	100	**-5**
7	Pulse Rate (bpm)	PULSE	Screening		63	62	**.**
8	Pulse Rate (bpm)	PULSE	Run-In		67	62	**.**
9	Pulse Rate (bpm)	PULSE	Baseline	Y	62	62	**0**
10	Pulse Rate (bpm)	PULSE	Week 24		66	62	**4**
11	Pulse Rate (bpm)	PULSE	Week 48		70	62	**8**
12	Pulse Rate (bpm)	PULSE	Week 52		64	62	**2**

Now consider the potential function column LOG10 = Log10(AVAL). This function satisfies all three conditions of Rule 1 and as such is allowed as a function column. However, LOG10BAS = Log10(BASE) and LOG10CHG = Log10(AVAL) – Log10(BASE) are not allowable columns as they involve a transform of BASE.

Therefore, if it is desired to perform change from baseline analysis in LOG10, columns for LOG10, baseline of LOG10 and change from baseline of LOG10 would be needed for analysis and review, then the Log10 transformation should instead be created as a new parameter, so that the usual columns AVAL, BASE and CHG can be used. This is because columns for baseline of LOG10 and change from baseline of LOG10 would not satisfy the conditions of Rule 1. Baseline of LOG10 violates the first condition, because it is not generally a function of AVAL on the same row (does not generally vary by AVAL), and instead is a function only of AVAL on the baseline row. “Change from baseline of LOG10” =LOG10(AVAL) - LOG10(BASE) violates the third condition, because it contains the Log10 transform of BASE.

The intent is to use the standard columns as much as possible, to keep the structure as standard as possible, and avoid undue “horizontalization,” while still permitting efficient use of function columns.

Any function that satisfies the three conditions of Rule 1 is allowed as a column. If the function is listed in Section 3, then the ADaM standard column name must be used just as CHG is used in Table 4.2.1.1.

Rule 2. A transformation of AVAL that does not meet the conditions of Rule 1 should be added as a new parameter, and AVAL should contain the transformed value.

If the intention is to redefine AVAL, BASE, CHG, etc. in terms of a transform of AVAL, then a new parameter must be added, in which PARAM describes the transform. **The creation of a new parameter results by definition in the creation of a new set of rows.**

For example, as described in the discussion of Rule 1, in a change from baseline analysis of the logarithm of weight, AVAL should contain the log of weight, BASE should contain the baseline value of the log of weight, and CHG should contain the difference between the two. PARAM should contain a description of the transformed data contained in AVAL, e.g., “Log10 (Weight (kg))”. In this way the ADaM standard accommodates an analysis of transformed data in the standard columns without creating a multiplicity of new special-purpose columns.

In Table 4.2.1.2Table we see that the producer has chosen values of AVISITN that correspond to week number and which serve well for sorting and for plotting. VISITNUM is the SDTM visit number.

Note that when SDTM variables, such as USUBJID, SUBJID, SITEID, VISIT, VISITNUM and --SEQ, are included in an ADaM dataset with their original SDTM variable names, their values must not be altered in any way.

Table 4.2.1.2 Illustration of Rule 2: Creation of a New Parameter to Handle a Transformation

Row	PARAM	PARAMCD	VISIT	AVISIT	AVISITN	VISITNUM	ABLFL	AVAL	BASE	CHG
1	Weight (kg)	WEIGHT	Visit -1	Screening	-4	1		99	100	.
2	Weight (kg)	WEIGHT	Visit 0	Run-In	-2	2		101	100	.
3	Weight (kg)	WEIGHT	Visit 1	Baseline	0	3	Y	100	100	0
4	Weight (kg)	WEIGHT	Visit 12	Week 24	24	4		94	100	-6
5	Weight (kg)	WEIGHT	Visit 24	Week 48	48	5		92	100	-8
6	Weight (kg)	WEIGHT	Visit 26	Week 52	52	6		95	100	-5
7	Log10(Weight (kg))	L10WT	Visit -1	Screening	-4	1		1.9956	2	.
8	Log10(Weight (kg))	L10WT	Visit 0	Run-In	-2	2		2.0043	2	.
9	Log10(Weight (kg))	L10WT	Visit 1	Baseline	0	3	Y	2	2	0
10	Log10(Weight (kg))	L10WT	Visit 12	Week 24	24	4		1.9731	2	-0.0269

Row	PARAM	PARAMCD	VISIT	AVISIT	AVISITN	VISITNUM	ABLFL	AVAL	BASE	CHG
11	Log10(Weight (kg))	L10WT	Visit 24	Week 48	48	5		1.9638	2	-0.0362
12	Log10(Weight (kg))	L10WT	Visit 26	Week 52	52	6		1.9777	2	-0.0223

A related application of Rule 2 is in the case where it is necessary to support analysis and reporting in two different systems of units. In SDTM Findings domains such as LB, QS, EG, etc., the --STRESN column is the only numeric result column, and is also the only standardized numeric result column. The --ORRES column contains a character representation of the collected result, in the collected units specified in the --ORRESU column. The --ORRES column is not standardized. So for example, if data are typically collected in conventional units, SDTM cannot accommodate standardized data in both conventional units and the International System of Units (SI). In SDTM, for any given --TEST, a producer can standardize in one system of units but not two. If one wishes to be able to analyze standardized results in both conventional units and in SI units, a transform in an ADaM dataset is needed. In each such case, a new parameter must be created in order to accommodate standardized data in the other system of units.

The description in the PARAM column must contain the units, as well as any other information such as location and specimen type that is needed to ensure that PARAM uniquely describes what is in AVAL, and differentiates between parameters as needed. PARAM cannot be the same for different units.

Table 4.2.1.3 shows an example of data supporting analyses of low-density lipoprotein (LDL) cholesterol in both conventional units (mg/dL) and SI units (mmol/L). In this study, SDTM cholesterol data were standardized in mg/dL. In the ADaM dataset, two records, one for each system of units, were generated from each original SDTM record. As described in Section 4.9.5, as a general rule, when a record is derived from a single record in the dataset, retain on the derived record any variable values from the original record that do not change and that make sense in the context of the new record (e.g., --SEQ, VISIT, VISITNUM, --TPT, covariates, etc.).

Table 4.2.1.3 Illustration of Rule 2: Creation of a New Parameter to Handle a Second System of Units

Row	PARAM	PARAMCD	AVISIT	AVISITN	VISITNUM	LBSEQ	ABLFL	AVAL	BASE	CHG	PCHG
1	LDL Cholesterol (mg/dL)	LDL	Screening	-2	1	2829		206.3	213.4		
2	LDL Cholesterol (mg/dL)	LDL	Run-In	-1	2	2830		202.1	213.4		
3	LDL Cholesterol (mg/dL)	LDL	Week 0	0	3	2831	Y	213.4	213.4	0.0	0.00
4	LDL Cholesterol (mg/dL)	LDL	Week 5	5	4	2832		107.4	213.4	-106.0	-49.67
5	LDL Cholesterol (mg/dL)	LDL	Week 11	11	5	2833		90.2	213.4	-123.2	-57.73
6	LDL Cholesterol (mg/dL)	LDL	Week 17	17	6	2834		96.8	213.4	-116.6	-54.64
7	LDL Cholesterol (mg/dL)	LDL	Week 23	23	7	2835		104.0	213.4	-109.4	-51.27
8	**LDL Cholesterol (mmol/L)**	**LDLT**	**Screening**	**-2**	**1**	**2829**		**5.3349**	**5.5185**		
9	**LDL Cholesterol (mmol/L)**	**LDLT**	**Run-In**	**-1**	**2**	**2830**		**5.2263**	**5.5185**		
10	**LDL Cholesterol (mmol/L)**	**LDLT**	**Week 0**	**0**	**3**	**2831**	**Y**	**5.5185**	**5.5185**	**0.0000**	**0.00**
11	**LDL Cholesterol (mmol/L)**	**LDLT**	**Week 5**	**5**	**4**	**2832**		**2.7773**	**5.5185**	**-2.7412**	**-49.67**
12	**LDL Cholesterol (mmol/L)**	**LDLT**	**Week 11**	**11**	**5**	**2833**		**2.3326**	**5.5185**	**-3.1859**	**-57.73**
13	**LDL Cholesterol (mmol/L)**	**LDLT**	**Week 17**	**17**	**6**	**2834**		**2.5032**	**5.5185**	**-3.0153**	**-54.64**
14	**LDL Cholesterol (mmol/L)**	**LDLT**	**Week 23**	**23**	**7**	**2835**		**2.6894**	**5.5185**	**-2.8291**	**-51.27**

Rule 3. A function of one or more rows within the same parameter for the purpose of creating an analysis timepoint should be added as a new row for the same parameter.

For analysis purposes, there is often a need to impute missing data, or to create a derived conceptual timepoint. Such derivations should result in the creation of new derived records within the same parameter.

As described in Section 4.9.5, as a general rule, when a record is derived from a single record in the dataset, retain on the derived record any variable values from the original record that do not change and that make sense in the context of the new record (e.g., --SEQ, VISIT, VISITNUM, --TPT, covariates, etc.). When a record is derived from multiple records, then retain on the derived record all variable values that are constant across the original records, that do not change, and that make sense in the context of the new record. Note that there are situations when retention of values from an original record or records would make no sense on the derived record; in such cases, do not retain those values.

For example, suppose that the analysis endpoint value is defined as the average of the last two available post-baseline values. In this case, a new row should be added, with a corresponding description in AVISIT, and the DTYPE (derivation type) column should contain a description on that row such as “AVERAGE” to indicate both that the row was derived, and also the derivation method. The metadata associated with AVISIT=Endpoint should adequately describe which records are used in the definition of the average. Note that even though the set of records for the log transformation of weight are derived, DTYPE is not populated for every row. DTYPE should be used to indicate rows that are derived within a given value of PARAM and is not to be used as an indication of whether the record exists in SDTM.

In Table 4.2.1.4, VISITNUM is not retained on the derived record because VISITNUM is not constant on the precursor records, and also makes no sense in the derived analysis timepoint, which is an average that in most cases will span multiple VISITs. Similarly VSSEQ is not constant across multiple original records, so VSSEQ is not populated on the derived record. PARAM and BASE should be retained because they are constant on the precursor records and make sense in the context of the new record. For the new record, AVAL and change are recalculated, and AVISIT, AVISITN, and DTYPE are populated appropriately. Note that the metadata will specify the algorithm used for the calculation (in this example, the rows being averaged).

AVISIT and AVISITN are defined by the producer. AVISIT and AVISITN are not necessarily defined the same for the individual parameters within a dataset. The definition and derivation of the values of AVISIT, and any dependence on parameter, should be described in metadata. In this example, the producer decided to set AVISITN to 9999 on the derived AVISIT=Endpoint records.

Table 4.2.1.4 Illustration of Rule 3: Creation of a New Row to Handle a Derived Analysis Timepoint

Row	PARAM	AVISIT	AVISITN	VISITNUM	VSSEQ	ABLFL	AVAL	BASE	CHG	DTYPE
1	Weight (kg)	Screening	-4	1	1164		99	100	.	
2	Weight (kg)	Run-In	-2	2	1165		101	100	.	
3	Weight (kg)	Baseline	0	3	1166	Y	100	100	0	
4	Weight (kg)	Week 24	24	4	1167		94	100	-6	
5	Weight (kg)	Week 48	48	5	1168		92	100	-8	
6	Weight (kg)	Week 52	52	6	1169		95	100	-5	
7	**Weight (kg)**	**Endpoint**	**9999**				**93.5**	**100**	**-6.5**	**AVERAGE**
8	Log10(Weight (kg))	Screening	-4	1	1164		1.9956	2	.	
9	Log10(Weight (kg))	Run-In	-2	2	1165		2.0043	2	.	
10	Log10(Weight (kg))	Baseline	0	3	1166	Y	2	2	0	

Row	PARAM	AVISIT	AVISITN	VISITNUM	VSSEQ	ABLFL	AVAL	BASE	CHG	DTYPE
11	Log10(Weight (kg))	Week 24	24	4	1167		1.9731	2	-0.0269	
12	Log10(Weight (kg))	Week 48	48	5	1168		1.9638	2	-0.0362	
13	Log10(Weight (kg))	Week 52	52	6	1169		1.9777	2	-0.0223	
14	**Log10(Weight (kg))**	**Endpoint**	**9999**				**1.9708**	**2**	**-0.0292**	**AVERAGE**

An extension of Rule 3 is necessary in the case where there is value-level (record-level) population flagging. For example, assume the Statistical Analysis Plan states that if the subject is off drug for seven days prior to a visit, the measurement collected at that visit is not included in the per-protocol analysis. Then for some subjects, the last two available values may be different for Intent-to-Treat and for Per-Protocol analyses, so that the calculated endpoint averages would be different. For such subjects, two distinct derived endpoint rows would be needed, the appropriate row for each analysis indicated by the record-level population flags ITTRFL and PPROTRFL.

In Table 4.2.1.5, the analyzed endpoint value varies according to the population. For example, for PARAM=Weight (kg), the last two available ITT values are 92 and 95, whose average is 93.5; whereas the last two Per-Protocol values are 94 and 92, whose average is 93. That is why two derived Endpoint rows are required for this subject. For other subjects, the ITT and Per-Protocol data that are input to the Endpoint average may be the same; in that case, only one Endpoint record would be needed, on which ITTRFL and PPROTRFL would both be set to Y. Values of AVISIT and AVISITN are producer-controlled. As in the example in Table 4.2.1.4, the producer decided to set AVISITN to 9999 on the derived AVISIT=Endpoint records. Note that the metadata will specify the algorithm used for the calculation (in this example, the rows being averaged).

Table 4.2.1.5 Illustration of Rule 3: Creation of New Rows to Handle a Derived Analysis Timepoint When There is Value-Level Population Flagging

Row	PARAM	AVISIT	AVISITN	VISITNUM	VSSEQ	ABLFL	AVAL	BASE	CHG	DTYPE	ITTRFL	PPROTRFL
1	Weight (kg)	Screening	-4	1	1164		99	100	.		Y	Y
2	Weight (kg)	Run-In	-2	2	1165		101	100	.		Y	Y
3	Weight (kg)	Baseline	0	3	1166	Y	100	100	0		Y	Y
4	Weight (kg)	Week 24	24	4	1167		94	100	-6		Y	Y
5	Weight (kg)	Week 48	48	5	1168		92	100	-8		Y	Y
6	Weight (kg)	Week 52	52	6	1169		95	100	-5		Y	
7	**Weight (kg)**	**Endpoint**	**9999**				**93.5**	**100**	**-6.5**	**AVERAGE**	**Y**	
8	**Weight (kg)**	**Endpoint**	**9999**				**93**	**100**	**-7**	**AVERAGE**		**Y**
9	Log10 (Weight (kg))	Screening	-4	1	1164		1.9956	2	.		Y	Y
10	Log10 (Weight (kg))	Run-In	-2	2	1165		2.0043	2	.		Y	Y
11	Log10 (Weight (kg))	Baseline	0	3	1166	Y	2	2	0		Y	Y
12	Log10 (Weight (kg))	Week 24	24	4	1167		1.9731	2	-0.0269		Y	Y
13	Log10 (Weight (kg))	Week 48	48	5	1168		1.9638	2	-0.0362		Y	Y
14	Log10 (Weight (kg))	Week 52	52	6	1169		1.9777	2	-0.0223		Y	
15	**Log10 (Weight (kg))**	**Endpoint**	**9999**				**1.9708**	**2**	**-0.0292**	**AVERAGE**	**Y**	
16	**Log10 (Weight (kg))**	**Endpoint**	**9999**				**1.9685**	**2**	**-0.0315**	**AVERAGE**		**Y**

In the example in Table 4.2.1.6, missing post-baseline values are imputed by last observation carried forward, and also by worst observation carried forward.
In this study, at Week 8, there is a scheduled visit (visit number 6). At that visit, blood pressure should be collected. However, for this subject, either there was no visit 6, or there was a visit 6, but no data on blood pressure were collected. The SAP says that missing post-baseline data should be imputed (derived) by two methods: LOCF (last observation carried forward), and WOCF (worst observation carried forward).

For LOCF analysis, the missing Week 8 (VISITNUM 6) result is imputed by carrying forward the most recent prior available post-baseline value, which is the VISITNUM 5 value. That the Week 8 value is imputed is indicated by LOCF in the derivation type (DTYPE) column.

For WOCF analysis, even though the unscheduled VISITNUM 4.1 value was not chosen to represent the Week 2 analysis timepoint, it is used to impute the missing Week 8 timepoint because it was the worst post-baseline result up to that point.

The exact algorithms employed in the record derivation methods (LOCF and WOCF in this case) must be indicated in the metadata for DTYPE.

Traceability is enhanced by the addition of the SDTM VISITNUM and --SEQ columns. The combination of USUBJID and VSSEQ provides a link to the exact input record in the SDTM VS dataset. On the derived LOCF and WOCF rows, VISITNUM and VSSEQ provide clarity about where the value came from. There are several other concepts presented in this example. Analysis relative day (ADY) in this protocol is defined relative to date of first dose. In many but not all protocols, ADY would equal the value of the SDTM --DY variable (or --STDY for some kinds of data). The data presented here illustrate that this particular subject did not take drug until two days after randomization, so the value of ADY is -2 at the randomization visit, Visit 3 (VISITNUM 3). As is the case for SDTM study day, there is no day 0 for ADY.

In this protocol, if there are multiple datapoints within an analysis time window, the value that is observed closest to a pre-specified target planned relative day is the value that is chosen to represent the analysis timepoint. For this study and parameter, AWTARGET = VISITDY (Planned Study Day) from SDTM, and ADY=VSDY. AWTDIFF is the absolute value of ADY - AWTARGET, adjusted for the fact that there is no day 0 (so that if ADY and AWTARGET have different signs, then AWTDIFF = |ADY - AWTARGET| - 1).

For AVISIT=Week 2, there were two values observed, at study days 13 and 17 (rows 4 and 5). Day 13 is closer to the target, day 14. So the day 13 record (row 4) is chosen for analysis, as denoted by the analysis flag ANL01FL = Y.

AVISIT by itself functions as a description of an analysis time window. AVISIT, DTYPE, and ANL01FL are all needed to identify the records to be used in a given analysis.

On the derived AVIST=Week 8 records, AWTARGET was set to the target for Week 8, and AWTDIFF was calculated accordingly. It did not make sense to retain the values of AWTARGET and AWTDIFF from the original records.

Table 4.2.1.6 Illustration of Rule 3: Creation of New Rows to Handle Imputation of Missing Values by Last Observation Carried Forward and Worst Observation Carried Forward

Row	PARAM	AVISIT	AVISITN	VISITNUM	VSSEQ	ABLFL	AVAL	BASE	CHG	DTYPE	ADY	AWTARGET	AWTDIFF	ANL01FL
1	Systolic BP (mm Hg)	Screening	-4	1	3821		120	114	.		-30	-28	2	Y
2	Systolic BP (mm Hg)	Run-In	-2	2	3822		116	114	.		-16	-14	2	Y
3	Systolic BP (mm Hg)	Week 0	0	3	3823	Y	114	114	0		-2	1	2	Y
4	Systolic BP (mm Hg)	Week 2	2	4	3824		118	114	4		13	14	1	Y
5	Systolic BP (mm Hg)	Week 2	2	4.1	3825		126	114	12		17	14	3	
6	Systolic BP (mm Hg)	Week 4	4	5	3826		122	114	8		23	28	5	Y
7	Systolic BP (mm Hg)	Week 8	8	5	3826		122	114	8	LOCF	23	56	33	Y
8	Systolic BP (mm Hg)	Week 8	8	4.1	3825		126	114	12	WOCF	17	56	39	Y
9	Systolic BP (mm Hg)	Week 12	12	7	3827		134	114	20		83	84	1	Y

Table 4.2.1.7 contains an example of data supporting change from baseline analyses of migraine pain. In this study, missing post-baseline data are imputed by the methods of Baseline Observation Carried Forward (BLOCF) and Last Observation Carried Forward (LOCF).

When a migraine headache occurs, subjects self-administer a single dose of blinded study treatment. Subjects assess migraine pain at planned timepoints Pre-Dose, 30 Minutes Post-Dose, 1 Hour Post-Dose, and 2 Hours Post-Dose. Collected data on migraine pain are tabulated in the SDTM Findings About domain.

ATPT is the analysis timepoint description. ATPTN is the analysis timepoint number. FATPTNUM is the collected timepoint number from SDTM. AVALC contains the pain assessment, and AVAL contains the numeric coded value of the assessment. AVAL is a one-to-one map to AVALC.

Subject 000276 did not continue to provide data after 1 Hour Post-Dose. For this subject, the 2 Hours Post-Dose planned observation must be imputed. Therefore, subject 000276 is excluded from an observed case analysis of Migraine Pain at 2 Hours Post-Dose.

Subject 001863 had complete data, so no imputation was necessary.

The data for both subjects are included in the BLOCF and LOCF analyses of Migraine Pain at 2 Hours Post-Dose.

Table 4.2.1.7 Illustration of Rule 3: Creation of New Rows to Handle Imputation of Missing Values by Baseline Observation Carried Forward and Last Observation Carried Forward

Row	USUBJID	TRTP	PARAM	ATPT	ATPTN	FATPTNUM	FASEQ	ABLFL	AVAL	AVALC	BASE	CHG	DTYPE
1	000276	Placebo	Migraine Pain	Pre-Dose	0	1	14	Y	3	Severe Pain	3	0	
2	000276	Placebo	Migraine Pain	30 Minutes Post-Dose	0.5	2	22		2	Moderate Pain	3	-1	
3	000276	Placebo	Migraine Pain	1 Hour Post-Dose	1	3	27		1	Mild Pain	3	-2	
4	**000276**	Placebo	**Migraine Pain**	**2 Hours Post-Dose**	**2**	**1**	**14**		**3**	**Severe Pain**	**3**	**0**	**BLOCF**
5	**000276**	Placebo	**Migraine Pain**	**2 Hours Post-Dose**	**2**	**3**	**27**		**1**	**Mild Pain**	**3**	**-2**	**LOCF**
6	001863	Soma 30 mg	Migraine Pain	Pre-Dose	0	1	638	Y	3	Severe Pain	3	0	
7	001863	Soma 30 mg	Migraine Pain	30 Minutes Post-Dose	0.5	2	639		1	Mild Pain	1	-2	
8	001863	Soma 30 mg	Migraine Pain	1 Hour Post-Dose	1	3	640		1	Mild Pain	1	-2	
9	001863	Soma 30 mg	Migraine Pain	2 Hours Post-Dose	2	4	641		1	Mild Pain	1	-2	

Table 4.2.1.8 contains an example of some of the columns in a dataset supporting analysis of a 2-period crossover study.

In a crossover trial design, all subjects are planned to receive all of the study treatments. The sequence of treatments is randomized. If in a study there are two treatments in a crossover design, two treatment periods are necessary.

In this example, the planned visits are 1 (Screening and beginning of placebo run-in period), 2 (Week -2, halfway through placebo run-in period), 3 (Week 0, end of placebo run-in and randomization), 4 (Week 4, the end of the first treatment period), and 5 (Week 8, the end of the second treatment period). Baseline is defined in the Statistical Analysis Plan as the average of the Week -2 (VISIT 2) and Week 0 (VISIT 3) measurements. This baseline is used for the analysis of both the first and the second crossover periods. USUBJID 0987_4252 has no VISIT 2 measurement, so the average is just the Week 0 (VISIT 3) measurement.

Within any post-baseline week window, the last observation is used to characterize that week. For example, for USUBJID 0987_3984, the VISIT 5 (row 7) value is used to characterize AVISIT=Week 8, as opposed to the earlier VISIT 4.1 value (row 6), which was also observed during the Week 8 time window. The variable ANL01FL is used in this study to identify the record selected for analysis when there are multiple records for a given AVISIT, and must be used in conjunction with other selection variables in order to identify the exact set of records used in a given analysis or summary.

APERIODC is the crossover period character description.

Note that in general, APERIODC is not the same as EPOCH. For example, it is possible in some cases that boundaries of APERIODs would not align exactly with boundaries of EPOCHs. A simple example is a post-discontinuation record that is associated with the most recent treatment period for analysis.

TRTSEQP, from ADSL, is the planned ordering of crossover treatments. TRTP is the treatment variable that will be used in the analysis of this dataset. The two endpoint records are derived only for the subjects who have data for both periods.

The conventions used in AVISITN are producer-defined. In this example, the producer has decided that AVISITN contains -8888 for the derived baseline records, 9999 for the derived endpoint records, and week number otherwise.

It should be noted that in this example, the producer elected to define APERIOD only for the on-treatment visits, therefore leaving TRTP, APERIOD, and APERIODC empty on other records. This is not meant to imply a standard or best practice.

Table 4.2.1.8 Illustration of Rule 3: Creation of Endpoint Rows to Facilitate Analysis of a Crossover Design

Row	USUBJID	PARAMCD	AVISIT	AVISITN	VISITNUM	DTYPE	ANL01FL	TRTP	APERIOD	APERIODC	TRTSEQP	AVAL	ABLFL	BASE	CHG
1	0987_3984	ALT	Screening	-4	1		Y				Drug B, Drug A	16		17	.
2	0987_3984	ALT	Week -2	-2	2		Y				Drug B, Drug A	16		17	.
3	0987_3984	ALT	Week 0	0	3		Y				Drug B, Drug A	18		17	.
4	0987_3984	ALT	Baseline	-8888		**AVERAGE**	Y				Drug B, Drug A	17	Y	17	0
5	0987_3984	ALT	Week 4	4	4		Y	Drug B	1	Period 1	Drug B, Drug A	14		17	-3
6	0987_3984	ALT	Week 8	8	4.1			Drug A	2	Period 2	Drug B, Drug A	10		17	-7
7	0987_3984	ALT	Week 8	8	5		Y	Drug A	2	Period 2	Drug B, Drug A	12		17	-5
8	**0987_3984**	**ALT**	**Endpoint**	**9999**	**4**	**ENDPOINT**	**Y**	**Drug B**	**1**	**Period 1**	**Drug B, Drug A**	**14**		**17**	**-3**
9	**0987_3984**	**ALT**	**Endpoint**	**9999**	**5**	**ENDPOINT**	**Y**	**Drug A**	**2**	**Period 2**	**Drug B, Drug A**	**12**		**17**	**-5**
10	0987_4252	ALT	Screening	-4	1		Y				Drug A, Drug B	12		11	.
11	0987_4252	ALT	Week 0	0	3		Y				Drug A, Drug B	11		11	.
12	0987_4252	ALT	Baseline	-8888		**AVERAGE**	Y				Drug A, Drug B	11	Y	11	0
13	0987_4252	ALT	Week 4	4	4		Y	Drug A	1	Period 1	Drug A, Drug B	14		11	3
14	0987_4252	ALT	Week 8	8	5		Y	Drug B	2	Period 2	Drug A, Drug B	15		11	4
15	**0987_4252**	**ALT**	**Endpoint**	**9999**	**4**	**ENDPOINT**	**Y**	**Drug A**	**1**	**Period 1**	**Drug A, Drug B**	**14**		**11**	**3**
16	**0987_4252**	**ALT**	**Endpoint**	**9999**	**5**	**ENDPOINT**	**Y**	**Drug B**	**2**	**Period 2**	**Drug A, Drug B**	**15**		**11**	**4**

Rule 4. A function of multiple rows within a parameter should be added as a new parameter.

Rule 4 is a special case of Rule 2. The functions covered by this rule violate the second condition of Rule 1 (they are not same-row functions of AVAL), and may also violate the first and third conditions.

Table 4.2.1.9 shows an example of a clinical trial of a Human Immunodeficiency Virus (HIV) vaccine, where blood samples are drawn at each visit, and CD4 cell count is measured. To assess efficacy, it is important to look at the cumulative effect over time on CD4 cell count during follow-up after administration.

Let AVAL(t) equal the value of CD4 cell count at post-baseline visit t, and let VISITDY(t) be the planned study day of visit t.

CD4AUC (cumulative daily CD4 count over follow-up) is calculated at any given post-baseline visit as follows:

- CD4AUC at baseline visit is set to 0.
- CD4AUC(t) = CD4AUC(t-1) + [0.5 * AVAL(t-1) + 0.5 * AVAL(t)] * [VISITDY(t) - VISITDY(t-1)].

CD4AUC is not a simple same-row function of BASE and AVAL. It is calculated based on data from multiple observations (rows) of CD4 data, so it should be added as a new parameter rather than as a new column. CD4AUC is not defined pre-baseline, which is why there is no Week -1 for this parameter.

CD4AUCMB (cumulative average change from baseline in daily CD4 count over follow-up) is calculated as

- CD4AUCMB(t) = CD4AUC(t) / [VISITDY(t) - 1] - baseline value of CD4 cell count.

CD4AUCMB is a function of both CD4AUC and the baseline value of CD4, so it also must be its own parameter (see Rule 5 below). CD4AUCMB is not defined for pre-baseline and baseline records and therefore these records are not represented within this value of PARAM.

Table 4.2.1.9 Illustration of Rule 4: Creation of a New Parameter to Handle a Function of More Than One Row of a Parameter

Row	PARAM	PARAMCD	AVISIT	VISITDY	ABLFL	AVAL	BASE
1	CD4 (cells/mm3)	CD4	Week -1	-7		75	76
2	CD4 (cells/mm3)	CD4	Week 0	1	Y	76	76
3	CD4 (cells/mm3)	CD4	Week 2	15		128	76
4	CD4 (cells/mm3)	CD4	Week 4	29		125	76
5	CD4 (cells/mm3)	CD4	Week 8	57		191	76
6	CD4 (cells/mm3)	CD4	Week 12	85		167	76
7	CD4 (cells/mm3)	CD4	Week 16	113		136	76
8	**CD4 Cumulative AUC**	**CD4AUC**	**Week 0**	**1**	**Y**	**0**	**0**
9	**CD4 Cumulative AUC**	**CD4AUC**	**Week 2**	**15**		**1428**	**0**
10	**CD4 Cumulative AUC**	**CD4AUC**	**Week 4**	**29**		**3199**	**0**
11	**CD4 Cumulative AUC**	**CD4AUC**	**Week 8**	**57**		**7623**	**0**
12	**CD4 Cumulative AUC**	**CD4AUC**	**Week 12**	**85**		**12635**	**0**
13	**CD4 Cumulative AUC**	**CD4AUC**	**Week 16**	**113**		**16877**	**0**
14	**CD4 Cumulative AUCMB**	**CD4AUCMB**	**Week 2**	**15**		**26**	**.**
15	**CD4 Cumulative AUCMB**	**CD4AUCMB**	**Week 4**	**29**		**38.25**	**.**
16	**CD4 Cumulative AUCMB**	**CD4AUCMB**	**Week 8**	**57**		**60.125**	**.**
17	**CD4 Cumulative AUCMB**	**CD4AUCMB**	**Week 12**	**85**		**74.4167**	**.**
18	**CD4 Cumulative AUCMB**	**CD4AUCMB**	**Week 16**	**113**		**74.6875**	**.**

Rule 5. A function of more than one parameter should be added as a new parameter.

There is often a need to derive for analysis a parameter that was not collected. Such parameters may be quite complex functions of data from multiple SDTM domains and domain classes. Rule 5 addresses the case where a parameter is derived from other parameters already present in the dataset.

For example, a questionnaire total domain score is calculated as a function of more than one observed question. The total domain score should be added as a new parameter, with its corresponding set of derived rows. For this derived parameter, the value of PARAM would be e.g., “Total Domain Score”, and the value of the total domain score would be stored in the standard AVAL column, the baseline value would be stored in the standard BASE column, change from baseline would be stored in CHG, as usual.

In the example in Table 4.2.1.10, blood samples are drawn at every visit, and laboratory test measurements of total cholesterol and high-density lipoprotein cholesterol are found in the SDTM LB dataset. The protocol calls for analysis of each individual lab analyte, and also for an analysis of the ratio of total cholesterol (C) to high-density lipoprotein (HDL) cholesterol. The ADaM dataset contains parameters for each of the two measured lab tests, as well as a new set of derived rows where the description in PARAM is “Total Cholesterol:HDL-C ratio”, and AVAL contains the calculated ratio at each timepoint.

The analysis of percent change from baseline (PCHG) is of interest for all three parameters and is therefore populated on all records. In general, however, if percent change is not analyzed for a particular value of PARAM, then it is not necessary to populate PCHG for those rows.

Table 4.2.1.10 Illustration of Rule 5: Creation of New Parameter to Handle a Function of More Than One Parameter

Row	PARAM	PARAMCD	AVISIT	AVISITN	VISITNUM	LBSEQ	ABLFL	AVAL	BASE	CHG	PCHG
1	Total Cholesterol (mg/dL)	CHOL	Screening	-2	1	39394		265	266	.	.
2	Total Cholesterol (mg/dL)	CHOL	Run-In	-1	2	25593		278	266	.	.
3	Total Cholesterol (mg/dL)	CHOL	Week 0	0	3	23213	Y	266	266	0	0.000
4	Total Cholesterol (mg/dL)	CHOL	Week 2	2	4	32952		259	266	-7	-2.632
5	Total Cholesterol (mg/dL)	CHOL	Week 4	4	5	12768		235	266	-31	-11.654
6	Total Cholesterol (mg/dL)	CHOL	Week 8	8	6	18773		242	266	-24	-9.023
7	Total Cholesterol (mg/dL)	CHOL	Week 12	12	7	28829		217	266	-49	-18.421
8	High-Density Lipoprotein Chol (mg/dL)	HDL	Screening	-2	1	32437		44	42	.	.
9	High-Density Lipoprotein Chol (mg/dL)	HDL	Run-In	-1	2	26884		40	42	.	.
10	High-Density Lipoprotein Chol (mg/dL)	HDL	Week 0	0	3	52657	Y	42	42	0	0.000
11	High-Density Lipoprotein Chol (mg/dL)	HDL	Week 2	2	4	38469		43	42	1	2.381
12	High-Density Lipoprotein Chol (mg/dL)	HDL	Week 4	4	5	12650		47	42	5	11.905
13	High-Density Lipoprotein Chol (mg/dL)	HDL	Week 8	8	6	24345		46	42	4	9.524
14	High-Density Lipoprotein Chol (mg/dL)	HDL	Week 12	12	7	23484		47	42	5	11.905
15	**Total Cholesterol:HDL-C ratio**	**CHOLH**	**Screening**	**-2**	**1**			**6.023**	**6.333**	**.**	**.**
16	**Total Cholesterol:HDL-C ratio**	**CHOLH**	**Run-In**	**-1**	**2**			**6.950**	**6.333**	**.**	**.**
17	**Total Cholesterol:HDL-C ratio**	**CHOLH**	**Week 0**	**0**	**3**		**Y**	**6.333**	**6.333**	**0.000**	**0.000**
18	**Total Cholesterol:HDL-C ratio**	**CHOLH**	**Week 2**	**2**	**4**			**6.023**	**6.333**	**-0.310**	**-4.896**
19	**Total Cholesterol:HDL-C ratio**	**CHOLH**	**Week 4**	**4**	**5**			**5.000**	**6.333**	**-1.333**	**-21.053**
20	**Total Cholesterol:HDL-C ratio**	**CHOLH**	**Week 8**	**8**	**6**			**5.261**	**6.333**	**-1.072**	**-16.934**
21	**Total Cholesterol:HDL-C ratio**	**CHOLH**	**Week 12**	**12**	**7**			**4.617**	**6.333**	**-1.716**	**-27.100**

Rule 6. When there is more than one definition of baseline, each additional definition of baseline requires the creation of its own set of rows.

In case there is more than one definition of baseline in an ADaM dataset, new rows must be created for each additional alternative definition of baseline. There will therefore be multiple sets of rows, where each set of rows corresponds to a particular definition of baseline. Whenever there is more than one definition of baseline, the BASETYPE column is required. BASETYPE identifies the definition of baseline that corresponds to the value of BASE in each row. There is only one BASE column, and only one column for each qualifying function of AVAL and BASE.

The example in Table 4.2.1.11 presents a dataset supporting shift analysis from three different baselines. Accordingly, it makes use of the BASETYPE variable described above. The ANRIND, BNRIND, and SHIFTy variables are also illustrated. In this example, the three baselines of interest characterize different portions of the study: run-in, double-blind, and open-label. For any datapoint, it is desired to have the ability to analyze shift from the most recent baseline or any prior baseline. Rows 1-12 are the initial set of rows representing all of the collected data. They permit analysis of the shift in normal range indicator from the run-in baseline to any value in the run-in, double-blind, or open-label portions of the study. Additional sets of rows are added to support analysis of shift from the double-blind and open-label baselines: rows 13-19 permit analysis of the shift from the double-blind baseline normal range indicator for data in either the double-blind or open-label portions of the study; and rows 20-22 support analysis of shift from open-label baseline for data in the open-label portion of the study.

Note that only the rows needed for the analysis are included in the additional sets. For example, the set of rows for the shift from the double-blind baseline does not include the rows for EPOCH=”RUN-IN” and EPOCH=”STABILIZATION” as they are not analyzed using the double-blind baseline.

For space reasons, the ANLzzFL variable is not shown, although it would be needed to identify which record is selected in cases of multiple observed records within an analysis timepoint, as is the case for AVISIT=WEEK 12 (DB) for this subject and parameter.

Table 4.2.1.11 Illustration of Rule 6: Creation of New Rows to Handle Multiple Baseline Definitions - Supporting Comparisons to Any Prior Baseline

Row	BASETYPE	EPOCH	AVISIT	LBSEQ	AVAL	ANRLO	ANRHI	ANRIND	ABLFL	BASE	BNRIND	SHIFT1
1	RUN-IN	RUN-IN	BSLN (RUN-IN)	111	34.5	15.4	48.5	NORMAL	Y	34.5	NORMAL	
2	RUN-IN	RUN-IN	WK 8 (RUN-IN)	168	11.6	15.4	48.5	LOW		34.5	NORMAL	NORMAL to LOW
3	RUN-IN	RUN-IN	END POINT (RUN-IN)	168	11.6	15.4	48.5	LOW		34.5	NORMAL	NORMAL to LOW
4	RUN-IN	STABILIZATION	WK 14 (STAB.)	200	13.1	15.4	48.5	LOW		34.5	NORMAL	NORMAL to LOW
5	RUN-IN	STABILIZATION	END POINT (STAB.)	200	13.1	15.4	48.5	LOW		34.5	NORMAL	NORMAL to LOW
6	RUN-IN	DOUBLE-BLIND	BSLN (DB)	200	13.1	15.4	48.5	LOW		34.5	NORMAL	NORMAL to LOW
7	RUN-IN	DOUBLE-BLIND	WK 12 (DB)	295	13.7	15.4	48.5	LOW		34.5	NORMAL	NORMAL to LOW
8	RUN-IN	DOUBLE-BLIND	WK 12 (DB)	300	19.7	15.4	48.5	NORMAL		34.5	NORMAL	NORMAL to NORMAL
9	RUN-IN	DOUBLE-BLIND	END POINT (DB)	300	19.7	15.4	48.5	NORMAL		34.5	NORMAL	NORMAL to NORMAL
10	RUN-IN	OPEN-LABEL	BSLN (OPEN)	300	19.7	15.4	48.5	NORMAL		34.5	NORMAL	NORMAL to NORMAL
11	RUN-IN	OPEN-LABEL	WK 24 (OPEN)	350	28.1	15.4	48.5	NORMAL		34.5	NORMAL	NORMAL to NORMAL
12	RUN-IN	OPEN-LABEL	END POINT (OPEN)	350	28.1	15.4	48.5	NORMAL		34.5	NORMAL	NORMAL to NORMAL
13	DBL-BLIND	DOUBLE-BLIND	BSLN (DB)	200	13.1	15.4	48.5	LOW	Y	13.1	LOW	
14	DBL-BLIND	DOUBLE-BLIND	WK 12 (DB)	295	13.7	15.4	48.5	LOW		13.1	LOW	LOW to LOW
15	DBL-BLIND	DOUBLE-BLIND	WK 12 (DB)	300	19.7	15.4	48.5	NORMAL		13.1	LOW	LOW to NORMAL
16	DBL-BLIND	DOUBLE-BLIND	END POINT (DB)	300	19.7	15.4	48.5	NORMAL		13.1	LOW	LOW to NORMAL
17	DBL-BLIND	OPEN-LABEL	BSLN (OPEN)	300	19.7	15.4	48.5	NORMAL		13.1	LOW	LOW to NORMAL
18	DBL-BLIND	OPEN-LABEL	WK 24 (OPEN)	350	28.1	15.4	48.5	NORMAL		13.1	LOW	LOW to NORMAL
19	DBL-BLIND	OPEN-LABEL	END POINT (OPEN)	350	28.1	15.4	48.5	NORMAL		13.1	LOW	LOW to NORMAL

Row	BASETYPE	EPOCH	AVISIT	LBSEQ	AVAL	ANRLO	ANRHI	ANRIND	ABLFL	BASE	BNRIND	SHIFT1
20	OPEN-LABEL	OPEN-LABEL	BSLN (OPEN)	300	19.7	15.4	48.5	NORMAL	Y	19.7	NORMAL	
21	OPEN-LABEL	OPEN-LABEL	WK 24 (OPEN)	350	28.1	15.4	48.5	NORMAL		19.7	NORMAL	NORMAL to NORMAL
22	OPEN-LABEL	OPEN-LABEL	END POINT (OPEN)	350	28.1	15.4	48.5	NORMAL		19.7	NORMAL	NORMAL to NORMAL

The example in Table 4.2.1.11 supports the ability to analyze shift from the most recent baseline or any prior baseline. In contrast, if it is needed only to have the ability to analyze shift from the most recent baseline, then the dataset does not need as many rows. Table 4.2.1.12 illustrates an arrangement supporting analysis from the most recent baseline only. Because there is more than one definition of baseline, the BASETYPE variable is still needed.

Table 4.2.1.12 Illustration of Rule 6: Creation of New Rows to Handle Multiple Baseline Definitions - Supporting Comparison to Most Recent Baseline

Row	BASETYPE	EPOCH	AVISIT	LBSEQ	AVAL	ANRLO	ANRHI	ANRIND	ABLFL	BASE	BNRIND	SHIFT1
1	RUN-IN	RUN-IN	BSLN (RUN-IN)	111	34.5	15.4	48.5	NORMAL	Y	34.5	NORMAL	
2	RUN-IN	RUN-IN	WK 8 (RUN-IN)	168	11.6	15.4	48.5	LOW		34.5	NORMAL	NORMAL to LOW
3	RUN-IN	RUN-IN	END POINT (RUN-IN)	168	11.6	15.4	48.5	LOW		34.5	NORMAL	NORMAL to LOW
4	RUN-IN	STABILIZATION	WK 14 (STAB.)	200	13.1	15.4	48.5	LOW		34.5	NORMAL	NORMAL to LOW
5	RUN-IN	STABILIZATION	END POINT (STAB.)	200	13.1	15.4	48.5	LOW		34.5	NORMAL	NORMAL to LOW
6	DBL-BLIND	DOUBLE-BLIND	BSLN (DB)	200	13.1	15.4	48.5	LOW	Y	13.1	LOW	
7	DBL-BLIND	DOUBLE-BLIND	WK 12 (DB)	295	13.7	15.4	48.5	LOW		13.1	LOW	LOW to LOW
8	DBL-BLIND	DOUBLE-BLIND	WK 12 (DB)	300	19.7	15.4	48.5	NORMAL		13.1	LOW	LOW to NORMAL
9	DBL-BLIND	DOUBLE-BLIND	END POINT (DB)	300	19.7	15.4	48.5	NORMAL		13.1	LOW	LOW to NORMAL
10	OPEN-LABEL	OPEN-LABEL	BSLN (OPEN)	300	19.7	15.4	48.5	NORMAL	Y	19.7	NORMAL	
11	OPEN-LABEL	OPEN-LABEL	WK 24 (OPEN)	350	28.1	15.4	48.5	NORMAL		19.7	NORMAL	NORMAL to NORMAL
12	OPEN-LABEL	OPEN-LABEL	END POINT (OPEN)	350	28.1	15.4	48.5	NORMAL		19.7	NORMAL	NORMAL to NORMAL

Table 4.2.1.11 and Table 4.2.1.12 illustrate example solutions in the case where different baselines are needed to characterize different portions of a study. In general, however, there might be other reasons that more than one definition of baseline might be needed. It could also be that there are multiple ways to construct a particular baseline value, e.g., last value prior to treatment, average value at the baseline visit, minimum value prior to treatment, etc.

For a given parameter, whenever there is more than one definition of baseline, BASETYPE is required and must be populated. For any given parameter and subject, whenever there is more than one definition of baseline, the number of records flagged with ABLFL=Y is equal to the number of values of BASETYPE.

4.3 Inclusion of All Observed and Derived Records for a Parameter versus the Subset of Records Used for Analysis

This section discusses whether the ADaM dataset should include all rows of an analysis parameter, or only the subset of rows that are used for analysis. A value of AVAL or AVALC for an analysis parameter at a specific timepoint may be observed (i.e., collected on the case report form or in an electronic diary at that timepoint), it may be imputed because it was missing, or it may be derived from a combination of other values.

To illustrate the issue being presented, assume that the total scores for Questionnaire A (administered at Visits 1, 2, and 3) are in the SDTM QS dataset as illustrated below. Any missing total scores are imputed by carrying the last post-baseline (post-Visit 1) total score forward. The total score for visit 3 will be analyzed.

In the SDTM QS dataset data shown below, subject 0001 has data for visits 1, 2, and 3; subject 0002 will not be included in the analysis, as there are no post-baseline data for the subject; subject 0003 has data for visits 1 and 2, but is missing data for visit 3.

Table 4.3.1 Illustration of Issue, Data as Found in SDTM QS Dataset

Row	DOMAIN	USUBJID	VISITNUM	QSSEQ	QSCAT	QSTESTCD	QSSTRESN
1	QS	0001	1	101	QUES-A	TOTSCORE	7
2	QS	0001	2	201	QUES-A	TOTSCORE	12
3	QS	0001	3	555	QUES-A	TOTSCORE	14
4	QS	0002	1	91	QUES-A	TOTSCORE	4
5	QS	0003	1	156	QUES-A	TOTSCORE	2
6	QS	0003	2	300	QUES-A	TOTSCORE	6

The questions that arise are whether or not the ADaM dataset should contain data for subject 0002 even though the subject is not included in the analysis and if the ADaM dataset should contain totals for visits 1 and 2 even though the data being analyzed are from visit 3.

4.3.1 ADaM Methodology and Examples

The ADaM methodology is to include all observed and derived rows for a given analysis parameter. The inclusion of all the rows in the ADaM dataset, including those not used in the analysis, requires a way to identify the rows used in the specified analysis. The advantage to this approach is that the inclusion of all rows makes it easier to verify that the selection and derived timepoint processing was done correctly, thus providing useful traceability. In addition, the data are also then available to enable other analyses, including sensitivity analyses. However, this approach increases the size of the dataset, and introduces a risk that the appropriate selection criteria will not be incorporated and thereby generate incorrect analysis results.

Regulatory reviewers prefer that the path followed in creating and/or selecting analysis rows be clearly delineated and traceable all the way back to the originating rows in the SDTM dataset, if possible and within reason. Simply including the algorithm in the metadata is often not sufficient, as any complicated data manipulations may not be clearly identified (e.g., how missing pieces of the input data were handled). Retaining in one dataset all of the observed and derived rows for the analysis parameter provides the clearest traceability in the most flexible manner within the standard BDS. The resulting dataset also provides the most flexibility for testing the robustness of an analysis (e.g., using a different imputation method).

Example 1

In the example discussed above (Table 4.3.1), the ADaM dataset would contain the following rows (Table 4.3.1.1) for the total score parameter:

Table 4.3.1.1 Example 1: ADaM Dataset

Row	PARAMCD	USUBJID	VISITNUM	AVISITN	AVISIT	AVAL	DTYPE	QSSEQ
1	TOTSCORE	0001	1	1	Visit 1	7		101
2	TOTSCORE	0001	2	2	Visit 2	12		201
3	TOTSCORE	0001	3	3	Visit 3	14		555
4	TOTSCORE	0002	1	1	Visit 1	4		91
5	TOTSCORE	0003	1	1	Visit 1	2		156
6	TOTSCORE	0003	2	2	Visit 2	6		300
7	TOTSCORE	0003	2	3	Visit 3	6	LOCF	300

For the analysis discussed above, the data to be analyzed are selected by specifying that AVISITN = 3 (or AVISIT=Visit 3).

It should be noted that this approach does not require the inclusion of all rows from the input dataset. For example, if the input dataset contains data for several different questionnaires, the extraneous data (e.g., for questionnaires other than the one being addressed) do not have to be included in the ADaM dataset.

Example 2

In the following example (Table 4.3.1.2 and Table 4.3.1.3), the Q01 assessment is scheduled to be performed at visits 1, 3, 5, and 7, and results are to be summarized at those visits. Subject 1099 has data for the assessment at visits 1, 2, and 7. (Note that though the assessment was not scheduled to be performed at Visit 2, the data show the assessment was performed at that time for that subject.) Subject 2001 is not in the Full Analysis Set. Subject 3023 has two assessments at visit 5, and the study's analysis plan specifies that only the first occurrence within a visit will be analyzed; however, as this subject does not have a visit 7 row in the data, the later of the visit 5 rows is carried forward into visit 7. The SDTM dataset that is the basis for the ADaM dataset has the following rows:

Table 4.3.1.2 Example 2: Data as Found in SDTM QS Dataset

Row	QSTESTCD	USUBJID	QSSEQ	VISITNUM	VISIT	QSSTRESN	QSDTC
1	Q01	1099	111	1	BASELINE	25	2005-04-04
2	Q01	1099	121	2	VISIT 2	24	2005-05-02
3	Q01	1099	132	7	VISIT 7	15	2005-08-22
4	Q01	2001	150	1	BASELINE	27	2005-02-05
5	Q01	3023	117	1	BASELINE	31	2005-06-30
6	Q01	3023	123	3	VISIT 3	29	2005-07-25
7	Q01	3023	134	5	VISIT 5	28	2005-08-20
8	Q01	3023	135	5	VISIT 5	25	2005-08-21

The ADaM dataset contains rows corresponding to those found in SDTM as well as rows created by LOCF for the missing visit assessments, together with the flags and other columns needed to identify the rows to be included in a given analysis:

Table 4.3.1.3 Example 2: ADaM Dataset

Row	PARAMCD	USUBJID	VISITNUM	VISIT	AVISITN	AVISIT	AVAL	DTYPE	ANL01FL	FASFL	QSSEQ
1	Q01	1099	1	BASELINE	1	BASELINE	25		Y	Y	111
2	Q01	1099	2	VISIT 2			24			Y	121
3	Q01	1099	2	VISIT 2	3	VISIT 3	24	LOCF	Y	Y	121
4	Q01	1099	2	VISIT 2	5	VISIT 5	24	LOCF	Y	Y	121
5	Q01	1099	7	VISIT 7	7	VISIT 7	15		Y	Y	132
6	Q01	2001	1	BASELINE	1	BASELINE	27		Y	N	150
7	Q01	3023	1	BASELINE	1	BASELINE	31		Y	Y	117
8	Q01	3023	3	VISIT 3	3	VISIT 3	29		Y	Y	123
9	Q01	3023	5	VISIT 5	5	VISIT 5	28		Y	Y	134
10	Q01	3023	5	VISIT 5	5	VISIT 5	25			Y	135
11	Q01	3023	5	VISIT 5	7	VISIT 7	25	LOCF	Y	Y	135

Selection criteria applicable to this example include:

- DTYPE null identifies the data as found in the SDTM dataset.
- DTYPE="LOCF" specifies the method used to derive the added rows, and indicates that those rows were derived.
- FASFL="Y" (copied from ADSL) identifies the subjects who are members of the Full Analysis Set.
- ANL01FL="Y" identifies the rows chosen to represent each AVISIT. There were multiple observations for subject 3023 at AVISITN=5 and therefore in this example, rows with ANL01FL="Y" are the ones that have been chosen to represent their respective analysis timepoints.
- ANL01FL=null for subject 1099 for VISIT="VISIT 2" (row 2) because visit 2 is an unscheduled visit for this questionnaire and Visit 2 will not be presented in the analyses; AVISITN and AVISIT are also null because they do not map to visits used for analyses described in the study's analysis plan.
- The combination of "(ANL01FL="Y" and FASFL="Y" and AVISITN=5)" identifies the rows used in a FAS analysis of Visit 5 data.

Approaches Considered and Not Adopted
The other approach considered was to include in the ADaM dataset only the rows that are actually used in the analysis of the analysis parameter. In Example 1 above, only Visit 3 rows that were either observed or derived by LOCF would be included in the ADaM dataset. The main advantage of this approach would be to simplify the analysis, as no selection clause would need to be used to identify the appropriate rows for inclusion in the analysis. However, the primary disadvantages would be the loss of traceability and the loss of flexibility for testing the robustness of the analysis. Because of these disadvantages, this approach was not chosen.

4.4 Inclusion of Input Data that are not Analyzed but that Support a Derivation in the ADaM Dataset

Section 4.3 states that for a given analysis parameter, all observed and derived rows of that parameter should be included in the dataset, not just the rows that are used in the analysis. Section 4.3 is a simple case of a more general topic addressed here in Section 4.4.

This section addresses the broader issue of whether an ADaM dataset should contain the input data used in the derivation of the analysis data as well as the actual data being analyzed. This includes:

- Input data rows and columns to support traceability of the derivation of analyzed rows and columns, and
- Raw or derived predecessor parameters that are not analyzed themselves but are used to derive an analyzed parameter.

The above input data rows and columns could come from one SDTM dataset or multiple datasets as necessary to derive the analysis data captured in AVAL or AVALC, as described by the analysis parameter.

4.4.1 ADaM Methodology and Examples

ADaM datasets are developed to facilitate intended analyses. In the ADaM model, it is assumed that the original data sources for ADaM datasets are SDTM datasets, even when ADaM datasets are derived from other ADaM datasets. ADaM has features that enable traceability from analysis results to ADaM datasets and from ADaM datasets to SDTM datasets.

The ADaM methodology to achieve the expected traceability is to describe the derivation algorithms in the metadata and, if practical and feasible, to include supportive *rows* as appropriate for traceability. To include the input data as rows in the ADaM dataset, columns should be added where feasible to indicate the source of the input data – domain, variable name, and sequence number. While this methodology increases both the size of the dataset and the complexity of selecting the appropriate rows for analysis, it also provides input data in an immediately accessible manner. In addition, intermediate values can be retained if appropriate flags are used to distinguish them.

In general, it is strongly recommended to include as much supporting data as is needed for traceability. However, there are situations in which it may not be practical. For example, if an analyzed parameter is a summary derived from a very large number of raw e-diary input records, it may be neither useful nor practical to include all of the raw e-diary records as rows in the ADaM dataset.

The remainder of this section addresses cases where the ADaM datasets contain not only the analysis data but also input data that are necessary to provide clearer traceability of the algorithms used to derive the analysis data. In addition to the actual values used in the analysis, the dataset may include rows not used in the analysis, rows containing input data, and rows containing intermediate values computed during the derivation of the analysis data. Flags or other columns are used to distinguish the various data types as well as to provide a traceable path from the input data to the value used in the analysis. The analysis results metadata specify how the appropriate rows are identified (by a specific selection clause). The identification of rows used in an analysis is addressed in Sections 4.5 and 4.6.

Unless the input data are already present as column(s) on the row (e.g., as covariate(s) or supportive variable(s)), the input data will be retained as rows in the ADaM dataset. The analysis value column (AVAL and/or AVALC) on the retained input data row will contain a value for the analysis parameter. Not all columns from the input dataset are carried into the ADaM dataset; instead additional variables will be included indicating the source of the input data – domain, variable name, and sequence number. This approach will allow the inclusion of input data from multiple domains. If the input data are already included in columns on the analysis parameter row (e.g., as covariates or supportive information), there is no need to include additional rows for those input data. The decision on keeping the input data as rows or columns will therefore be dictated by the types of input data and whether they are used for other purposes in the ADaM dataset.

Retaining in one dataset all data used in the determination of the analysis parameter value will provide the clearest traceability in the most flexible manner within the standard ADaM BDS. This large dataset also provides the most flexibility for testing the robustness of an analysis.

If it is determined that this large dataset is too cumbersome, the producer can choose to provide two datasets, one that contains all rows and another that is a subset of the first, containing only the rows used in the specified analysis. To ensure traceability, the metadata for the subset ADaM dataset will refer back to the full ADaM dataset as the immediate predecessor. Though this approach provides the needed traceability as well as providing a dataset that can be used in an analysis without specifying a selection clause, the total file size is even larger. More importantly, the developer will need to ensure consistency is maintained between the two datasets and validation will need to be done for both datasets. There is also potential confusion about which dataset supported an analysis, if analysis results metadata is not provided for that analysis.

Example 1

An ADaM dataset is created to support time-to-event analysis of a hypertension event. The analysis parameter is the study day of a hypertension event, defined to be the earliest study day among those of the following events: hospital admission, diastolic blood pressure exceeded 90, and systolic blood pressure exceeded 140. If a subject does not experience any of these events, the subject will be analyzed as censored on the day he/she exited the study.

Table 4.4.1.1 Example 1: Data as Found in SDTM VS Dataset

Row	USUBJID	VISITNUM	VSSEQ	VSDTC	VSDY	VSTESTCD	VSSTRESN
1	2010	1	22	2004-08-05	1	SYSBP	115
2	2010	1	23	2004-08-05	1	DIABP	75
3	2010	2	101	2004-08-12	8	SYSBP	120
4	2010	2	102	2004-08-12	8	DIABP	90
5	2010	3	207	2004-08-19	15	SYSBP	135
6	2010	3	208	2004-08-19	15	DIABP	92
7	2010	4	238	2004-08-25	21	SYSBP	138
8	2010	4	239	2004-08-25	21	DIABP	95
9	3082	1	27	2004-09-08	1	SYSBP	120
10	3082	1	28	2004-09-08	1	DIABP	80
11	3082	2	119	2004-09-15	8	SYSBP	125
12	3082	2	120	2004-09-15	8	DIABP	84

Table 4.4.1.2 Example 1: Data as Found in SDTM DS Dataset

Row	USUBJID	DSSEQ	DSSTDTC	DSSTDY	DSDECOD	DSTERM
1	2010	25	2004-08-05	1	RANDOMIZED	Subject Randomized
2	2010	301	2004-08-26	22	COMPLETED	Subject Completed
3	3082	20	2004-09-08	1	RANDOMIZED	Subject Randomized
4	3082	130	2004-09-17	10	COMPLETED	Subject Completed

Table 4.4.1.3 Example 1: Data as Found in SDTM HO Dataset

Row	USUBJID	HOSEQ	HOTERM	HODECOD	HOSTDTC	HOENDTC	HOSTDY	HOENDY
1	2010	99	HOSPITAL	HOSPITAL	2004-08-13	2004-08-15	9	11
2	2010	199	HOSPITAL	HOSPITAL	2004-08-20	2004-08-22	16	18

The ADaM dataset contains the sub-event data used to derive the analysis parameter “HYPEREVT”.

The ADaM methodology is illustrated in Table 4.4.1.4. Using this methodology, one would include all of the sub-events as analysis parameters (i.e., rows) and create the input domain, input variable, and input sequence columns (SRC* columns) to identify where the input rows came from. AVAL for PARAMCD=“HOSPADM” is the earliest relative day of hospitalization. AVAL for PARAMCD=“DBP” is the earliest relative day that diastolic blood pressure exceeded 90. AVAL for PARAMCD=“SBP” is the earliest relative day that systolic blood pressure exceeded 140. If a subject did not experience a particular sub-event, a row is still created for that sub-event indicating the subject was censored (CNSR=1) on the day the subject exited the study and the SRC* columns reference the DS dataset. AVAL for PARAMCD=“HYPEREVT” is derived as the earliest event of the three: HOSPADM, DBP, and SBP (the minimum AVAL of those three that have CNSR=0 will be the earliest relative day of the three types of events); a subject who meets one of these three conditions has CNSR=0 for PARAMCD=“HYPEREVT” to indicate the subject had an event. If a subject does not meet one of the three conditions (i.e., all three records have CNSR=1), then the subject is censored; that is, AVAL for PARAMCD=“HYPEREVT” is derived as the relative day that the subject exited the study and CNSR=1 is used to indicate the subject is censored. The analysis will focus on HYPEREVT, but HOSPADM, DBP and SBP are included to support traceability, and also to enable future analysis of the sub-events should it be desired. In this example, the SRC* variables were populated for the derived event (PARAMCD=“HYPEREVT”), as described in Section 3.3.9.

The main advantage of this structure is that it can handle sub-event input rows from many domains in only 3 standard supportive columns (i.e., SRCDOM, SRCVAR, and SRCSEQ). This approach is preferred because it is standardized, scalable, and supports analysis of sub-events.

Table 4.4.1.4 Example 1: ADaM Dataset

Row	USUBJID	PARAM	PARAMCD	AVAL	CNSR	EVNTDESC	SRCDOM	SRCVAR	SRCSEQ
1	2010	Time to First Hospital Admission (day)	HOSPADM	9	0	FIRST HOSPITAL ADMISSION	HO	HOSTDY	99
2	2010	Time to First DBP>90 (day)	DBP	15	0	FIRST DBP>90	VS	VSDY	208
3	2010	Time to First SBP>140 (day)	SBP	22	1	COMPLETED THE STUDY	DS	DSSTDY	301
4	2010	Time to Hypertension Event (day)	HYPEREVT	9	0	HYPERTEN. EVENT	DS	DSSTDY	99
5	3082	Time to First Hospital Admission (day)	HOSPADM	10	1	COMPLETED THE STUDY	DS	DSSTDY	130
6	3082	Time to First DBP>90 (day)	DBP	10	1	COMPLETED THE STUDY	DS	DSSTDY	130
7	3082	Time to First SBP>140 (day)	SBP	10	1	COMPLETED THE STUDY	DS	DSSTDY	130
8	3082	Time to Hypertension Event (day)	HYPEREVT	10	1	COMPLETED THE STUDY	DS	DSSTDY	130

Example 2

The analysis parameter is glomerular filtration rate (GFR) estimated from serum creatinine using the MDRD Equation (Modification of Diet in Renal Disease Study Group). The analysis value for this parameter is derived from plasma creatinine, BUN, and albumin values from the LB dataset, as well as age, race, and sex.

Table 4.4.1.3 Example 2: Data as Found in SDTM LB Dataset

Row	USUBJID	VISITNUM	LBSEQ	LBTEST	LBTESTCD	LBSTRESN	LBSTRESU
1	3000	3	98	Creatinine	CREAT	78.2	micromol/L
2	3000	3	115	Blood Urea Nitrogen	BUN	9.1	mmol/L
3	3000	3	120	Albumin	ALB	40	g/L

Additional rows are not created for the input data age, race, and sex, as they are covariates in the ADaM dataset. The analysis records in Table 4.4.1.4 are identified by PARAMCD=MDRD_GFR, the parameter code for PARAM = Glomerular Filtration Rate (GFR) (ml/min/1.73m**2). In this example, because all of the data come from a single source dataset (in this example, the LB dataset), the LBSEQ variable is retained for traceability, though it would also be valid to instead use the ADaM SRC variables.

Table 4.4.1.4 Example 2: ADaM Dataset

Row	USUBJID	AGE	SEX	RACE	PARAM	PARAMCD	VISITNUM	AVAL	LBSEQ
1	3000	52	F	ASIAN	Creatinine	CREAT	3	78.2	98
2	3000	52	F	ASIAN	Blood Urea Nitrogen	BUN	3	9.1	115
3	3000	52	F	ASIAN	Albumin	ALB	3	40	120
4	3000	52	F	ASIAN	Glomerular Filtration Rate (GFR) (ml/min/1.73m**2)	MDRD_GFR	3	76.77	

Example 3

An ADaM dataset is created to contain the time to pain relief (ADTTPRLF), based on data in another ADaM dataset (ADPAIN). Pain relief is defined as a reduction in pain from moderate or severe at baseline (i.e., pain severity of at least 2) to mild or no pain (i.e., pain severity of no more than 1), with no use of rescue medication from baseline to that timepoint (i.e., RESCUEFL null at that timepoint and for the subject’s records prior to that timepoint). Subjects who do not achieve pain relief are censored at their last pain severity assessment. Missing data are imputed in ADPAIN using LOCF. Because the source dataset is an ADaM dataset, the SRCDOM, SRCVAR, and SRCSEQ variables are used for datapoint traceability.

Table 4.4.1.5 Example 3: Data as Found in ADPAIN (Source ADaM Dataset)

Row	USUBJID	ASEQ	PARAM	PARAMCD	ATPT	ATPTN	AVAL	AVALC	BASEC	CRIT1	CRIT1FL	DTYPE	RESCUEFL	QSSEQ
1	101-001	1	Pain Severity	SEVERITY	BSLN	0	3	Severe	Severe	Pain relief	N			100
2	101-001	2	Pain Severity	SEVERITY	30 MIN	30	2	Moderate	Severe	Pain relief	N			101
3	101-001	3	Pain Severity	SEVERITY	1 HR	60	1	Mild	Severe	Pain relief	Y			102
4	101-001	4	Pain Severity	SEVERITY	90 MIN	90	1	Mild	Severe	Pain relief	Y			103
5	101-001	5	Pain Severity	SEVERITY	2 HR	120	0	None	Severe	Pain relief	Y			104
6	101-002	1	Pain Severity	SEVERITY	BSLN	0	3	Severe	Severe	Pain relief	N			111
7	101-002	2	Pain Severity	SEVERITY	30 MIN	30	3	Severe	Severe	Pain relief	N			112
8	101-002	3	Pain Severity	SEVERITY	1 HR	60	2	Moderate	Severe	Pain relief	N		Y	113
9	101-002	4	Pain Severity	SEVERITY	90 MIN	90	2	Moderate	Severe	Pain relief	N		Y	114
10	101-002	5	Pain Severity	SEVERITY	2 HR	120	1	Mild	Severe	Pain relief	N		Y	115
11	101-003	1	Pain Severity	SEVERITY	BSLN	0	3	Severe	Severe	Pain relief	N			276
12	101-003	2	Pain Severity	SEVERITY	30 MIN	30	2	Moderate	Severe	Pain relief	N			277
13	101-003	3	Pain Severity	SEVERITY	1 HR	60	1	Mild	Severe	Pain relief	Y			278
14	101-003	4	Pain Severity	SEVERITY	90 MIN	90	1	Mild	Severe	Pain relief	Y	LOCF		278
15	101-003	5	Pain Severity	SEVERITY	2 HR	120	1	Mild	Severe	Pain relief	Y	LOCF		278

Table 4.4.1.6 Example 3: ADaM Dataset ADTTPRLF

Row	USUBJID	PARAM	PARAMCD	AVAL	CNSR	SRCDOM	SRCVAR	SRCSEQ
1	101-001	Time to First Pain Relief (minutes)	TTPRLF	60	0	ADPAIN	ATPTN	3
2	101-002	Time to First Pain Relief (minutes)	TTPRLF	120	1	ADPAIN	ATPTN	5
3	101-003	Time to First Pain Relief (minutes)	TTPRLF	60	0	ADPAIN	ATPTN	3

Approaches Considered and Not Adopted

A second approach that was considered was to describe the derivation algorithms in metadata and include the input data as columns in the ADaM dataset. Pointer columns would be added to indicate the source of the input data – variable name and sequence number. This option would allow all pertinent input data to be retained on the relevant analyzed row (i.e., all sub-events would be shown on the same row as a compound event), which might help simplify verification of the calculation of the analysis parameter. However, this approach would clearly increase the number of columns in the ADaM dataset and would require naming the variables in a clear and concise manner. The approach also assumes that the only data to be retained are the original input values. Another drawback of this approach is that if there were a need in the future to analyze the sub-events, sub-event parameters would have to be added to have an ADaM-conformant structure supporting the analysis of sub-events. For these reasons, this approach was not chosen.

A third approach that was considered was to describe the derivation algorithms in metadata and include no input data or identification of the input data in the ADaM dataset. The advantage to this approach would be simplification of the ADaM dataset. However, due to the simplified structure, there would be a loss of traceability between the data collected in the study (i.e., SDTM dataset) and the data analyzed (i.e., ADaM dataset). Unless the derivation algorithms described in the metadata are straightforward, verification of the analysis data computation could be very challenging or even impossible. This approach should not be used.

4.5 Identification of Rows Used for Analysis

This section addresses how to identify the rows of an ADaM dataset that are used for analysis. The four specific issues addressed include: 1) identification of the rows used in a last observation carried forward (LOCF) analysis; 2) identification of the row containing the baseline value; 3) identification of post-baseline conceptual timepoint rows, such as endpoint, minimum, maximum, or average; and 4) identification of specific rows used in an analysis.

4.5.1 Identification of Rows Used in a Timepoint Imputation Analysis

This section considers the issue of how to identify rows used in a timepoint-related imputation analysis as well as how to represent data imputed for missing timepoints in an ADaM dataset. Last observation carried forward (LOCF) is one of the most commonly used timepoint-related imputation analyses, and is therefore specifically mentioned. However, the methodology is general and is not restricted to LOCF analysis. Worst observation carried forward (WOCF) analysis is also mentioned to emphasize the generalizability.

4.5.1.1 ADaM Methodology and Examples

When an analysis timepoint is missing, the ADaM methodology is to create a new row in the ADaM dataset to represent the missing timepoint and identify these imputed rows by populating the derivation type variable DTYPE.

For example, when an LOCF/WOCF analysis is being performed, create LOCF/WOCF rows when the LOCF/WOCF analysis timepoints are missing, and identify these imputed rows by populating the derivation type variable DTYPE with values LOCF or WOCF. All of the original rows would have null values in DTYPE. It would be very simple to select the appropriate rows for analysis by selecting DTYPE = null for Data as Observed (DAO) analysis, DTYPE = null or LOCF for LOCF analysis, and DTYPE = null or WOCF for WOCF analysis. This approach would require understanding and communication that if the DTYPE flag were not referenced correctly, the analysis would default to using all rows, including the DAO rows, plus the rows derived by LOCF and WOCF. To perform a correct DAO analysis, one would need to explicitly select DTYPE = null.

Example 1
Identification of rows used in a LOCF analysis.

In the example below (Table 4.5.1.1), some subjects have complete data and others have rows imputed by one method (LOCF). Subjects with no missing data have the observed number of rows with all DTYPE values blank. Subject 1001 has complete data. DTYPE is blank for all rows indicating they are not imputed. AVISIT matches VISIT (from SDTM) in this example. AVISIT does not always match VISIT from SDTM even in scenarios where there is no missing data. Subject 1002 is missing the Week 2 assessment. Week 2 is imputed using the LOCF method. AVISIT=Week 2 but VISIT=Week 1 so one can see where the imputed value came from in the original data. Subject 1003 is missing Week 2 and 3 data. A Data as Observed (DAO) analysis can be performed by selecting only those rows where DTYPE is null. For a LOCF analysis, all rows (DTYPE=null or DTYPE="LOCF") should be used.

Table 4.5.1.1 Example 1: ADaM Dataset with Identification of Rows Used in a LOCF Analysis

Row	USUBJID	VISIT	AVISIT	ADY	PARAM	AVAL	DTYPE	VSSEQ
1	1001	Baseline	Baseline	-4	SUPINE SYSBP (mm Hg)	145		171
2	1001	Week 1	Week 1	3	SUPINE SYSBP (mm Hg)	130		191
3	1001	Week 2	Week 2	9	SUPINE SYSBP (mm Hg)	133		201
4	1001	Week 3	Week 3	20	SUPINE SYSBP (mm Hg)	125		211

Row	USUBJID	VISIT	AVISIT	ADY	PARAM	AVAL	DTYPE	VSSEQ
5	1002	Baseline	Baseline	-1	SUPINE SYSBP (mm Hg)	145		50
6	1002	Week 1	Week 1	7	SUPINE SYSBP (mm Hg)	130		60
7	**1002**	**Week 1**	**Week 2**	**7**	**SUPINE SYSBP (mm Hg)**	**130**	**LOCF**	**60**
8	1002	Week 3	Week 3	22	SUPINE SYSBP (mm Hg)	135		70
9	1003	Baseline	Baseline	1	SUPINE SYSBP (mm Hg)	150		203
10	1003	Week 1	Week 1	8	SUPINE SYSBP (mm Hg)	140		213
11	**1003**	**Week 1**	**Week 2**	**8**	**SUPINE SYSBP (mm Hg)**	**140**	**LOCF**	**213**
12	**1003**	**Week 1**	**Week 3**	**8**	**SUPINE SYSBP (mm Hg)**	**140**	**LOCF**	**213**

Example 2
Identification of rows used in both LOCF and WOCF analyses.

This set of rows (Table 4.5.1.2) shows a situation where there is more than one imputation method used. In this case, additional rows are generated for each type of imputation. A DAO analysis can be performed by selecting only those rows where DTYPE is null. For LOCF analysis, all rows with DTYPE=null or DTYPE="LOCF" should be used. For WOCF analysis, all rows with DTYPE=null or DTYPE="WOCF" should be used.

Table 4.5.1.2 Example 2: ADaM Dataset with Identification of Rows Used in Both LOCF and WOCF Analyses

Row	USUBJID	VISIT	AVISIT	ADY	PARAM	AVAL	DTYPE	VSSEQ
1	1002	Baseline	Baseline	-4	SUPINE SYSBP (mm Hg)	145		77
2	1002	Week 1	Week 1	3	SUPINE SYSBP (mm Hg)	130		78
3	1002	Week 2	Week 2	9	SUPINE SYSBP (mm Hg)	138		79
4	1002	Week 3	Week 3	18	SUPINE SYSBP (mm Hg)	135		80
5	**1002**	**Week 3**	**Week 4**	**18**	**SUPINE SYSBP (mm Hg)**	**135**	**LOCF**	**80**
6	**1002**	**Week 2**	**Week 4**	**9**	**SUPINE SYSBP (mm Hg)**	**138**	**WOCF**	**79**
7	1002	Week 5	Week 5	33	SUPINE SYSBP (mm Hg)	130		81
8	1003	Baseline	Baseline	-1	SUPINE SYSBP (mm Hg)	145		122
9	1003	Week 1	Week 1	7	SUPINE SYSBP (mm Hg)	140		123
10	1003	Week 2	Week 2	15	SUPINE SYSBP (mm Hg)	138		124
11	**1003**	**Week 2**	**Week 3**	**15**	**SUPINE SYSBP (mm Hg)**	**138**	**LOCF**	**124**
12	**1003**	**Week 2**	**Week 4**	**15**	**SUPINE SYSBP (mm Hg)**	**138**	**LOCF**	**124**
13	**1003**	**Week 2**	**Week 5**	**15**	**SUPINE SYSBP (mm Hg)**	**138**	**LOCF**	**124**
14	**1003**	**Week 1**	**Week 3**	**7**	**SUPINE SYSBP (mm Hg)**	**140**	**WOCF**	**123**
15	**1003**	**Week 1**	**Week 4**	**7**	**SUPINE SYSBP (mm Hg)**	**140**	**WOCF**	**123**
16	**1003**	**Week 1**	**Week 5**	**7**	**SUPINE SYSBP (mm Hg)**	**140**	**WOCF**	**123**

Approaches Considered and Not Adopted
Another approach considered is to create a complete separate set of rows for each analysis type (or a separate dataset), indicating the various analysis types by assigning unique values of the analysis timepoint description AVISIT, e.g., "Week 4", "Week 4 (LOCF)" and "Week 4 (WOCF)". This approach would make it more foolproof to perform the DAO, LOCF, and WOCF analysis in one step by referencing only AVISIT. However, because so many rows would be duplicated, a very large dataset is one of the major disadvantages for this approach. In addition, this approach might be less tool-friendly, in that one might need to parse AVISIT searching for a key substring, e.g., "(LOCF)". This approach should not be used.

A third approach considered is to create a flag (LOCFFL/LOCFFN) to indicate when a row is created by virtue of last observation carried forward; and similarly for WOCF. This is similar to the specified ADaM methodology, except that a separate flag is created for each derivation type, rather than indicating row derivation type in one column DTYPE. This approach might result in fewer rows than the recommended approach (for example if the WOCF row is the same as the LOCF row). In other respects, this approach shares the advantages and disadvantages of the recommended approach. This approach of creating separate flags for each derivation type is not recommended.

4.5.2 Identification of Baseline Rows

Many statistical analyses require the identification of a baseline value. This section describes how a record used as a baseline is identified.

4.5.2.1 ADaM Methodology and Examples

The ADaM methodology is to create a baseline flag column to indicate the row used as baseline (the row whose value of AVAL is used to populate the BASE variable). This method does not require duplication of rows in the event that the baseline row is not derived.

Though a baseline row flag variable ABLFL is created and used to identify the row that is the baseline row, this does not prohibit also providing a row with a unique value of AVISIT, e.g., “Baseline”, designating the baseline row used for analysis, even if redundant with another row. For more complicated baseline definitions (functions of multiple rows), a derived baseline row would have to be created as described in Rule 3. This methodology requires that clear metadata be provided for the baseline row variable so that the value can be reproduced accurately.

Example 1
Identification of baseline rows - using screening visit to impute a baseline row.

This example (Table 4.5.2.1) illustrates the use of a baseline flag variable ABLFL. It also illustrates the inclusion of an additional row for a baseline analysis timepoint (row 6). In this example, a unique value of AVISIT has been defined for the baseline record used for analysis. Subject 1001 had complete data. There was no record that qualified as a baseline value for Subject 1002 in the source data. A derived baseline record (AVISIT=“Baseline”) is added with DTYPE=“LVPD” (Last Value Prior to Dosing) to indicate that the record is imputed to be used as baseline.

Table 4.5.2.1 Example 1: ADaM Dataset with Identification of Baseline Rows When Imputation is Used

Row	USUBJID	VISIT	AVISIT	ADY	ABLFL	PARAM	AVAL	BASE	DTYPE	VSSEQ
1	1001	Screening	Screening	-12		SUPINE SYSBP (mm Hg)	144			1
2	1001	Baseline	Baseline	1	Y	SUPINE SYSBP (mm Hg)	145			2
3	1001	Week 1	Week 1	6		SUPINE SYSBP (mm Hg)	130	145		3
4	1001	Week 2	Week 2	12		SUPINE SYSBP (mm Hg)	133	145		4
5	1002	Screening	Screening	-14		SUPINE SYSBP (mm Hg)	144			1
6	**1002**	**Screening**	**Baseline**	**-14**	**Y**	**SUPINE SYSBP (mm Hg)**	**144**		**LVPD**	**1**
7	1002	Week 1	Week 1	8		SUPINE SYSBP (mm Hg)	130	144		2
8	1002	Week 2	Week 2	14		SUPINE SYSBP (mm Hg)	133	144		3

Example 2
Identification of baseline rows - using an average of multiple visits to derive a baseline row.

This example (Table 4.5.2.2) illustrates the use of a baseline flag variable ABLFL to identify the record used as baseline for analysis in a scenario where the baseline value is based on the average of the non-missing values collected prior to dosing. Row 3 is a derived "Baseline" record using the average of the values of row 1 and row 2. DTYPE = "AVERAGE" to indicate that row 3 is derived. The Baseline flag (ABLFL="Y") indicates that AVAL from row 3 is used to populate the BASE (Baseline) column. VISIT (from SDTM) is left blank on row 3 since AVAL on that record is not merely a copy of AVAL on another record.

Table 4.5.2.2 Example 2: ADaM Dataset with Identification of Baseline Rows When Baseline is an Average

Row	USUBJID	VISIT	AVISIT	ADY	ABLFL	PARAM	AVAL	BASE	DTYPE
1	1001	Screening	Screening	-12		SUPINE SYSBP (mm Hg)	144	144.5	
2	1001	Baseline	Baseline	1		SUPINE SYSBP (mm Hg)	145	144.5	
3	1001		**Baseline**		**Y**	SUPINE SYSBP (mm Hg)	144.5	144.5	**AVERAGE**
4	1001	Week 1	Week 1	12		SUPINE SYSBP (mm Hg)	130	144.5	
5	1001	Week 2	Week 2	-14		SUPINE SYSBP (mm Hg)	133	144.5	

Example 3

Identification of baseline rows - using an average of multiple visits to derive a baseline row.

This example (Table 4.5.2.3) is the same as Example 2 except that the analysis timepoint description "Screening/Baseline Combination" helps differentiate the derived average baseline record from an existing observed record whose timepoint description is "Baseline." This was helpful in analysis and reporting because it was desired to summarize all scheduled visits in addition to the average baseline visit. The analysis was straightforward using the distinct descriptions of AVISIT. The choice of AVISIT values is up to the producer.

Table 4.5.2.3 Example 3: ADaM Dataset with Identification of Baseline Rows, Including Description in Analysis Timepoint Variable

Row	USUBJID	VISIT	AVISIT	ADY	ABLFL	PARAM	AVAL	BASE	DTYPE
1	1001	Screening	Screening	-12		SUPINE SYSBP (mm Hg)	144	144.5	
2	1001	Baseline	Baseline	1		SUPINE SYSBP (mm Hg)	145	144.5	
3	1001		**Screening/Baseline Combination**		**Y**	SUPINE SYSBP (mm Hg)	144.5	144.5	**AVERAGE**
4	1001	Week 1	Week 1	12		SUPINE SYSBP (mm Hg)	130	144.5	
5	1001	Week 2	Week 2	-14		SUPINE SYSBP (mm Hg)	133	144.5	

4.5.3 Identification of Post-Baseline Conceptual Timepoint Rows

When analysis involves cross-timepoint derivations such as endpoint, minimum, maximum and average post-baseline, questions such as "Should distinct rows with unique value of AVISIT always be created even if redundant with an observed value record, or should these rows just be flagged?" should be considered. There are two approaches presented in this section.

4.5.3.1 ADaM Methodology and Examples

The ADaM methodology is to create a new row with a unique value of AVISIT in cases where analysis is based on AVISIT. The advantage of this approach is that it is simple and analysis friendly. It is recognized that such new rows might be redundant with observed rows for some kinds of conceptual timepoint definitions.

Always creating a row with a unique value of AVISIT designating the row used for analysis (e.g., “Endpoint”, “Post-Baseline Minimum”, “Post-Baseline Maximum”) has the advantage that once the AVISIT values are understood, producers, consumers, and software can rely on these values of AVISIT. This approach represents the general case since any such cross-timepoint derivation can be represented in a new row with a unique AVISIT description. The disadvantage is that the dataset would contain more rows, and conventions would have to be communicated and understood.

In cases where analysis is not based on AVISIT, then either solution is valid. It is recognized that in cases where the AVISIT values are not defined in the analysis documentation, then adding a flag may be more appropriate. Which methodology is appropriate for situations where an “analysis visit” value is not defined can be driven by how the analysis will be performed. In cases where only a subset of data is analyzed (i.e., only on treatment minimum values), then flagging the values that qualify for analysis might be a better choice than creating an additional row to contain the minimum value. However, where the subset of data is analyzed within the context of a greater pool of data, then creating an additional row to contain the minimum value would help facilitate analysis-ready usage and review.

Example 1
Identification of endpoint rows.

This example (Table 4.5.3.1) shows the creation of an added row with a unique value of AVISIT designating the Endpoint record used for analysis. Subject 1001 discontinued at Week 2, and a derived Endpoint record (AVISIT=“Endpoint”) is added using the Week 2 visit. DTYPE=“LOV” (Last Observed Value) indicates how the AVISIT=“Endpoint” record is populated. Subject 1002 did not have any post-baseline visits, and therefore has no Endpoint record.

Table 4.5.3.1 Example 1: ADaM Dataset with Identification of Endpoint Rows

Row	USUBJID	VISIT	AVISIT	ADY	PARAM	AVAL	DTYPE
1	1001	Screening	Screening	-12	SUPINE SYSBP (mm Hg)	144	
2	1001	Baseline	Baseline	1	SUPINE SYSBP (mm Hg)	145	
3	1001	Week 1	Week 1	6	SUPINE SYSBP (mm Hg)	130	
4	1001	Week 2	Week 2	12	SUPINE SYSBP (mm Hg)	133	
5	**1001**	**Week 2**	**Endpoint**	**12**	**SUPINE SYSBP (mm Hg)**	**133**	**LOV**
6	1002	Screening	Screening	-14	SUPINE SYSBP (mm Hg)	144	
7	1002	Baseline	Baseline	-1	SUPINE SYSBP (mm Hg)	144	

Example 2
Identification of endpoint and post-baseline minimum, maximum, and average rows.

This example (Table 4.5.3.2) shows the creation of rows with unique values of AVISIT designating the Endpoint record, and the Post-Baseline Minimum, Maximum, and Average rows. Subject 1001 had minimum post-baseline result at Week 1, maximum post-baseline result at Week 2, and the average post-baseline result was based on the average of Week 1 and Week 2. This subject discontinued at Week 2. A derived Endpoint record (AVISIT=“Endpoint”) is added using the Week 2 visit. DTYPE=“LOV” (last observed value) indicates that the AVISIT=“Endpoint” record is a derived record. Subject 1002 did not have any post-baseline visit. Therefore, the Post-Baseline Minimum, Post-Baseline Maximum, Post-Baseline Average, and Endpoint rows could not be derived for that subject.

Table 4.5.3.2 Example 2: ADaM Dataset with Identification of Endpoint and Post-Baseline Minimum, Maximum, and Average Rows

Row	USUBJID	VISIT	AVISIT	ADY	PARAM	AVAL	DTYPE	VSSEQ
1	1001	Screening	Screening	-12	SUPINE SYSBP (mm Hg)	144		1
2	1001	Baseline	Baseline	1	SUPINE SYSBP (mm Hg)	145		2
3	1001	Week 1	Week 1	6	SUPINE SYSBP (mm Hg)	130		3
4	1001	Week 2	Week 2	12	SUPINE SYSBP (mm Hg)	133		4
5	**1001**	**Week 1**	**Post-Baseline Minimum**	**6**	**SUPINE SYSBP (mm Hg)**	**130**	**Minimum**	**3**
6	**1001**	**Week 2**	**Post-Baseline Maximum**	**12**	**SUPINE SYSBP (mm Hg)**	**133**	**Maximum**	**4**
7	**1001**		**Post-Baseline Average**		**SUPINE SYSBP (mm Hg)**	**131.5**	**Average**	
8	**1001**	**Week 2**	**Endpoint**	**12**	**SUPINE SYSBP (mm Hg)**	**133**	**LOV**	**4**
9	1002	Screening	Screening	-14	SUPINE SYSBP (mm Hg)	144		22
10	1002	Baseline	Baseline	-1	SUPINE SYSBP (mm Hg)	144		23

Example 3

Identification of post-baseline minimum and maximum rows.

This example (Table 4.5.3.3) shows the identification of the Post-Baseline Minimum and Maximum rows. Subject 1001 had minimum post-baseline result at Week 1 (identified with ANL01FL=Y) and maximum post-baseline result at Week 2 (identified with ANL02FL=Y). Subject 1002 did not have any post-baseline visit. Therefore, the Post-Baseline Minimum and Post-Baseline Maximum could not be identified for that subject.

Table 4.5.3.3 Example 3: ADaM Dataset with Identification of Post-Baseline Minimum and Maximum Rows

Row	USUBJID	VISIT	AVISIT	ADY	PARAM	AVAL	ANL01FL	ANL02FL
1	1001	Screening	Screening	-12	SUPINE SYSBP (mm Hg)	144		
2	1001	Baseline	Baseline	1	SUPINE SYSBP (mm Hg)	145		
3	1001	Week 1	Week 1	6	SUPINE SYSBP (mm Hg)	130	Y	
4	1001	Week 2	Week 2	12	SUPINE SYSBP (mm Hg)	133		Y
9	1002	Screening	Screening	-14	SUPINE SYSBP (mm Hg)	144		
10	1002	Baseline	Baseline	-1	SUPINE SYSBP (mm Hg)	144		

4.5.4 Identification of Rows Used for Analysis – General Case

It is important to identify the rows used in or excluded from analysis. Should rows used in the analysis be identified via flags or by unique values of analysis timepoint window description AVISIT?

4.5.4.1 ADaM Methodology and Examples

The ADaM methodology is to use an analysis flag (ANL*zz*FL) to indicate the rows that fulfill specific requirements for one or more analyses. For example, ANL*zz*FL=Y indicates rows meeting the requirements for analysis and is blank (null) in other rows such as a duplicate row that was not the one selected for analysis, or pre-specified post-study timepoints not included in the analysis. This allows multiple rows within a parameter with the same value of AVISIT. However, it also requires flags to be added to the dataset to be used in selecting appropriate rows for analysis. Understanding of the flags is required for correct analysis results to be generated. In addition to ANL*zz*FL, additional flags might also be required, such as row-based population flags, e.g., ITTRFL and PPROTRFL.

Please note that there can be multiple ANL*zz*FL variables. In this case it will be imperative to have clear and robust metadata to indicate the basis for creation and populating of each ANL*zz*FL variable.

Example 1
Identification of rows used for analysis – multiple visits that fall within a visit window.

This example (Table 4.5.4.1) illustrates the use of the analysis flag variable ANL*zz*FL to indicate the rows that were chosen for analysis from among the multiple visits that fall within the analysis timepoint windows of "Baseline" and "Week 2". Subject 1001 had two observed Baseline and Week 2 analysis timepoints according to analysis window definitions. The one that is used in analysis is flagged with ANL01FL=Y. This approach is used because all original visits (rows) are included in the dataset, and those selected for analysis must be identified. For traceability reasons, the AWTARGET and AWTDIFF columns are included in order to indicate more clearly how the analyzed rows were selected from among the candidate rows within each analysis window. In this example, the record that falls closest to the scheduled visit day is the one that will be analyzed.

Table 4.5.4.1 Example 1: ADaM Dataset with Identification of Rows Used for Analysis When Multiple Visits Fall Within a Visit Window

Row	USUBJID	VISIT	AVISIT	ADY	PARAM	AVAL	DTYPE	ANL01FL	AWTARGET	AWTDIFF
1	1001	Screening	Baseline	-5	SUPINE SYSBP (mm Hg)	144			1	5
2	1001	Baseline	Baseline	1	SUPINE SYSBP (mm Hg)	145		Y	1	0
3	1001	Week 1	Week 1	7	SUPINE SYSBP (mm Hg)	130		Y	7	0
4	1001	Week 2	**Week 2**	12	SUPINE SYSBP (mm Hg)	133		Y	14	2
5	1001	Week 3	**Week 2**	17	SUPINE SYSBP (mm Hg)	125			14	3
6	1001	Week 4	Week 4	30	SUPINE SYSBP (mm Hg)	128		Y	28	2

Example 2
Identification of rows used for analysis – visit falls outside of a target window.

In this example (Table 4.5.4.2), the Week 3 visit for subject 1001 was outside the day window of analysis Week 3, so "Post-Study" was assigned to AVISIT. This visit as well as the first baseline visit were excluded from the analysis per the Statistical Analysis Plan. The "Worst Post-Baseline" analysis timepoint (Row 6) was imputed by worst observed case (DTYPE=WOC). The "Endpoint" row was derived using the "Week 2" visit, since it was the last available eligible observation based on the Statistical Analysis Plan. Both of the derived rows are flagged with ANL01FL=Y since they were rows selected for analysis.

Table 4.5.4.2 Example 2: ADaM Dataset with Identification of Rows Used for Analysis When Visit Falls Outside of a Target Window

Row	USUBJID	VISIT	AVISIT	ADY	VISITDY	PARAM	AVAL	DTYPE	ANL01FL
1	1001	Screening	Baseline	-5	1	SUPINE SYSBP (mm Hg)	144		
2	1001	Baseline	Baseline	1	1	SUPINE SYSBP (mm Hg)	145		Y
3	1001	Week 1	Week 1	7	7	SUPINE SYSBP (mm Hg)	150		Y
4	1001	Week 2	Week 2	12	14	SUPINE SYSBP (mm Hg)	133		Y
5	1001	Week 3	**Post-Study**	**40**	**21**	SUPINE SYSBP (mm Hg)	140		
6	**1001**	**Week 1**	**Worst Post-Baseline**	**7**	**7**	**SUPINE SYSBP (mm Hg)**	**150**	**WOC**	**Y**
7	**1001**	**Week 2**	**Endpoint**	**12**	**14**	**SUPINE SYSBP (mm Hg)**	**133**	**LOV**	**Y**

Example 3

Identification of rows used for analysis – a visit not flagged for the analysis is used to create imputed LOCF rows.

This example (Table 4.5.4.3) illustrates a scenario where two visits occur within a window (Week 2). The first record (on row 4) is analyzed as is (it is the record chosen to represent analysis timepoint Week 2 based on an algorithm defined in the SAP and referred to in the metadata of ANL01FL). The second Week 2 timepoint record (on row 5) is the basis for the LOCF derivation of analysis timepoints Week 3, 4 and 5 (rows 6, 7, and 8). In the LOCF analysis, Week 2 is based on the observed data on row 4, and Weeks 3, 4, and 5 are imputed using the last available observation on row 5.

Table 4.5.4.3 Example 3: ADaM Dataset with a Value that is Carried Forward but Not Included in the Analysis

Row	USUBJID	VISIT	AVISIT	ADY	PARAM	AVAL	DTYPE	ANL01FL
1	1001	Screening	Baseline	-5	SUPINE SYSBP (mm Hg)	144		
2	1001	Baseline	Baseline	1	SUPINE SYSBP (mm Hg)	145		Y
3	1001	Week 1	Week 1	7	SUPINE SYSBP (mm Hg)	130		Y
4	**1001**	**Week 2**	**Week 2**	**12**	**SUPINE SYSBP (mm Hg)**	**133**		**Y**
5	**1001**	**Week 3**	**Week 2**	**17**	**SUPINE SYSBP (mm Hg)**	**125**		
6	1001	Week 3	Week 3	17	SUPINE SYSBP (mm Hg)	125	LOCF	Y
7	1001	Week 3	Week 4	17	SUPINE SYSBP (mm Hg)	125	LOCF	Y
8	1001	Week 3	Week 5	17	SUPINE SYSBP (mm Hg)	125	LOCF	Y

Approaches Considered and Not Adopted

Another option considered was to create unique values of the timepoint window description AVISIT. For example, add an asterisk to the end of AVISIT such as "Week 2 *" if not analyzed. This approach might be less confusing because one would not need to be aware of a flag. The disadvantage is that one would need to have a convention for AVISIT values, and tools would need to parse values of AVISIT for correct results to be generated. For these reasons, this approach was not chosen.

4.6 Identification of Population-Specific Analyzed Rows

It is not uncommon in the statistical analysis of clinical trials to conduct analyses based on multiple populations of interest. The population of interest can be defined either at the subject level, the row (measurement) level, or both. For example, when defining an analysis population, a subject may be included in one analysis population such as Intent-to-Treat but may be excluded from another analysis population such as Per-Protocol. Analysis populations may also be defined using characteristics of individual measurements. For example, a measurement that was assessed outside of a pre-specified time window for a particular visit may not be included in a per-protocol visit-level population. In this section, it is assumed that the definition of a row-level analysis population is dependent on the definition of the subject-level population. In other words, if a subject is excluded from the subject-level Per-Protocol population, then none of that subject's rows would be candidates for inclusion within the row-level Per-Protocol population. Given the variety of possible population definitions, the same row in an analysis dataset could be included in one analysis and excluded from another, depending on characteristics of the subject as a whole and the characteristics of the individual measurement. Therefore, the issue becomes how best to select rows for each analysis.

4.6.1 ADaM Methodology and Examples

The ADaM methodology for this analysis issue is to create one ADaM dataset that can be used to perform multiple analyses using population flag variables to identify rows that are used for each type of analysis. An advantage of this approach is that the one ADaM dataset can be used for multiple analyses. Flag variables obviate the need to replicate rows for each type of analysis. This approach promotes efficiency in the operational aspects of electronic submissions, clarity of analyses, and ease in comparing selected values for each population. This approach does, however, require that clear metadata be provided for the flag variables so that each specific analysis can be reproduced accurately. Below are several examples of the use of population flag variables to identify rows used for different analyses.

Example 1
Use of subject-level flag variables (ITTFL and PPROTFL) and row-level flag variables (ANL01FL and PPROTRFL).

In some statistical analyses, even if a subject is included in the Per-Protocol population, some or all data for that subject in a particular dataset may not be appropriate for a per-protocol analysis. Consider a situation in HIV studies where a Per-Protocol analysis excludes all data after permanent discontinuation of study medication or addition of other antiretroviral therapy. An example of an ADaM dataset to support this type of analysis is illustrated in Table 4.6.1.1. This ADaM dataset (Table 4.6.1.1) can be used to repeat analyses based on multiple populations of interest either at the subject level or at the row (measurement) level.

ITTFL and PPROTFL are subject-level analysis population flags. If a subject is in the Intent-to-Treat population, then the column ITTFL will have the value of “Y” (“N” if not). In Table 4.6.1.1, subjects 1001, 1002, and 1003 are in the Intent-to-Treat population. Similarly, if a subject is in the Per-Protocol population, the column PPROTFL will have the value of “Y” (“N” if not). Subjects 1001 and 1003 in Table 4.6.1.1 are in the Per-Protocol population while subject 1002 with PPROTFL=N is excluded from any Per-Protocol analysis. These indicator variables are used to identify individual subjects that belong to each subject-level population.

In contrast to the subject-level population flags, the column PPROTRFL is the per-protocol analysis flag at the row level. As illustrated in Table 4.6.1.1, if a row is a candidate for the Per-Protocol analysis, the variable PPROTRFL is set to “Y”, it is null if the row does not fulfill the criteria for this analysis. In the example, subjects 1001 and 1002 continue with study medication after Week 2; the last dose of study medication for subject 1003 is at Week 1. In Table 4.6.1.1, all three rows for subject 1002 and two of four rows for subject 1003 are not row-level Per-Protocol data and would not be selected for a Per-Protocol analysis when we apply the subset condition: PPROTRFL=“Y”. PPROTRFL is null on the last two rows for subject 1003 and will be excluded from any row-level Per-Protocol data analysis as they occur after the subject discontinued study medication.

Not all rows in Table 4.6.1.1 are included for analysis purposes. In this example, the analysis flag ANL01FL is null for one row (USUBJID=1003, VISIT=Week 1, AVISIT=Week 1, AVAL=999) because its value was replaced by the retest result in the next row (USUBJID=1003, VISIT=Retest, AVISIT=Week 1, AVAL=49). The analysis flag for the Retest record is Y.

Table 4.6.1.1 Example 1: ADaM Dataset with Subject-Level and Row-Level Indicator Variables

Row	USUBJID	ITTFL	PPROTFL	VISIT	AVISIT	PARAMCD	AVAL	ANL01FL	PPROTRFL
1	1001	Y	Y	Week 0	Week 0	TEST1	500	Y	Y
2	1001	Y	Y	Week 1	Week 1	TEST1	400	Y	Y
3	1001	Y	Y	Week 2	Week 2	TEST1	600	Y	Y
4	1002	Y	N	Week 0	Week 0	TEST1	500	Y	
5	1002	Y	N	Week 2	Week 1	TEST1	48	Y	
6	1002	Y	N	Week 2	Week 2	TEST1	46	Y	
7	1003	Y	Y	Week 0	Week 0	TEST1	999	Y	Y
8	1003	Y	Y	Week 1	Week 1	TEST1	999		Y
9	1003	Y	Y	Retest	Week 1	TEST1	49	Y	
10	1003	Y	Y	Week 2	Week 2	TEST1	499	Y	

To identify rows used for an Intent-to-Treat analysis for parameter code “TEST1” at Week 1 requires the following selection specification:

AVISIT=“Week 1” & PARAMCD=“TEST1” & ANL01FL=“Y” & ITTFL=“Y”

Similarly, to identify rows used for a Per-Protocol analysis of values of for parameter code “TEST1” at Week 1 requires the following selection specification:

AVISIT=“Week 1” & PARAMCD=“TEST1” & ANL01FL=“Y” & PPROTRFL="Y"

Since an error in the specification of the selection for either of the above conditions will yield incorrect results, it is important that the metadata be clear for each indicator variable. In addition, ADaM analysis results metadata will specify the selection criteria to provide clear documentation of how the indicator variables were used to select analyzed rows for identified analyses.

Example 2

Use of subject-level indicator variables and parameter-level indicator variables.

For the purposes of this example, it is assumed that the producer’s statistical analysis plan included a definition of an efficacy analysis population, defining it as consisting of subjects with a baseline efficacy assessment and at least one post-baseline efficacy assessment, without restriction to a specific assessment. In this example, there are two efficacy parameters (Test 1 and Test 2), and three visits (Week -1, Baseline, and Week 2). Subjects have results for the assessments as noted in Table 4.6.1.2.

Table 4.6.1.2 Example 2: Data available for each subject in illustration

Subject	Does Subject have a Baseline TEST 1 Assessment?	Does Subject have a Post-Baseline TEST 1 Assessment?	EFFPFL for TEST1	Does Subject have a Baseline TEST 2 Assessment?	Does Subject have a Post-Baseline TEST 2 Assessment?	EFFPFL for TEST2	EFFFL
1001	Y	Y	Y	Y	Y	Y	Y
1002	Y	Y	Y	N	N		Y
1003	Y	N		Y	Y	Y	Y
1004	Y	N		Y	N		N

In contrast to subject-level population flags, the column EFFPFL is a parameter-level population flag. A subject is included in the efficacy analysis population for a specific parameter if the subject has a baseline efficacy assessment and at least one post-baseline efficacy assessment for that parameter. If a subject is eligible for the efficacy analysis for the specific parameter, the variable EFFPFL is set to “Y” for the subject’s records within the parameter; it is null if the subject is not a candidate for the analysis of the parameter. In Table 4.6.1.3, the efficacy analysis population for TEST1 includes subjects 1001 and 1002; the efficacy analysis population for TEST2 includes subjects 1001 and 1003.

Table 4.6.1.3 Example 2: ADaM Dataset with Subject-Level and Parameter-Level Indicator Variables

Row	USUBJID	EFFFL	AVISIT	PARAMCD	AVAL	EFFPFL
1	1001	Y	Wk -1	TEST1	500	Y
2	1001	Y	Bsln	TEST1	500	Y
3	1001	Y	Wk 2	TEST1	600	Y
4	1001	Y	Wk -1	TEST2	10	Y
5	1001	Y	Bsln	TEST2	10	Y
6	1001	Y	Wk 2	TEST2	12	Y
7	1002	Y	Wk -1	TEST1	500	Y
8	1002	Y	Bsln	TEST1	500	Y
9	1002	Y	Wk 2	TEST1	46	Y
10	1002	Y	Wk -1	TEST2	11	
11	1003	Y	Wk -1	TEST1	780	
12	1003	Y	Bsln	TEST1	799	
13	1003	Y	Wk -1	TEST2	28	Y
14	1003	Y	Bsln	TEST2	30	Y
15	1003	Y	Wk 2	TEST2	32	Y
16	1004	N	Wk -1	TEST1	250	
17	1004	N	Bsln	TEST1	300	
18	1004	N	Wk -1	TEST2	15	
19	1004	N	Bsln	TEST2	15	

4.7 Identification of Rows Which Satisfy a Predefined Criterion for Analysis Purposes

For analysis purposes, criteria are often defined to group results based on the collected value's relationship to one or more algorithmic conditions. For example, subjects who had a result greater than five times the upper limit of the normal range or subjects who had a systolic blood pressure value > 160 mmHg with at least a 25 point increase from the BASE value. In addition to creating subgroups of subjects, the categorization of the presence or absence of a criterion is often used in listings, tabular displays or statistical modeling (as a covariate or a response variable).

4.7.1 ADaM Methodology and Examples When the Criterion Has Binary Responses

ADaM methodology provides an analysis criterion variable, CRITy, paired with a criterion evaluation result flag, CRITyFL, to identify whether a criterion is met. These variables are defined in Section 3.3.4. The variables MCRITy and MCRITyML are defined in Section 3.3.4 for use in situations where the criterion can have multiple responses (as opposed to CRITy which has binary responses).

CRITy is populated with a text description defining the conditions necessary to satisfy the presence of the criterion. The definition of CRITy can use any variable(s) located on the row, and the definition must stay constant across all rows within the same value of PARAM. A complex criterion which draws from multiple rows (different parameters or multiple rows for a single parameter) will require a new PARAM be created (see Example 3 Table 4.7.1.3).

CRITyFL, "Criterion Evaluation Result Flag", is the character indicator of whether the criterion described in CRITy was met. Variable CRITyFL must be present on the dataset if variable CRITy is present. CRITyFN is permitted if a numeric result flag is needed.

ADaM methodology allows the option of only populating CRITy on a row if the CRITy criterion is met for that row (see Example 1 Table 4.7.1.1). In that case, CRITyFL is set to "Y" only if CRITy is populated and is null otherwise. If this option is not used and CRITy is populated on all rows within the parameter (see Example 2 Table 4.7.1.2), then CRITyFL is set to "Y" or "N" or null. The choice of populating CRITy on only the rows where the criteria is met versus on all rows is dependent on the analysis need, as shown in the examples that follow.

CRITy and CRITyFL facilitate subgroup analyses. ADaM methodology does not preclude the addition of rows (in contrast to the addition of multiple CRITy and CRITyFL columns) to the BDS for the criterion CRITy. However, CRITy must be kept constant (if populated) across all rows within the same value of PARAM. CRITy, CRITyFL and CRITyFN are not parameter-invariant in that CRITy can vary across parameters within a dataset, as can the controlled terminology used for the corresponding CRITyFL and CRITyFN. In other words, CRITy for one parameter can be different than CRITy for a different parameter in the same dataset. (See Example 8: Categorical Analysis of Subjects Meeting Hy's Law Criteria in the document "ADaM Examples in Commonly Used Statistical Analysis Methods.")

Example 1
CRIT*y* populated only when criterion met.

Using this approach, when a criterion is defined for a PARAM but conditions are not met on a specific row, both CRITy and CRITyFL are set to null. CRITy and CRITyFL are also set to null if one or more missing data inputs to a criterion result in an unevaluable criterion (unevaluability is producer-defined, and is not necessarily triggered by missing data inputs).

One purpose of this option is to facilitate subsetting within a parameter when the interest is in the subgroup of subjects who fulfilled the criterion. It is also relevant when simple counts of criteria are desired. The following conditions must be true when this option is used:

1. Variables CRITy and CRITyFL are present on the dataset;
2. Analysis Variable Metadata defines CRITy relative to the specific parameter;
3. CRITy and CRITyFL are set to null for rows within the parameter where the criterion is not met or is unevaluable.

Table 4.7.1.1 illustrates ADaM methodology option "CRITy populated only when criterion met". The presence of a value in CRIT1 indicates Subject 1001 satisfied the criterion. With this option, CRIT1 facilitates subsetting when the interest is in the subgroup of subjects who fulfilled the criterion. The null value in CRIT1 is because Subject 1002 did not satisfy the criterion. The null value in CRIT1 is because the criterion is unevaluable due to missing inputs for Subject 1003.

Table 4.7.1.1 Example 1: ADaM Dataset with CRITy Populated Only When Criterion Met

Row	USUBJID	PARAM	AVAL	BASE	CHG	CRIT1	CRIT1FL
1	1001	Systolic Blood Pressure (mm Hg)	163	148	15	Systolic Pressure >160	Y
2	1002	Systolic Blood Pressure (mm Hg)	140	148	-8		
3	1003	Systolic Blood Pressure (mm Hg)		120			

Example 2

CRITy populated on all rows within a parameter.

Using this approach, CRITy is populated on all rows within the parameter and CRITyFL is set to “Y” or “N” or null. The purpose of this option is to facilitate analyses where the criterion is used in tabular displays and/or statistical modeling for the parameter.

Table 4.7.1.2 illustrates ADaM methodology option “CRITy populated on all rows within a parameter”. Since this criterion is used for modeling or analysis in this example, it is necessary to populate the rows which fail to satisfy the criterion. CRIT1FL indicates whether or not the subject meets the criterion. CRIT1FL is set to null for Subject 1005 because the criterion is unevaluable due to missing input(s).

Table 4.7.1.2 Example 2: ADaM Dataset with CRITy Populated on All Rows within a Parameter

Row	USUBJID	PARAM	AVAL	BASE	CHG	CRIT1	CRIT1FL
1	1001	Systolic Blood Pressure (mm Hg)	163	148	15	Systolic Pressure >160 and Change from Baseline in Systolic Pressure>10	Y
2	1002	Systolic Blood Pressure (mm Hg)	140	148	-8	Systolic Pressure >160 and Change from Baseline in Systolic Pressure>10	N
3	1005	Systolic Blood Pressure (mm Hg)	120			Systolic Pressure >160 and Change from Baseline in Systolic Pressure>10	

Example 3

Compound criteria.

If the definition of a criterion uses values located on multiple rows (different parameters or multiple rows for a single parameter), then a new row must be added with the value of PARAM being the textual description of the criterion (see Section 4.2.1, Rule 4 and Rule 5). The text of PARAM and CRITy are producer-defined and can be as long or as short as needed to be meaningful, within the 200-character limitation for the columns.

For compound criterion rows, AVALC must always be populated with Y/N/null. If an analysis also requires a numeric indicator variable, either of the following two options may be chosen:

1. CRITy may be set to the same criterion text as PARAM, CRITyFL set to the same Y/N/null value as AVALC, and CRITyFN set to 1/0/null.
2. AVAL may be set to a numeric 1/0/null indicator value.

If an analysis requires only simple subsetting of the “hits” on a particular compound criterion, it is acceptable to add only the “compound criterion met” (AVALC=“Y”) rows to the dataset. If this option is chosen, rows are not added where the assessment of a compound criterion in PARAM would result in AVALC=“N” or null.

Note that if a compound criterion is defined, then its components do not have to exist on their own in the dataset unless these components are themselves used for subsetting, display, or modeling purposes, or are needed for traceability.

Table 4.7.1.3 illustrates a compound criterion (row 3) included in the same dataset with non-compound criteria (rows 1 and 2).

Table 4.7.1.3 Example 3: ADaM Dataset with Both Compound and Non-compound Criteria

Row	USUBJID	PARAM	AVAL	AVALC	BASE	CHG	CRIT1	CRIT1FL	CRIT1FN	CRIT2	CRIT2FL	CRIT2FN
1	1001	Systolic Blood Pressure (mm Hg)	163		148	15	Systolic Pressure > 160	Y	1	Change from Baseline in Systolic Pressure > 10	Y	1
2	1001	Diastolic Blood Pressure (mm Hg)	96		87	9	Diastolic Pressure > 95	Y	1			
3	1001	Systolic Pressure >160 and Diastolic Pressure > 95		Y								

Note that criterion “Diastolic Pressure >95” (Row 2) can coexist in the same CRIT1 column with “Systolic Pressure >160” (Row 1). Each of these criteria is specific to its own subset of PARAM rows.

4.7.2 ADaM Methodology and Examples When the Criterion Has Multiple Responses

ADaM methodology provides an analysis criterion variable, MCRITy, paired with a criterion evaluation result flag, MCRITyML, to identify which level of a multiple response criterion is met. These variables are defined in Section 3.3.4.

MCRITy is populated with a text description identifying the criterion being evaluated. The definition of MCRITy can use any variable(s) located on the row and the definition must stay constant across all rows within the same value of PARAM. A complex criterion which draws from multiple rows (different parameters or multiple rows for a single parameter) will require a new PARAM be created.

MCRITyML, “Multi-Response Criterion y Evaluation”, is the character flag variable that indicates which level of the criterion defined in MCRITy was met. Variable MCRITyML must be present on the dataset if variable MCRITy is present. MCRITyMN is permitted if a numeric result flag is needed.

MCRITy and MCRITyML facilitate subgroup analyses. ADaM methodology does not preclude the addition of rows (in contrast to the addition of multiple MCRITy and MCRITyML columns) to the BDS for the criterion MCRITy. However, MCRITy must be kept constant (if populated) across all rows within the same value of PARAM.

MCRITy, MCRITyML and MCRITyMN are not parameter-invariant in that MCRITy can vary across parameters within a dataset, as can the controlled terminology used for the corresponding MCRITyML and MCRITyMN. In other words, MCRITy for one parameter can be different than MCRITy for a different parameter in the same dataset.

Example 1

Table 4.7.2.1 illustrates partial laboratory data for Alanine Aminotransferase (IU/L). As with other examples, some of the necessary columns for analysis and traceability (e.g., CHG, PCHG, AWTARGET, AWTDIFF, LBSEQ) have been excluded from the illustration. In this example, ALT values are evaluated for changes in toxicity grade. (Laboratory Grading in this example is based on CTCAE Version 4 so Grade 1 is >ULN - 3.0 x ULN, Grade 2 is >3.0 - 5.0 x ULN, Grade 3 is >5.0 - 20.0 x ULN, and Grade 4 >20.0 x ULN.) In a typical analysis situation, it is of interest to know the shift in the number of toxicity grades from baseline. Generally, an increase from a Grade 1 at baseline to a Grade 3 at a post baseline visit is treated the same as an increase from Grade 2 to Grade 4; that is, both of these are considered an increase in 2 Grades.

Note that for some laboratory analytes, only increases in toxicity grades are of interest. For other analytes, the interest is only in decreases in toxicity grades. Finally, for a few analytes, the change in toxicity in both directions (increases and decreases) is of interest.

In this example, it is increases in toxicity grades that are of interest. Within the analysis dataset, MCRIT1 identifies the criterion being evaluated and MCRIT1ML contains the level of criterion met, and MCRIT1MN contains a numeric version of the response. In contrast, CRIT1 assesses whether or not the value of ALT exceeded 8 times ULN.

Note that in this example, the producer has elected to not populate the values of MCRIT1ML on the screening and baseline records. Values of MCRIT1ML represent the number of grade increases from baseline. Note that for visits where there is either no increase in toxicity grade OR where the grade decreases, MCRIT1ML is given a value of ‘No Grade Increase’. This approach is suitable for analytes where only grade increases is of interest. Should decreases also be of interest, then a second set of MCRIT variables would be added to contain the observed number of toxicity grade decreases.

This example also illustrates other BDS variables, notably SHIFT1 and SHIFT2. SHIFT1 is defined as the shift between BNRIND and ANRIND while SHIFT2 is defined as the shift between BTOXGR and ATOXGR. These shifts of normal ranges and CTCAE toxicity grades are often of interest for analysis.

Table 4.7.2.1 ADaM Dataset with a Criterion that Has Multiple Responses

Row	USUBJID	PARAMCD	AVISIT	VISIT	ADY	AVAL	ANRLO	ANRHI	ANRIND	ATOXGR	ABLFL	ANL01FL	BASE	BNRIND	BTOXGR
1	ABC-0001	ALT	Baseline	SCREENING	-14	30	0	35	Normal	0	Y	Y	30	Normal	0
2	ABC-0001	ALT	Week 1	WEEK 1	2	31	0	35	Normal	0		Y	30	Normal	0
3	ABC-0001	ALT	Week 3	WEEK 3	22	45	0	35	High	1		Y	30	Normal	0
4	ABC-0001	ALT	Week 5	WEEK 5	34	81	0	35	High	1		Y	30	Normal	0
5	ABC-0001	ALT	Week 7	WEEK 7	51	110	0	35	High	2		Y	30	Normal	0
6	ABC-0001	ALT	Week 9	WEEK 9	65	554	0	35	High	3		Y	30	Normal	0
7	ABC-0001	ALT	Month 3	MONTH 3	92	1077	0	35	High	4		Y	30	Normal	0
8	ABC-0002	ALT	Screening	SCREENING	-14	30	0	31	Normal	0			32	High	1
9	ABC-0002	ALT	Baseline	WEEK 1	1	32	0	31	High	1	Y	Y	32	High	1
10	ABC-0002	ALT	Week 3	WEEK 3	21	23	0	31	Normal	0		Y	32	High	1
11	ABC-0002	ALT	Week 3	UNSCHEDULED	25	25	0	31	Normal	0			32	High	1
12	ABC-0002	ALT	Week 5	WEEK 5	39	33	0	31	High	1		Y	32	High	1
13	ABC-0002	ALT	Week 7	WEEK 7	53	100	0	31	High	2		Y	32	High	1
14	ABC-0002	ALT	Week 9	WEEK 9	64	27	0	31	Normal	0		Y	32	High	1
15	ABC-0002	ALT	Month 3	MONTH 3	89	22	0	31	Normal	0		Y	32	High	1
16	ABC-0002	ALT	Month 6	MONTH 6	181	20	0	31	Normal	0		Y	32	High	1

Row	SHIFT1	SHIFT2	MCRIT1	MCRIT1ML	MCRIT1MN	CRIT1	CRIT1FL
1 (cont)			ALT Grade Increase			ALT > 8*ULN	N
2 (cont)	Normal to Normal	0 to 0	ALT Grade Increase	No Grade Increase	0	ALT > 8*ULN	N
3 (cont)	Normal to High	0 to 1	ALT Grade Increase	Increase of 1 Grade	1	ALT > 8*ULN	N
4 (cont)	Normal to High	0 to 1	ALT Grade Increase	Increase of 1 Grade	1	ALT > 8*ULN	N
5 (cont)	Normal to High	0 to 2	ALT Grade Increase	Increase of 2 Grades	2	ALT > 8*ULN	N
6 (cont)	Normal to High	0 to 3	ALT Grade Increase	Increase of 3 Grades	3	ALT > 8*ULN	Y
7 (cont)	Normal to High	0 to 4	ALT Grade Increase	Increase of 4 Grades	4	ALT > 8*ULN	Y
8 (cont)			ALT Grade Increase			ALT > 8*ULN	N
9 (cont)			ALT Grade Increase			ALT > 8*ULN	N
10 (cont)	High to Normal	1 to 0	ALT Grade Increase	No Grade Increase	0	ALT > 8*ULN	N
11 (cont)	High to Normal	1 to 0	ALT Grade Increase	No Grade Increase	0	ALT > 8*ULN	N
12 (cont)	High to High	1 to 1	ALT Grade Increase	No Grade Increase	0	ALT > 8*ULN	N
13 (cont)	High to High	1 to 2	ALT Grade Increase	Increase of 1 Grade	1	ALT > 8*ULN	N
14 (cont)	High to Normal	1 to 0	ALT Grade Increase	No Grade Increase	0	ALT > 8*ULN	N
15 (cont)	High to Normal	1 to 0	ALT Grade Increase	No Grade Increase	0	ALT > 8*ULN	N
16 (cont)	High to Normal	1 to 0	ALT Grade Increase	No Grade Increase	0	ALT > 8*ULN	N

4.8 Examples of Timing Variables

4.8.1 Example of Phase, Period and Subperiod Variables

Table 4.8.1.1 provides a schematic example of the use of the BDS variables for phase, period, and subperiod. The example is of a study in which there are three analysis phases: Screening, Treatment, and Follow-up. The treatment phase consists of a two-period crossover design. In each period of the treatment phase, there are distinct subperiods in which the dose of the corresponding therapy is escalated, then maintained, and then de-escalated.

Table 4.8.1.1 Example of Phase, Period and Subperiod Variables

Variable	Variable Values							
APHASE	Screening	Treatment						Follow-up
APHASEN	1	2						3
APERIOD		1			2			
APERIODC		Crossover Period 1			Crossover Period 2			
ASPER		1	2	3	1	2	3	
ASPERC		Escalation	Maintenance	De-escalation	Escalation	Maintenance	De-escalation	

Note that, in general, there is no requirement to use all three of APHASE, APERIOD and ASPER when only one or two suffice. Also note that, in general, there is no requirement that the number and nature of subperiods, if any, be the same in each period. If ASPER is used, APERIOD must also be present.

4.9 Other Issues to Consider

The issues presented in the previous sections represent analysis decisions that commonly occur when creating ADaM datasets. However, the ADaM Team recognizes that those are not an exhaustive list. This section provides comment on some additional issues that may arise.

4.9.1 Adding Records to Create a Full Complement of Analysis Timepoints for Every Subject

It is not unusual for a given subject to have missing data for a specified analysis timepoint. For example, suppose an analysis is to be performed for the data obtained at each of 4 visits and that no imputation is to be performed. For subjects who did not attend all 4 visits, it would be possible to create records in the ADaM dataset for these missed assessments, with AVAL and AVALC missing (null) and appropriate variable(s) set to indicate these added records. For example, DTYPE could contain a producer-defined value such as “PHANTOM.” There are some advantages of having an ADaM dataset contain the same number of observations for each subject. For example, programming is facilitated by having the same data dimensions for all subjects, and by explicitly representing missing data rather than implicitly representing it by the absence of a record. This also allows ADaM datasets to support listing creation, especially for data that is not present in SDTM (e.g., added analysis parameters). For some categorical analyses, the denominators can be obtained directly from the ADaM dataset rather than from another input such as ADSL. The disadvantage of this approach is that it may require additional metadata to explain the use of these derived blank records and would require in some cases that subsetting statements be used to exclude the rows on which AVAL is missing. In general, the ADaM Team neither advocates nor discourages this practice for BDS datasets.

4.9.2 Creating Multiple Datasets to Support Analysis of the Same Type of Data

The statistical analysis plan often specifies that an analysis will be performed using slightly different methodologies. For example, the primary efficacy analysis may be performed using two different imputation algorithms for missing values. The producer must decide whether to include both sets of the imputed observations in one ADaM dataset or create two ADaM datasets, each representing just one of the imputation algorithms. ADaM provides variables that can be used to identify records that are used for different purposes. However, this does not imply that the producer should not or cannot submit multiple ADaM datasets of similar content, each designed for a specific analysis.

4.9.3 Size of ADaM Datasets

It is important to consider the size of ADaM datasets, because large datasets can pose problems for transferring between parties, loading into data warehouses, or software processing. The maximum size of a dataset and how to handle large datasets should be discussed with the recipient of the data and clearly documented. Refer to the FDA website (see Section 1.2) for recommendations concerning sizes of submitted datasets.

4.9.4 Traceability When the Multiple Imputation Method is Used

There has been increased attention in the analysis of clinical trial data to address problems associated with missing data and with this increased attention has come new ways to deal with this problem. In the past, simple methods such as ‘last observation carried forward’ or ‘baseline observation carried forward’ were routinely used to replace missing values. Such ‘single point imputation’ methods have been shown to underestimate the standard error of the estimates of various statistics computed from the data. A more sophisticated method, termed ‘multiple imputation’ was introduced in 1987 and is now fully supported by frequently used software packages. In brief, this methodology deals with the uncertainty of the missing data by employing a three step process. The first step is the creation of multiple datasets in which plausible values for each missing data value are imputed. The second step is to analyze each of these datasets with the desired statistical procedure and capture the resulting statistical estimates. The third and final step is to use these estimates to generate a combined (pooled) estimate. It is these estimates which are based on the pooling of the estimates from the multiple imputation datasets that are used to evaluate statistical significance.

Using the SAS software as an example, the above three step process is achieved using PROC MI to create the multiple datasets (Step 1). Step 2 would utilize the procedure associated with the desired statistical model, such as PROC LOGISTIC. Step 3 would use PROC MIANALYZE to create the combined pooled estimates.

In ADaM, the documentation of derived variables via variable level metadata, and statistical results via analysis results metadata, is paramount to achieving the concept of traceability. However, documenting the traceability of estimates created via multiple imputation cannot be achieved with these current metadata methods. Additionally, it would not be practical to include all datasets that are created from the PROC MI process as part of a submission. To address traceability, the recommendation from ADaM is to provide the program statements from the three procedures mentioned above as a part of the analysis results metadata. This would allow the reviewer to re-create the analysis as desired. Of primary importance is to ensure that the options used in PROC MI, specifically the value of the seed, the number of iterations, and the method used for imputation are clearly denoted.

4.9.5 Copying Values onto a New Record

As a general rule, when a new record is derived from a single record in the dataset, retain on the derived record any variable values from the original record that do not change and that make sense in the context of the new record (e.g., --SEQ, VISIT, VISITNUM, --TPT, covariates, etc.). When a record is derived from multiple records, then retain on the derived record all variable values that are constant across the original records, do not change, and that make sense in the context of the new record. Note that there are situations when retention of values from an original record or records would make no sense on the derived record; in such cases, do not retain those values. Refer to Table 4.2.1.3 and Table 4.2.1.4 for two examples.

Appendices

Appendix A: Abbreviations and Acronyms

The following is a list of abbreviations and acronyms used multiple times in this document. Not included here are explanations of the various SDTM domains (e.g., QS, DM). Also not included is a description of the variables referenced.

ADaM	CDISC Analysis Data Model
ADaM model document	The document entitled “Analysis Data Model (ADaM)”
ADaMIG	Analysis Data Model Implementation Guide
ADSL	ADaM Subject-Level Analysis Dataset
BDS	ADaM Basic Data Structure
BLOCF	Baseline Observation Carried Forward
CDASH	Clinical Data Acquisition Standards Harmonization
CDISC	Clinical Data Interchange Standards Consortium
DAO	Data as Observed
eCTD	electronic Common Technical Document
FDA	United States Food and Drug Administration
ITT	Intent-to-Treat
LOCF	Last Observation Carried Forward
LOV	Last Observed Value
LVPD	Last Value Prior to Dosing
OCCDS	ADaM Occurrence Data Structure
SAP	Statistical Analysis Plan
SDS	Submission Data Standards
SDTM	Study Data Tabulation Model
SDTMIG	Study Data Tabulation Model Implementation Guide
TAUG	Therapeutic Area User Guide
FDA TCG	FDA Study Data Technical Conformance Guide
WOC	Worst Observed Case
WOCF	Worst Observation Carried Forward
XML	Extensible Markup Language

Appendix B: Revision History

This section lists changes in the ADaMIG from version 1.0 to version 1.1.

Category/Section	Type	Description
General	Clarification	Made changes throughout the document to increase clarity.
General	Clarification	Clarified scope (i.e., within a variable, within a parameter, within a dataset, or within a study) throughout the document.
General	Update	Changed wording from “analysis datasets” to “ADaM datasets” where applicable.
General	Clarification	Changed wording from “analysis variable” to “analysis value” where applicable.
General	Clarification	Changed wording from originator/sender/owner/sponsor of the data to “producer” where appropriate. Changed wording from reviewer/user/recipient of the data to “consumer” where appropriate.
General	Clarification	Increased precision and consistency of text referring to date, time and datetime variables. For example, used “datetime” instead of variants such as “date/time” when describing ADaM numeric datetime variables.
Section 1.1 Purpose	Clarification	Updated Section 1.1, “Purpose”, to provide more detail.
Section 1.2 Background	Clarification	Updated Section 1.2, “Background”, to provide more detail, and to refer readers to the FDA website for additional information.
Section 1.3 What is Covered in the ADaMIG	Addition	Added reference to the external document that describes the OCCDS data structure.
Section 1.3.1 Other ADaM-related CDISC Documents	Addition	Added section 1.3.1, “Other ADaM-related CDISC Documents”.
Section 1.4 Organization of the Document	Addition	Noted changes in wording from the terms originator/sender/owner/sponsor to “producer”, and from the terms reviewer/user/recipient to “consumer”.
Section 1.5.1 General ADaM Definitions	Update	Updated definition of “record” to include “observation” in addition to “row” in a dataset.
Section 1.5.2 Basic Data Structure Definitions	Clarification	Added clarity to the definition of parameter-invariant. Added definition for term parameter-variant.
Section 1.6 Analysis Datasets and ADaM Datasets	Addition	Added section 1.6, “Analysis Datasets and ADaM Datasets”, to clarify the distinctions among “analysis datasets”, “ADaM Datasets”, and “non-ADaM analysis datasets”.
Section 2.1 Fundamental Principles	Clarification	Minor modifications to text to clarify concepts.
Section 2.2 Traceability	Addition	Added a cross-reference to section 1.5.1. Added mention of the BDS and OCCDS structures.
Section 2.3 The ADaM Data Structures	Clarification	Minor modifications to text to clarify concepts.
Section 2.3.1 The ADaM Subject-Level Analysis Dataset (ADSL)	Update	Significant updates to the text to clarify the contents of ADSL and to add tips on how ADSL variables should be carried forward to BDS datasets.
Section 2.3.2 The ADaM Basic Data Structure (BDS)	Update	Clarified analysis timepoints and included a reference to section 1.5.2. Clarified the description of the structure of the BDS.
Section 3 Standard ADaM Variables	Clarification	Expanded and added clarity to the first three paragraphs, including to the concept of standard naming fragments.

Category/Section	Type	Description
Section 3 Standard ADaM Variables	Format	Consolidated and updated existing content into new section 3.1, “ADaM Variable Conventions”. Numbering of all following subsections of section 3 updated accordingly.
Section 3.1.1 General Variable Conventions	Format	Moved general variable naming conventions into new section 3.1.1. Reordered that content.
Section 3.1.1 General Variable Conventions	Clarification	Clarified that there is no requirement that digits be consecutive when replacing w, xx, y, zz in variable names.
Section 3.1.1 General Variable Conventions, Item 2a	Addition	Added “w” reference in variable name as an index to the wth variable
Section 3.1.1 General Variable Conventions, Item 2c	Update	Updated reference to “y” variables to allow non-left-zero-padded indices in variable names to go from 1-99 now.
Section 3.1.1 General Variable Conventions, Item 3	Update	Added statement that variable length of SDTM variables can differ between SDTM and ADaM in order to optimize file size.
Section 3.1.1 General Variable Conventions, Item 4	Addition	Added requirement that if an ADaM standard variable name has been defined for a specific concept, then the ADaM standard variable name must be used.
Section 3.1.1 General Variable Conventions, Items 5 and 6	Clarification	Clarified paired variables and one-to-one mapping.
Section 3.1.1 General Variable Conventions, Item 8	Clarification	Clarified the instructions for FL variables.
Section 3.1.1, General Variable Conventions, Item 9	Update	Added Gy and CATy variables to the discussion about GRy variables.
Section 3.1.2 Timing Variable Conventions	Format	Moved timing variable conventions into new section 3.1.2. Reordered that content.
Section 3.1.2 Timing Variable Conventions, Item 9	Clarification	Clarified the concept of anchor dates and times. Stated that relevant anchor date and time variables should be specified in metadata and included in ADSL or current dataset.
Section 3.1.3 Date and Time Imputation Flag Variables	Addition	Moved date and time imputation flag variable conventions into new section 3.1.3.
Section 3.1.4 Flag Variable Conventions	Format	Moved flag variable conventions into new section 3.1.4. Reordered that content.
Section 3.1.4 Flag Variable Conventions	Clarification	Added details to Items 1, 3, 6, 7, 8 and 9.
Section 3.1.5 Variable Naming Fragments	Addition	Added Section 3.1.5, “Variable Naming Fragments”.
Section 3.1.6 Additional Information about Section 3	Update	Moved text following “Additional Information about Section 3” to new section 3.1.6, and expanded.
Section 3.1.6 Additional Information about Section 3	Update	Clarified when and how variable labels may be modified.
Sections 3.2 and 3.3	Clarification	Clarified and added text in this section, including in the CDISC Notes of many variables.
Section 3.2 ADSL Variables	Format	Divided previous Table 3.1.1 into different tables (3.2.x) to separate ADSL variables into logical groups.
Section 3.2 ADSL Variables	Clarification	Clarified that an “ADaM compliant” ADSL is required in a CDISC submission.
Table 3.2.1 ADSL Identifier Variables	Clarification	Revised text to specify that SUBJID and SITEID are required in ADSL but permissible in other datasets.
Table 3.2.1 ADSL Identifier Variables	Addition	Added geographical region variables REGIONy and REGIONyN.

Category/Section	Type	Description
Table 3.2.2 ADSL Subject Demographics Variables	Addition	Added age grouping variables AGEGRy, AGEGRyN, and analysis age variable AAGE.
Table 3.2.2 ADSL Subject Demographics Variables	Clarification	Clarified CDISC notes for AGE and AGEU.
Table 3.2.4 ADSL Treatment Variables	Addition	Added SDTM variable ACTARM.
Table 3.2.4 ADSL Treatment Variables	Update	Corrected the Type of TRxxPGyN and TRxxAGyN from Char to Num.
Table 3.2.4 ADSL Treatment Variables	Addition	Added pooled treatment sequence variables TSEQPGy, TSEQPGyN, TSEQAGy, and TSEQAGyN.
Table 3.2.5 ADSL Dose Variables	Addition	Added dosing variables DOSExxP, DOSExxA, and DOSExxU.
General	Clarification	Changed occurrences of the terms “date-time” and “date/time” to “datetime” when referring to ADaM variables.
Table 3.2.7 Subject-Level Period, Subperiod, and Phase Timing Variables	Addition	Added new variables PxxSw, PxxSwSDT, PxxSwSTM, PxxSwSDM, PxxSwSDF, PxxSwSTF, PxxSwEDT, PxxSwETM, PxxSwEDM, PxxSwEDF, PxxSwETF, APHASEw, PHwSDT, PHwSTM, PHwSDTM, PHwSDTF, PHwSTMF, PHwEDT, PHwETM, PHwEDTM, PHwEDTF, and PHwETMF.
Table 3.2.8 ADSL Subject-Level Trial Experience Variables	Addition	Added new variables EOSSTT, EOSDT, DCSREAS, DCSREASP, EOTSTT, DCTREAS, DCTREASP, EOTxxSTT, DCTxxRS, DCTxxRSP, EOPxxSTT, DCPxxRS, DCPxxRSP, RFICDT, ENRLDT, RFICyDT, ENRLyDT, RANDyDT, LSTALVDT, TRCMP, TRCMPGy, TRCMPGyN, TRxxDURD, TRxxDURM, TRxxDURY, TRTDURD, TRTDURM, TRTDURY, DTHDT, DTHDTF, DTHCAUS, DTHCAUSN, DTHCGRy, and DTHCGRyN.
Table 3.3.1.1 Identifier Variables for BDS Datasets	Addition	Added variable ASEQ.
Section 3.3.2 Record-Level Treatment and Dose Variables for BDS Datasets	Addition	Added record-level dose variables DOSEP, DOSCUMP, DOSEA, DOSCUMA, and DOSEU.
Section 3.3.2 Record-Level Treatment Variables for BDS Datasets	Update	Updated section 3.3.2, including the CDISC Notes for TRTP, to state that at least one treatment variable is required, which may be a subject-level or record-level variable. Changed Core value from Required to Conditional for TRTP.
Table 3.3.3.1 Timing Variables for BDS Datasets	Clarification	Changed “reference date” to “anchor date” in CDISC Notes for ADY, ASTDY, AENDY. Changed “reference time” to “anchor time” in the CDISC Notes for ARELTM.
Table 3.3.3.1 Timing Variables for BDS Datasets	Update	Changed Core value from Conditional to Permissible for ARELTMU.
Table 3.3.3.1 Timing Variables for BDS Datasets	Addition	Added variables APHASEN, ASPER, ASPERC.
Table 3.3.3.2 Period, Sub-period, and Phase Start and End Timing Variables	Addition	Added variables APERSDT, APERSTM, APERSDTM, APERSDTF, APERSTMF, APEREDT, APERETM, APEREDTM, APEREDTF, APERETMF, ASPRSDT, ASPRSTM, ASPRSDTM, ASPRSDTF, ASPRSTMF, ASPREDT, ASPRETM, ASPREDTM, ASPREDTF, ASPRETMF, PHSDT, PHSTM, PHSDTM, PHSDTF, PHSTMF, PHEDT, PHETM, PHEDTM, PHEDTF, and PHETMF.
Section 3.3.4 Analysis Parameter Variables for BDS Datasets	Addition	Added text to clarify the purpose and relationships of PARAM, AVAL and AVALC, in contrast to SDTM Findings Class variables --TEST, --STRESN, and --STRESC.

Category/Section	Type	Description
Table 3.3.4.1 Analysis Parameter Variables for BDS Datasets	Update	Removed requirement that PARAMN must be integer.
Table 3.3.4.1 Analysis Parameter Variables for BDS Datasets	Update	Added note that PARAMTYP will be retired in the next version.
Table 3.3.4.1 Analysis Parameter Variables for BDS Datasets	Update	Specified Core separately for AVAL and AVALC. Changed Core value from Required to Conditional for AVAL and AVALC, but specified that at least one of the two is required.
Table 3.3.4.1 Analysis Parameter Variables for BDS Datasets	Addition	Added variables AVALCAyN, BASECAyN, CHGCATyN, and PCHGCAyN.
Table 3.3.4.2 Analysis Parameter Criteria Variables for BDS Datasets	Addition	Added multi-response criterion evaluation variables MCRITy, MCRITyML, and MCRITyMN.
Section 3.3.5 Analysis Descriptor Variables for BDS Datasets	Clarification	Added text to clarify DTYPE and visit windowing variables.
Section 3.3.6 Time-to-Event Variables for BDS Datasets	Update	Moved Time-to-Event Variables to new section 3.3.6, and aligned with the document titled "The ADaM Basic Data Structure for Time-to-Event Analyses", v1.0, including addition of variables STARTDTM, STARTDTF, STARTTMF, and CNSDTDSC.
Table 3.3.7 Toxicity and Range Variables for BDS Datasets	Update	Generalized the toxicity and range variables previously found in table 3.2.5.4, "Lab Related Analysis Variables for BDS Datasets", to be applicable to non-laboratory-test data. These variables are now in their own section 3.3.7, "Toxicity and Range Variables for BDS Datasets".
Table 3.3.7 Toxicity and Range Variables for BDS Datasets	Update	Corrected the Type of AyLO and AyHI from Char to Num.
Table 3.3.7 Toxicity and Range Variables for BDS Datasets	Addition	Added variables ANRLOC, ANRHIC, AyLOC, AyHIC, AyIND, and ByIND.
Table 3.3.8.2 BDS Population Indicator(s) Variables	Update	Corrected the Variable Name of variable COMPRFL to COMPLRFL, and corrected the Variable Name of COMPPFL to COMPLPFL, to make consistent with the subject-level population flag variable COMPLFL.
Table 3.3.8.2 BDS Population Indicator(s) Variables	Update	Changed the Core of parameter-level and record-level population flags from Cond to Perm.
Table 3.3.8.2 BDS Population Indicator(s) Variables	Update	Removed numeric versions of parameter-level and record-level population flag variables ITTRFN, SAFRFN, FASRFN, PPROTRFN, COMPLRFN, ITTPFN, SAFPFN, FASPFN, PPROTPFN, and COMPLPFN.
Section 3.3.9 Datapoint Traceability Variables	Update	Created new section about variables useful for datapoint traceability, including and expanding upon relevant content that was in the previous section 3.2.8, "Other Variables".
Section 3.4 Analysis-Enabling Variables	Format	Moved discussion of analysis-enabling variables from previous section 3.2.8, "Other Variables" to new section 3.4.
Section 3.5, Differences between SDTM and ADaM Population and Baseline Flags	Update	Corrected mention of ADaM variable name COMPFL to COMPLFL.
Example 1 in Section 4.1 Examples of Treatment Variables for Common Trial Designs	Clarification	Added ACTARM, TRTSDT, and TRTEDT to the parallel group design example. Removed TRT01SDT and TRT01EDT from the example because they are not required in a one-period treatment study. Clarification added to text.
Example 5 in Section 4.1 Examples of Treatment Variables for Common Trial Designs	Addition	Added new example showing TRTP representing treatment at time of actual visit within a BDS structure, along with TRT01P and TRT02P (from ADSL).

Category/Section	Type	Description
Example 6 in Section 4.1 Examples of Treatment Variables for Common Trial Designs	Addition	Added another example showing TRTP representing treatment used for analysis within a BDS structure in an alternative to Example 5 in the same section.
Section 4.2 Creation of Derived Columns versus Creation of Derived Rows	Addition	Added list of rules that are each covered in separate subsections of section 4.2.1.
Section 4.2.1 Rules for the Creation of Rows and Columns	Added	Added explanation of bolding in the examples in section 4.2.1.
Rule 1 in Section 4.2.1 Rules for the Creation of Rows and Columns	Clarification	Clarified text regarding parameter invariance.
Rule 2 in Section 4.2.1 Rules for the Creation of Rows and Columns	Clarification	Added VISIT to Table 4.2.1.2 for comparison with AVISIT. Added clarification about what variables and values to retain when deriving a record from an individual existing record.
Rule 3 in Section 4.2.1 Rules for the Creation of Rows and Columns	Update	Deleted mention of PARAMTYP from 3rd paragraph and deleted PARAMTYP from Table 4.2.1.4. Changed the SDTM source domain for the migraine data in table 4.2.1.7 from Clinical Findings (CF) to Findings About (FA). Added text to clarify crossover example shown in Table 4.2.1.8.
Rule 6 in Section 4.2.1 Rules for the Creation of Rows and Columns	Addition	Expanded description of original example shown in Table 4.2.1.11. Added Table 4.2.1.12 to show a solution for the case where only the most recent baseline is needed for change from baseline analysis. Added text regarding relationship among definitions of baseline, ABLFL, and BASETYPE.
Section 4.3.1 ADaM Methodology and Examples	Addition	Added QSSEQ to Tables 4.3.1.1 and 4.3.1.3.
Table 4.3.1.3	Addition	Added PARAMCD to table.
Section 4.4.1 ADaM Methodology and Examples	Clarification	Clarified wording about ADaM data sources in the first paragraph.
Section 4.4.1 ADaM Methodology and Examples	Update	Changed the SDTM source domain for the hospitalization data from Disposition (DS) to Hospitalization (HO), inserting table 4.4.1.3.
Example 2 in Section 4.4.1 ADaM Methodology	Update	Replaced SRC* variables with LBSEQ in Table 4.4.1.5, because all data came from a single source dataset. Clarified that SRC* variables are valid as well. Added PARAM to table. Removed eGFR equation footnote.
Example 3 Section 4.4.1 ADaM Methodology	Addition	Added new example 3 using SRC* variables pointing to predecessor ADaM dataset.
Example 1 in Section 4.5.1.1 ADaM Methodology and Examples	Update	Added VSSEQ to Table 4.5.1.1 to document traceability methodology.
Example 2 in Section 4.5.1.1 ADaM Methodology and Examples	Update	Added VSSEQ to Table 4.5.1.2 to document traceability methodology.
Example 1 in Section 4.5.2.1 ADaM Methodology and Examples	Update	Added VSSEQ to Table 4.5.2.1 to document traceability methodology.
All Examples in Section 4.5.2.1 ADaM Methodology and Examples	Update	Added BASE to all tables to demonstrate which value is used as baseline,
Example 1 in Section 4.5.4.1 ADaM Methodology and Examples	Update	Added AWTARGET and AWTDIFF to table and discussed in text in order to indicate more clearly how the analyzed rows were selected from the candidate rows.
Example 3 in Section 4.5.4.1 ADaM Methodology and Examples	Clarification	Added reference to windowing method defined in SAP.
Example 2 Section 4.5.4.2	Update	Changed values of DTYPE in example.

Category/Section	Type	Description
Example 1 in Section 4.6.1 ADaM Methodology and Examples/ Example 1	Clarification	Added text to explain the overall concept that subject-level and row-level flag variables allow analyses on multiple populations of interest within the same dataset.
Example 2 in Section 4.6.1 ADaM Methodology and Examples/ Example 2	Addition	Added new example 2 using subject-level and parameter-level population indicator (flag) variables.
Section 4.7.1 ADaM Methodology and Examples When the Criterion Has Binary Responses	Addition	Added mention of MCRIT* variables. Added text to clarify that the alternative methods for populating CRITy and CRITyFL support different analysis needs.
Example 3 of Section 4.7.1 ADaM Methodology and Examples When the Criterion Has Binary Responses	Deletion	Removed mention of PARAMTYP.
4.7.2 ADaM Methodology and Examples When the Criterion Has Multiple Responses	Addition	Added section 4.7.2 to provide examples of the new variables MCRITy, MCRITyML and MCRITyMN.
Section 4.8 Examples of Timing Variables, with 4.8.1 Example of Phase, Period and Subperiod Variables.	Addition	Inserted new section 4.8, containing new subsection 4.8.1, in order to provide an example of the relationship among phase, period and subperiod variables.
Section 4.9 Other Issues to Consider	Format	Moved previous section 4.8 to Section 4.9.
Section 4.9.3 Size of ADaM Datasets	Addition	Added new section 4.9.3.
Section 4.9.4 Traceability When the Multiple Imputation Method is Used	Addition	Added new section 4.9.4.
Section 4.9.5 Copying Values onto a New Record	Addition	Added new section 4.9.5.
Appendix B	Addition	Inserted Appendix B, “Change History”.

Appendix C: Representations And Warranties; Limitations of Liability, And Disclaimers

CDISC Patent Disclaimers
It is possible that implementation of and compliance with this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any claim or of any patent rights in connection therewith. CDISC, including the CDISC Board of Directors, shall not be responsible for identifying patent claims for which a license may be required in order to implement this standard or for conducting inquiries into the legal validity or scope of those patents or patent claims that are brought to its attention.

Representations and Warranties
"CDISC grants open public use of this User Guide (or Final Standards) under CDISC's copyright."

Each Participant in the development of this standard shall be deemed to represent, warrant, and covenant, at the time of a Contribution by such Participant (or by its Representative), that to the best of its knowledge and ability: (a) it holds or has the right to grant all relevant licenses to any of its Contributions in all jurisdictions or territories in which it holds relevant intellectual property rights; (b) there are no limits to the Participant's ability to make the grants, acknowledgments, and agreements herein; and (c) the Contribution does not subject any Contribution, Draft Standard, Final Standard, or implementations thereof, in whole or in part, to licensing obligations with additional restrictions or requirements inconsistent with those set forth in this Policy, or that would require any such Contribution, Final Standard, or implementation, in whole or in part, to be either: (i) disclosed or distributed in source code form; (ii) licensed for the purpose of making derivative works (other than as set forth in Section 4.2 of the CDISC Intellectual Property Policy ("the Policy")); or (iii) distributed at no charge, except as set forth in Sections 3, 5.1, and 4.2 of the Policy. If a Participant has knowledge that a Contribution made by any Participant or any other party may subject any Contribution, Draft Standard, Final Standard, or implementation, in whole or in part, to one or more of the licensing obligations listed in Section 9.3, such Participant shall give prompt notice of the same to the CDISC President who shall promptly notify all Participants.

No Other Warranties/Disclaimers. ALL PARTICIPANTS ACKNOWLEDGE THAT, EXCEPT AS PROVIDED UNDER SECTION 9.3 OF THE CDISC INTELLECTUAL PROPERTY POLICY, ALL DRAFT STANDARDS AND FINAL STANDARDS, AND ALL CONTRIBUTIONS TO FINAL STANDARDS AND DRAFT STANDARDS, ARE PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE, AND THE PARTICIPANTS, REPRESENTATIVES, THE CDISC PRESIDENT, THE CDISC BOARD OF DIRECTORS, AND CDISC EXPRESSLY DISCLAIM ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR OR INTENDED PURPOSE, OR ANY OTHER WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, FINAL STANDARDS OR DRAFT STANDARDS, OR CONTRIBUTION.

Limitation of Liability
IN NO EVENT WILL CDISC OR ANY OF ITS CONSTITUENT PARTS (INCLUDING, BUT NOT LIMITED TO, THE CDISC BOARD OF DIRECTORS, THE CDISC PRESIDENT, CDISC STAFF, AND CDISC MEMBERS) BE LIABLE TO ANY OTHER PERSON OR ENTITY FOR ANY LOSS OF PROFITS, LOSS OF USE, DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, OR SPECIAL DAMAGES, WHETHER UNDER CONTRACT, TORT, WARRANTY, OR OTHERWISE, ARISING IN ANY WAY OUT OF THIS POLICY OR ANY RELATED AGREEMENT, WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

Note: The CDISC Intellectual Property Policy can be found at http://www.cdisc.org/system/files/all/article/application/pdf/cdisc_20ip_20policy_final.pdf.