

## A Guide to the ADaM Basic Data Structure for Dataset Designers

Michelle Barrick, Eli Lilly and Company, Indianapolis, Indiana  
John Troxell, John Troxell Consulting LLC, Bridgewater, New Jersey

### Abstract

The Clinical Data Interchange Standards Consortium (CDISC) Analysis Data Model (ADaM) Implementation Guide (ADaMIG), published in 2009, describes the many components of a very powerful and flexible analysis dataset structure called the Basic Data Structure (BDS), and provides some rules and examples. The BDS is unique among CDISC data structures in the flexibility it provides for the addition of various kinds of derived rows to meet analysis needs. A companion CDISC document, "CDISC Analysis Data Model (ADaM) Examples in Commonly Used Statistical Methods," published in 2011, provides more in-depth examples of ADaM data and metadata solutions in particular scenarios. However, neither of the two documents provides a holistic explanation of the BDS that describes how the structurally-important variables and kinds of observed and derived rows function together. In this paper, the authors define categories of observed and derived rows. These definitions underpin a unified explanation of the BDS that provides an understanding of how the various kinds of rows and the structural BDS variables interact. Such knowledge is essential in order to design the appropriate solution for each data scenario and analysis need.

### Introduction

The motivation for this paper was the authors' need to provide guidance to new developers of ADaM BDS datasets. The explanation presented in this paper seems to satisfy the need of study dataset designers to understand the basics of how a BDS dataset works. The authors hope that the explanation might be of use to others, and also hope to stimulate progress toward greater clarity in the standard itself.

The paper describes the BDS in the context of change from baseline analysis, because it offers a good opportunity to describe important structural variables and types of rows in the BDS, and their interactions. The definitions and concepts other than those specific to baseline are relevant to other applications of the BDS.

## 1. Definitions of Observed and Derived

### 1.1 SDTM

For the purposes of this paper, all data values in Study Data Tabulation Model (SDTM) datasets are considered to be observed. This is despite the fact that in actuality a data value may have been derived prior to or during the creation of the SDTM datasets. For example, an SDTM value of LDL cholesterol derived by the Friedewald calculation from total cholesterol, HDL, and triglycerides, is considered to be observed, even though in actuality it was derived by the laboratory.

### 1.2 ADaM BDS Parameter

In this paper, a BDS parameter derived from SDTM data by any method, whether by simple copying of a data value, or through a complex derivation involving multiple inputs, is considered to be an observed parameter.

A BDS parameter that is derived from one or more other BDS parameters is considered to be a derived parameter.

See also section 2.4, "Parameter Type (PARAMTYP)."

### 1.3 Rows within BDS Parameters

In this paper, any BDS parameter, regardless of whether it is observed or derived, is considered to have an initial set of one or more observed rows.

Subsequent to the creation of the initial set of observed rows of a BDS parameter, additional rows may be derived from the initial set of observed rows in order to support analysis needs.

Derived rows may be created to represent imputation of missing data, to create summaries of multiple rows, and/or to make use of alternative baseline definitions.

This subject is discussed in some detail in section 4, "Categories of Rows within a Parameter," and elsewhere in the paper.

## 2. BDS Parameter Variables and Considerations

### 2.1 Parameter (PARAM) and Parameter Code (PARAMCD)

Parameter (PARAM) describes, and must uniquely and sufficiently identify, the contents of the relevant analysis value variable AVAL or AVALC (including necessary details such as units, specimen type, body position if relevant to the analysis, or anything else needed). Parameter (PARAM) and parameter code (PARAMCD) are a one-to-one map (ADaMIG v1.0 Sec. 3.2.4).

### 2.2 One of the Differences between PARAM and SDTM xxTEST

It is important to understand a key difference in approach between the SDTM Findings class variable xxTEST and the ADaM BDS variable PARAM. SDTM xxTEST 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. 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.

### 2.3 Parameter Category (PARCATy)

PARCATy is used to group parameters into categories. For example, PARCAT1 could be "Histology", or "Chemistry". Each parameter belongs to only one value of PARCAT1 (e.g., a parameter is in the Histology category or in the Chemistry category, but not both). Another example is PARCAT2 = "SI", "CN", or "SICN", for lab parameters whose units are S.I. units, conventional units, or both, respectively. PARCATy should not be used as a qualifier on PARAM. PARCATy cannot be used to subdivide the data within a given PARAM.

### 2.4 Parameter Type (PARAMTYP)

Parameter Type (PARAMTYP) is a permissible CDISC ADaM variable that flags "derived" parameters. PARAMTYP has the same constant value (either null or DERIVED) across all rows of a given parameter.

If a parameter is derived from other ADaM parameters, then PARAMTYP="DERIVED". Otherwise, PARAMTYP=null.

It is important to understand that whether or not PARAMTYP="DERIVED" or null, the rows within the parameter can still be classified as "observed" or "derived" per the row definitions described in section 4, "Categories of Rows within a Parameter."

See also section 1, "Definitions of Observed and Derived." PARAMTYP is also discussed in the introduction of section 7, "Examples of Row Categories and Selection Criteria," as well as in the notes for Table 5 in that section.

## 3. Derivation Type, Baseline Type, Baseline Record Flag, and Analysis Visit

Please refer to section 4, "Categories of Rows within a Parameter," for critical information that must be understood about the purpose and use of these variables.

### 3.1 Derivation Type (DTYPE)

DTYPE serves two functions: when populated on a given record, (1) it indicates that the record is derived from other records within the same parameter\*, and (2) it identifies the algorithm used to derive the analysis value (AVAL or AVALC) on the record. DTYPE is always non-null for Derived Timepoint rows (i.e., a row derived from other rows within the same parameter for the purpose of creating or imputing a timepoint). Each value of DTYPE refers to an algorithm for deriving records from other records in the parameter. The values of DTYPE and their corresponding definitions must be described in metadata.

Examples and discussion of DTYPE are provided in the notes following Tables 2, 3, 4, and 5 in section 7, "Examples of Row Categories and Selection Criteria."

\* Note that there may be missing value imputation methods available that draw upon other inputs than the existing records of the same parameter, but for the purposes of this paper, they are ignored.

### 3.2 Baseline Type (BASETYPE)

For a given parameter, if Baseline Value (BASE) is populated, and there is more than one definition of baseline, then BASETYPE must be non-null on all records of any type for that parameter. Each value of BASETYPE refers to a definition of baseline that characterizes the value of BASE on that row. The values of BASETYPE and their corresponding definitions must be described in metadata.

See also section 3.3, "Baseline Record Flag (ABLFL)."

Examples and discussion of BASETYPE are provided in Table 4 of section 7, "Examples of Row Categories and Selection Criteria" and its notes.

### 3.3 Baseline Record Flag (ABLFL)

If the baseline column BASE is populated for a parameter, then baseline flag ABLFL must be set to "Y" on the row whose analysis value AVAL is used to populate BASE for that parameter and subject. Importantly however, the BDS can handle situations where there is more than one definition of baseline. This capability is needed when different analyses call for different baseline definitions.

For example, assume that for a given BDS parameter, there are two definitions of baseline indicated in BASETYPE:

- "LAST" = last value prior to treatment;
- "AVERAGE" = average of multiple measurements made at the baseline visit prior to treatment.

Since there are two definitions of baseline (and hence two values of BASETYPE), there are two records for each subject where the baseline record flag ABLFL is set to "Y", one for each value of BASETYPE. Since there is more than one definition of baseline, variable BASETYPE is required, and must be populated.

Examples and discussion of ABLFL are provided in Tables 1 and 4 in section 7, "Examples of Row Categories and Selection Criteria," and their notes.

### 3.4 Analysis Visit (AVISIT)

When present, analysis visit AVISIT is used to describe the analysis visit or conceptual timepoint characterizing the row. For example, AVISIT could be "VISIT 1", or "WEEK 14", or "POST BASELINE MAX".

Examples of AVISIT for observed and derived rows are shown in Tables 1-5 of section 7, "Examples of Row Categories and Selection Criteria."

## 4. Categories of Rows within a Parameter

This section describes categories of rows within a parameter. The discussion in this section applies equally to parameters that are derived (PARAMTYP="DERIVED") and parameters that are not derived (PARAMTYP=null).

### 4.1 Row Structure

The BDS structure is unique among CDISC data structures in the critical importance of row structure. BDS row structure is variable and complex depending on the analysis need. One cannot understand a BDS dataset without understanding its row structure and how the standard variables interact with it. In addition to specifying the columns (variables) in the ADaM dataset, it is equally important to specify how to create rows (records), and how the variables and rows should interact. Within a given parameter (PARAM), Basic Data Structure dataset rows can be categorized as described here.

For clarity, this paper names and defines certain categories of BDS rows. The terms "Observed row" and "Derived row", as well as the names of the types of "Derived" rows ("Derived Timepoint" and "Alternative Baseline") have specific meanings defined here for discussion purposes. To date, row categories are not defined with this level of specificity in CDISC documentation.

See also section 1, "Definitions of Observed and Derived."

Examples of these row categories are shown and discussed in section 7, "Examples of Row Categories and Selection Criteria."

### 4.2 Observed Rows within a Parameter

An observed row is one that is created from source data (SDTM and/or ADaM), rather than from other rows of the current parameter. A row can be highly derived from one or more records in one or more SDTM and/or ADaM source datasets, and yet still be called "observed." For example, even in a parameter that is derived from other ADaM parameters (so PARAMTYP could be set to "DERIVED"), there are rows that are "observed" in the sense that they were not derived from other rows within the current parameter.

### 4.3 Derived Rows within a Parameter

A derived row is one that is derived from other rows in the current parameter. At a high level, there are two reasons to create a derived row: to create an analysis value, or to make use of an alternative definition of baseline. Note that sometimes both reasons apply.

### 4.3.1 Derived Timepoint Rows

A row could be derived from other rows within the current parameter for the purpose of creating an analysis value, such as for imputing AVAL at a missing timepoint, or averaging values collected at the same visit, or to define a conceptual timepoint, such as capturing the maximum post-baseline value, or defining an endpoint value such as the average of the last two post-baseline values. The type of derivation is indicated in the variable DTYPE (e.g. DTYPE = "AVERAGE", DTYPE = "LOCF", etc.). See also Rule 3 in section 4.2 of the ADaMIG v1.0.

### 4.3.2 Alternative Baseline Rows

A row could be derived to make use of an alternative definition of baseline. When there is more than one definition of baseline, new baseline and post-baseline rows are created for each new definition of baseline (and the definition of baseline applicable to every observed and derived row in the parameter is indicated in BASETYPE). See also Rule 6 in section 4.2 of the ADaMIG v1.0.

## 4.4 Row Requirements

For each of the row categories, the following table indicates when DTYPE, BASETYPE and PARAMTYP should be populated. Further information is provided in the notes following the table.

ROW CATEGORY	DTYPE (Derivation Type)	BASETYPE (Baseline Type)	PARAMTYP (Parameter Type)
Observed Row	null	populated if more than one baseline type is defined for the parameter	populated with "DERIVED" if parameter is derived
Derived Timepoint Row (record created to impute missing timepoint or create a summary or conceptual timepoint)	populated	populated if more than one baseline type is defined for the parameter	populated with "DERIVED" if parameter is derived
Alternative Baseline Row (record created to use an alternative definition of baseline BASE)	populated if record is a derived timepoint record	populated	populated with "DERIVED" if parameter is derived

- DTYPE is populated if the analysis value (AVAL or AVALC) for the current row is derived from one or more rows within the current parameter. Otherwise it is left null on the current row.
- BASETYPE is populated if there is more than one definition of baseline for the current parameter. Otherwise it is left null. BASETYPE identifies the definition of baseline corresponding to BASE on the current row. If there is more than one definition of baseline in a parameter, BASETYPE must be populated on all rows for that parameter.
- PARAMTYP is set to "DERIVED" if the parameter is derived. Otherwise it is left null. PARAMTYP is constant across all rows of a given parameter.

## 5. Population Flags

In ADaM there are population flags, which can be subject-level, parameter-level and record-level. It may be the case that two or even all three flag types will be needed for a given "population."

### 5.1 Subject-Level Population Flags

Subject-level population flags are contained in ADSL, and can be copied to BDS datasets. Subject-level population flag variable names end in FL and values must be "Y" or "N", and not null. Most commonly, population flags are subject-level.

For example, if a subject is a member of the Intent-to-Treat population, then ITTFL = "Y". If the subject is not a member of the Intent-to-Treat population, then ITTFL = "N". ITTFL would be populated for each subject.

## 5.2 Parameter-Level Population Flags

Parameter-level population flag variable names end in PFL and values must be "Y" or null. A parameter-level flag is useful when observed data influence whether or not a subject is considered to be in a population for reporting purposes.

For example, for the analysis of a given parameter, a subject is included in the analysis if (1) the subject is a member of the ITT population, and (2) for the given parameter, the subject has a non-missing baseline value (BASE) and at least one non-missing postbaseline value of AVAL. For the subjects who meet both conditions, the parameter-level Full Analysis Set population flag FASPFL is set to "Y" on all rows for the parameter.

## 5.3 Record-Level Population Flags

Record-level population flag variable names end in RFL and values must be "Y" or null. A record-level flag is useful when there are reasons for excluding some records within subject and parameter from the analysis.

For example, if the definition of per-protocol includes having taken at least one tablet within the three days prior to a blood sample, then only those collection dates where the subject has met that criterion in addition to other possible record-level criteria will be flagged with PPROTRFL="Y".

## 6. Row Selection Criteria

Given that the BDS is a vertical structure, which includes all observed and derived rows for a parameter and not just those used for analysis, and can support multiple analyses, it is critically important that the correct record selection criteria are provided in analysis results metadata for any given analysis.

A selection statement (or SAS® WHERE statement or clause) must be able to specify exactly which rows are input to a given analysis. Selection criteria may include variables such as PARAM, analysis visit AVISIT, population flags, derived record type DTYPE, baseline type Basetype, etc.

### 6.1 Analysis Record Flag (ANLzzFL)

When there is a tie produced by the selection criteria, for example, when there are two records at a given AVISIT that satisfy all of the selection criteria and there are no other existing variables that can be incorporated into the selection criteria to break the tie, then the analysis record flag ANLzzFL is used to break the tie (ADaMIG v1.0 p.72). In this sense, ANLzzFL is the tie breaker of last resort, and can be added to the variables used in the record selection criteria.

For example:

- If relative day windowing is used to define some analysis visits AVISIT, and there are potentially multiple records within any given AVISIT that satisfy other selection criteria, the record whose relative day is closest to the ideal study day for that AVISIT (and if there is still a tie, the later record) could be flagged by ANL01FL=Y as the record to represent the AVISIT. If there is only one record, that record would be flagged.
- E.g., if there are potentially multiple records where PARAMCD='LDLCF' and AVISIT='WEEK 4', the record whose relative day is closest to the ideal study day for Week 4 (and if there is still a tie, the later record) could be flagged by ANL01FL=Y as the record to represent AVISIT=WEEK 4.
- Note that if for a given subject, there is only one record for PARAMCD='LDLCF' in the AVISIT='WEEK 4' relative day window, that record should still be flagged by ANL01FL=Y, so that it will be included in the analysis for WEEK 4.
- The resulting record selection criteria for an ITT analysis in this example would be ITTFL='Y' and PARAMCD='LDLCF' and AVISIT='WEEK 4' and ANL01FL='Y'.

ANLzzFL is illustrated in Tables 1-5 of section 7, "Examples of Row Categories and Selection Criteria."

Examples of row selection criteria for various analyses are provided in Table 6 of that section.

## 7. Examples of Row Categories and Selection Criteria

Section 7 illustrates the BDS row categories presented above in the context of an ADLB laboratory dataset supporting analyses that use two different definitions of baseline. Row categories are displayed as they occur during the setup for one subject of low density lipoprotein (LDL) cholesterol data. Due to space limitations, only some of the ADLB columns are shown as needed to illustrate concepts.

Table 1 represents observed data for plasma LDL cholesterol determined by the Friedewald method in S.I. units, where PARAMCD is defined as “LDLCF”. The corresponding PARAM, “PLASMA LDL Cholesterol Friedewald (mmol/L)”, is not shown.

The first step is to create the initial "observed" set of rows. These are copied or derived directly from the source data in SDTM and/or ADaM. Table 1 shows the observed data as arranged in the ADaM BDS dataset ADLB after pulling in from SDTM.

This particular lab test result is derived by the laboratory as a function of total cholesterol, HDL and triglycerides, converted to S.I. units, and placed into SDTM. However the ADaM parameter created to contain this data is not derived from other ADaM parameters. Therefore, the ADaM parameter is not considered to be a derived parameter, and PARAMTYP=null.

As these are initial records created per the definition of the parameter, they are Row Category "Observed" records.

**Table 1: Observed Rows for PARAM= “PLASMA LDL Cholesterol Friedewald (mmol/L)”**

Row	PARAMCD	ADT	VISIT	AVISIT	ABLFL	ANL01FL	AVAL	BASE	CHG	DTYPE	BASETYPE	PARAMTYP	Row Categories
1	LDLCF	01AUG2012	VISIT 1	SCREENING		Y	2.50	4.10					Observed
2	LDLCF	17AUG2012	VISIT 2	BASELINE			3.04	4.10					Observed
3	LDLCF	17AUG2012	VISIT 2	BASELINE	Y	Y	4.10	4.10					Observed
4	LDLCF	29AUG2012	VISIT 3	WEEK 2		Y	3.47	4.10	-0.63				Observed
5	LDLCF	14SEP2012	VISIT 4	WEEK 4		Y	4.25	4.10	0.15				Observed
6	LDLCF	26SEP2012	VISIT 6	WEEK 8		Y	5.10	4.10	1.00				Observed

### Table 1 Notes:

- There were two blood samples drawn at the baseline analysis visit (AVISIT = "BASELINE").
- ANL01FL is used to flag the record chosen (by an algorithm defined in metadata) to represent each analysis visit. It is used to break ties when selecting records for analysis.
- AVAL is described by PARAM and contains the numeric value.
- ABLFL flags the record whose AVAL is used to populate the Baseline column BASE. The definition that happened to have been used in this study was last observed value prior to treatment.
- CHG = AVAL - BASE for post-baseline records.
- There is no observed data for this subject for scheduled visit 5 (AVISIT = "WEEK 6").
- No records have been derived from the observed records so far, hence DTYPE is null.
- There is only one baseline definition so far, hence BASETYPE is null.
- This parameter was not derived from other parameters, so PARAMTYP = null.
- As these are initial records created per the definition of the parameter, they are Row Category "Observed" records.

**Table 2: Imputation of a Missing Timepoint**

Row	PARAMCD	ADT	VISIT	AVISIT	ABLFL	ANL01FL	AVAL	BASE	CHG	DTYPE	BASETYPE	PARAMTYP	Row Categories
1	LDLCF	01AUG2012	VISIT 1	SCREENING		Y	2.50	4.10					Observed
2	LDLCF	17AUG2012	VISIT 2	BASELINE			3.04	4.10					Observed
3	LDLCF	17AUG2012	VISIT 2	BASELINE	Y	Y	4.10	4.10					Observed
4	LDLCF	29AUG2012	VISIT 3	WEEK 2		Y	3.47	4.10	-0.63				Observed
5	LDLCF	14SEP2012	VISIT 4	WEEK 4		Y	4.25	4.10	0.15				Observed
7	<b>LDLCF</b>	<b>14SEP2012</b>	<b>VISIT 4</b>	<b>WEEK 6</b>		<b>Y</b>	<b>4.25</b>	<b>4.10</b>	<b>0.15</b>	<b>LOCF</b>			<b>Derived Timepoint</b>
6	LDLCF	26SEP2012	VISIT 6	WEEK 8		Y	5.10	4.10	1.00				Observed

**Table 2 Notes:**

- No data was available for Week 6. On row 7, the AVISIT="WEEK 6" analysis timepoint was imputed.
- Row 7 was derived by the method of Last Observation Carried Forward. DTYPE is set to "LOCF".
- DTYPE is subject to controlled terminology, extendable if necessary at the study level. The algorithms behind each DTYPE are defined in metadata.
- Row Category is "Derived Timepoint" for the derived record on row 7.
- ANL01FL is Y on row 7; this is to allow selection criteria for several kinds of analysis (observed data only and LOCF).
- When the SDTM traceability variables (e.g. SDTM VISIT or LBSEQ, or ADaM ADT) on all of the precursor record(s) for a derived row have the same value, the traceability variable values are carried into the derived record. In this case, the Week 6 LOCF record on row 7 came from VISIT 4 (Row 5).

**Table 3: Creation of Derived Summary Timepoint Rows**

Row	PARAMCD	ADT	VISIT	AVISIT	ABLFL	ANL01FL	AVAL	BASE	CHG	DTYPE	BASETYPE	PARAMTYP	Row Categories
1	LDLCF	01AUG2012	VISIT 1	SCREENING		Y	2.50	4.10					Observed
2	LDLCF	17AUG2012	VISIT 2	BASELINE			3.04	4.10					Observed
3	LDLCF	17AUG2012	VISIT 2	BASELINE	Y	Y	4.10	4.10					Observed
4	LDLCF	29AUG2012	VISIT 3	WEEK 2		Y	3.47	4.10	-0.63				Observed
5	LDLCF	14SEP2012	VISIT 4	WEEK 4		Y	4.25	4.10	0.15				Observed
7	LDLCF	14SEP2012	VISIT 4	WEEK 6		Y	4.25	4.10	0.15	LOCF			Derived Timepoint
6	LDLCF	26SEP2012	VISIT 6	WEEK 8		Y	5.10	4.10	1.00				Observed
<b>8</b>	<b>LDLCF</b>			<b>POST BASELINE MAX</b>		<b>Y</b>	<b>5.10</b>	<b>4.10</b>	<b>1.00</b>	<b>MAXIMUM</b>			<b>Derived Timepoint</b>

**Table 3 Notes:**

- Row 8 was derived to contain the maximum observed (DTYPE=null) post-baseline value. DTYPE is set to "MAXIMUM", and AVISIT is set to "POST BASELINE MAX."
- On row 8, note that for this subject, there is not a tie among post-baseline observed records for maximum value, so ADT and VISIT could have been carried forward from that record (row 6). If there is a tie, the first or any maximum record could be chosen from which to carry forward the traceability variables, or they can just be left blank. In general, for example of an average over multiple visits, or a median, there is no single source record for a summary record.

**Table 4: Creation of an Alternative Baseline**

Row	PARAMCD	ADT	VISIT	AVISIT	ABLFL	ANL01FL	AVAL	BASE	CHG	DTYPE	BASETYPE	PARAMTYP	Row Categories
1	LDLCF	01AUG2012	VISIT 1	SCREENING		Y	2.50	4.10			LAST		Observed
2	LDLCF	17AUG2012	VISIT 2	BASELINE			3.04	4.10			LAST		Observed
3	LDLCF	17AUG2012	VISIT 2	BASELINE	Y	Y	4.10	4.10			LAST		Observed
<b>9</b>	<b>LDLCF</b>	<b>17AUG2012</b>	<b>VISIT 2</b>	<b>AVERAGE BASELINE</b>	<b>Y</b>	<b>Y</b>	<b>3.57</b>	<b>3.57</b>		<b>AVERAGE</b>	<b>AVERAGE</b>		<b>Derived Timepoint</b>
4	LDLCF	29AUG2012	VISIT 3	WEEK 2		Y	3.47	4.10	-0.63		LAST		Observed
5	LDLCF	14SEP2012	VISIT 4	WEEK 4		Y	4.25	4.10	0.15		LAST		Observed
7	LDLCF	14SEP2012	VISIT 4	WEEK 6		Y	4.25	4.10	0.15	LOCF	LAST		Derived Timepoint
6	LDLCF	26SEP2012	VISIT 6	WEEK 8		Y	5.10	4.10	1.00		LAST		Observed
8	LDLCF			POST BASELINE MAX		Y	5.10	4.10	1.00	MAXIMUM	LAST		Derived Timepoint

**Table 4 Notes:**

- Row 9 was derived by taking the average of the values observed at the SDTM scheduled baseline visit (VISIT=VISIT 2). DTYPE has been set to "AVERAGE" to indicate that the record was derived and to point to the algorithm used to derive the record.
- On row 9, AVISIT has been set to "AVERAGE BASELINE" to characterize this derived summary timepoint. Alternatively the AVISIT for row 9 could have been "BASELINE", since DTYPE and BASETYPE are enough to distinguish this derived record.
- This second definition of baseline means that there is now more than one definition of baseline. Thus, the BASETYPE column must be filled in for all records for this parameter. The BASETYPE corresponding to row 9 has been set to AVERAGE.
- The BASETYPE corresponding to the definition of baseline (last value observed prior to treatment) used in rows 1-8 has been called "LAST".
- Note that there must be exactly as many values of BASETYPE as there are records flagged with ABLFL="Y", and vice-versa.

**Table 5: Creation of Records to Make Use of an Alternative Baseline**

Row	PARAMCD	ADT	VISIT	AVISIT	ABLFL	ANL01FL	AVAL	BASE	CHG	DTYPE	BASETYPE	PARAMTYP	Row Categories
1	LDLCF	01AUG2012	VISIT 1	SCREENING		Y	2.50	4.10			LAST		Observed
2	LDLCF	17AUG2012	VISIT 2	BASELINE			3.04	4.10			LAST		Observed
3	LDLCF	17AUG2012	VISIT 2	BASELINE	Y	Y	4.10	4.10			LAST		Observed
9	LDLCF	17AUG2012	VISIT 2	AVERAGE BASELINE	Y	Y	3.57	3.57		AVERAGE	AVERAGE		Derived Timepoint
4	LDLCF	29AUG2012	VISIT 3	WEEK 2		Y	3.47	4.10	-0.63		LAST		Observed
5	LDLCF	14SEP2012	VISIT 4	WEEK 4		Y	4.25	4.10	0.15		LAST		Observed
7	LDLCF	14SEP2012	VISIT 4	WEEK 6		Y	4.25	4.10	0.15	LOCF	LAST		Derived Timepoint
6	LDLCF	26SEP2012	VISIT 6	WEEK 8		Y	5.10	4.10	1.00		LAST		Observed
8	LDLCF			POST BASELINE MAX		Y	5.10	4.10	1.00	MAXIMUM	LAST		Derived Timepoint
10	LDLCF	29AUG2012	VISIT 3	WEEK 2		Y	3.47	3.57	-0.10		AVERAGE		Alternative Baseline
11	LDLCF	14SEP2012	VISIT 4	WEEK 4		Y	4.25	3.57	0.68		AVERAGE		Alternative Baseline
12	LDLCF	14SEP2012	VISIT 4	WEEK 6		Y	4.25	3.57	0.68	LOCF	AVERAGE		Derived Timepoint, Alternative Baseline
13	LDLCF	26SEP2012	VISIT 6	WEEK 8		Y	5.10	3.57	1.53		AVERAGE		Alternative Baseline
14	LDLCF			POST BASELINE MAX		Y	5.10	3.57	1.53	MAXIMUM	AVERAGE		Derived Timepoint, Alternative Baseline

**Table 5 Notes:**

- Rows 10-14 were added to make use of the AVERAGE baseline definition.
- If a similar parameter were derived from "PLASMA LDL Cholesterol Friedewald (mmol/L)" to support analysis in conventional units, it could be called "PLASMA LDL Cholesterol Friedewald (mg/dL)", and Tables 1-5 would be built up the same except that PARAMCD would be different, e.g., "LDLCFC", and PARAMTYP would be "DERIVED" instead of null, and AVAL, BASE, and CHG would be expressed in mg/dL instead of mmol/L.

**Table 6: Examples of Record Selection Criteria**

Analysis	Record Selection Criteria
Data as Observed analysis of change from baseline at Week 6, where baseline is defined as the last value observed prior to treatment.	AVISIT='WEEK 6' and DTYPE='' and BASETYPE='LAST' and ANL01FL='Y'
Last Observation Carried Forward analysis of change from baseline at Week 6, where baseline is defined as the last value observed prior to treatment.	AVISIT='WEEK 6' and (DTYPE='' or DTYPE='LOCF') and BASETYPE='LAST' and ANL01FL='Y'
Analysis of change from baseline to maximum post-baseline value, where baseline is defined as the last value observed prior to treatment.	AVISIT='POST BASELINE MAX' and DTYPE='MAXIMUM' and BASETYPE='LAST'
Data as Observed analysis of change from baseline at Week 6, where baseline is defined as the average of the values collected at the baseline visit.	AVISIT='WEEK 6' and DTYPE='' and BASETYPE='AVERAGE' and ANL01FL='Y'
Last Observation Carried Forward analysis of change from baseline at Week 6, where baseline is defined as the average of the values collected at the baseline visit.	AVISIT='WEEK 6' and (DTYPE='' or DTYPE='LOCF') and BASETYPE='AVERAGE' and ANL01FL='Y'
Analysis of change from baseline to maximum post-baseline value, where baseline is defined as the average of the values collected at the baseline visit.	AVISIT='POST BASELINE MAX' and DTYPE='MAXIMUM' and BASETYPE='AVERAGE'

**Table 6 Notes:**

- It is critical to specify record selection criteria correctly.
- Population is ignored here for brevity, but typically, population flags would also be included in record selection criteria (e.g. "and ITTFL='Y'").

## CONCLUSION

The authors hope that this paper helps beginners to understand how the structurally important parts of the ADaM BDS dataset function together. In addition, the authors hope that the paper stimulates further discussion and clarification for the benefit of practitioners and the standard itself.

## REFERENCES

CDISC Analysis Data Model (ADaM) Implementation Guide, Version 1.0. 2009.

## ACKNOWLEDGMENTS

The authors of this paper would like to thank Shelley Dunn, Tara Erb, Isaac Swanson, and Richann Watson for their input.

## CONTACT INFORMATION

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

Michelle Barrick  
[mbarrick@lilly.com](mailto:mbarrick@lilly.com)

John Troxell  
[jktroxell@gmail.com](mailto:jktroxell@gmail.com)

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.