

When Should I Break ADaM Rules?

Sandra Minjoe, PRA Health Sciences

ABSTRACT

The ADaM Model document (ADaM v2.1), Implementation Guides (ADaMIG v1.x) and Occurrence Data Structure document (OCCDS v1.0) include many rules and requirements for a general ADaM dataset and for an ADaM dataset of class Basic Data Structure (BDS). We are often able to follow all of these rules, but it is not uncommon to run into issues here and there. This paper describes a few of the most common issues you might find when trying to design an ADaM BDS dataset, and offers recommendations on how to decide when to break the rules.

AUTHOR BACKGROUND

I've been on the CDISC ADaM team since 2001. I have led and co-led sub-teams, including those that wrote ADaMIG v1.1 ([Reference 1](#)) and OCCDS v1.0 ([Reference 6](#)). I joined the ADaM Leadership team in 2014, and was the ADaM Team Lead for the years 2018 and 2019. As an active participant on the ADaM team, I review all ADaM documents for adherence to the various ADaM rules. While I can't officially speak for the ADaM team, I am very familiar with the content and intent of all the ADaM documents.

INTRODUCTION

The ADaM model v2.1 ([Reference 1](#)) lists in section 3.1 the following fundamental principles, all of which are needed in order to call a dataset ADaM:

- Analysis datasets and their associated metadata must facilitate clear and unambiguous communication.
- Analysis datasets and their associated metadata must provide traceability between the analysis data and its source data (ultimately SDTM).
- Analysis datasets and their associated metadata must be readily useable by commonly available software tools.
- Analysis datasets must be accompanied by metadata.
- Analysis datasets must be analysis-ready.

As described in Section 1.6 of both ADaMIG v1.1 ([Reference 4](#)) and ADaMIG v1.2 ([Reference 5](#)), a dataset can be ADaM even if it is not in one of the standard ADaM structures: Subject Level Analysis Dataset (ADSL), Basic Data Structure (BDS), or Occurrence Data Structure (OCCDS). This means that a BDS or OCCDS dataset must first and foremost be an ADaM dataset, and follow all the ADaM fundamental principles.

An ADaM BDS dataset is designed to be analysis-ready for many for analyses based on parameter with or without analysis timepoints, and an ADaM OCCDS dataset is designed to be analysis-ready for occurrence analysis (such as is typical for adverse event and concomitant medications summaries).

It is worth noting that while many of the rules for dataset structure are documented in the ADaM Conformance Rules ([Reference 2](#)), others, like the fundamental principles of ADaM, are too difficult to programmatically check. Just because a fundamental principle or BDS structure rule isn't found in the ADaM Conformance Rules doesn't mean that it can be ignored!

EXAMPLE 1: MAKING PARAM UNIQUE

As described in Section 3.3.4 of ADaMIG v1.1 ([Reference 4](#)), "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."

When bringing data from an SDTM findings dataset into an ADaM dataset, the contents of PARAM often include things like the SDTM test name, units, and location: in other words, the topic variable plus all necessary qualifiers used to make the parameter unique. This allows us to have a different parameter for urine glucose and serum glucose, as shown in Table 1, since these would be analyzed separately.

Table 1: Unique Lab PARAMs

| LBTEST | LBCAT | LBSTRESU | LBSPEC | LBMETHOD | PARAM |
|---------|------------|----------|--------|------------|--|
| Glucose | CHEMISTRY | mg/dL | SERUM | | Serum Glucose (mg/dL) |
| Glucose | URINALYSIS | mg/dL | URINE | TEST STRIP | Urine Glucose Test Strip (mg/dL) |
| Glucose | URINALYSIS | | URINE | TEST STRIP | Urine Glucose Test Strip - Categorical |

In fact, a good way to think about what to put into PARAM is that any two values with the same value of PARAM would be reasonably analyzed together.

PARCATy

Sometimes people try to use variable PARCATy to “qualify” PARAM, but this is not allowed. Notice in Table 2 that some of the information needed to make PARAM unique is instead contained in PARCAT1.

Table 2: Incorrectly Assigned PARAM and PARCAT1

| LBTEST | LBCAT | LBSTRESU | LBSPEC | LBMETHOD | PARCAT1 | PARAM |
|---------|------------|----------|--------|------------|------------------|-----------------|
| Glucose | CHEMISTRY | mg/dL | SERUM | | Serum | Glucose (mg/dL) |
| Glucose | URINALYSIS | mg/dL | URINE | TEST STRIP | Urine Test Strip | Glucose (mg/dL) |
| Glucose | URINALYSIS | | URINE | TEST STRIP | Urine Test Strip | Glucose |

The ADaM Conformance Rules v3.0 ([Reference 2](#)) includes check #124 with failure criteria: “There is more than one value of PARCATy for a given value of PARAMCD.” This means if you were to pass the data from Table 2 through a validation tool such as Pinnacle 21 ([Reference 8](#)), it would generate an error message. The point of variable PARAM is to hold all of the information that describes AVAL/AVALC, so information like Serum, Urine, and Test Strip, which help make a parameter unique, must be part of the contents of PARAM.

PARQUAL

There are a couple Therapeutic Area User Guides (TAUGs) that have added a provisional variable PARQUAL (Parameter Qualifier). This variable splits out exactly one qualifier variable from the contents of PARAM in order to make analysis simpler. The Breast Cancer TAUG ([Reference 9](#)) and Prostate Cancer TAUG ([Reference 10](#)) both used PARQUAL to hold evaluator information, such as might be collected in SDTM variable TUEVAL. Table 3 contains a few rows and some pertinent variables from the relevant Prostate Cancer TAUG dataset that show how PARQUAL is used.

Table 3: PARQUAL Example from Prostate TAUG

| USUBJID | PARAM | PARQUAL | PARAMCD |
|-------------|------------------------------------|---------------------|---------|
| ABC-123-001 | Progression-free survival (months) | INDEPENDENT REVIEW | PFS |
| ABC-123-001 | Progression-free survival (months) | INVESTIGATOR REVIEW | PFS |
| ABC-123-002 | Progression-free survival (months) | INDEPENDENT REVIEW | PFS |
| ABC-123-002 | Progression-free survival (months) | INVESTIGATOR REVIEW | PFS |

Variable PARQUAL is included in both of these TAUGs as “provisional”, since it is not part of any ADaM standard. This means that, in these TAUGs, it is the combination of PARAM and PARQUAL that make the parameter unambiguous and unique. You would never analyze progression-free survival result as evaluated by the investigator together with the progression-free survival result as evaluated by the independent reviewer. If you choose to follow the TAUG and use variable PARQUAL, you are technically breaking a BDS rule.

It would be perfectly legal to insert the contents of PARQUAL into the variable PARAM in order to make it BDS-compliant, as shown in Table 4. Here the PARAMCD value is also modified in order to make it a 1:1 map with PARAM.

Table 4: PARQUAL Data from Prostate TAUG Combined into PARAM

| USUBJID | PARAM | PARAMCD |
|-------------|---|---------|
| ABC-123-001 | Progression-free survival (months) by Independent Review | PFSIND |
| ABC-123-001 | Progression-free survival (months) by Investigator Review | PFSINV |
| ABC-123-002 | Progression-free survival (months) by Independent Review | PFSIND |
| ABC-123-002 | Progression-free survival (months) by Investigator Review | PFSINV |

Note that if you follow the TAUG and create a PARQUAL variable, you probably will not get a Pinnacle 21 ([Reference 8](#)) error message. This is because an automated tool is unable to determine whether rows with the same PARAM value would be analyzed together. However, just because you don’t get an error from an automated tool, doesn’t mean that your dataset is compliant with all ADaM requirements!

EXAMPLE 2: NADIR

As described in Section 4.2 of ADaMIG v1.1 ([Reference 4](#)), “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 rest of the rows vs. column rules all result in adding new rows, meaning that anything other than a parameter-invariant function of AVAL and BASE must be added as a new row.

How do we handle derivations involving Nadir (the lowest observed value for the subject)? Well, there are a couple schools of thought:

1. Treat Nadir like a Baseline. Set ABLFL to “Y” on the nadir row, and that value is used for derivations like change from Nadir. This option is BDS-compliant but can be cumbersome to use.
2. Add a set of columns for Nadir, Change from Nadir, etc., similar to BDS standard variables Baseline, Change from Baseline. This option is not BDS-compliant but is probably the easiest to review.

Let’s look at each option above using the data from Table 5, which are all the values for one parameter for one subject, and discuss their pros and cons.

Table 5: Nadir Example Input Data

| --SEQ | AVISIT | AVAL |
|-------|---------------|------|
| 101 | Pre-Treatment | 25 |
| 102 | Week 1 | 23 |
| 103 | Week 2 | 21 |
| 104 | Week 3 | 24 |

Treat Nadir like a Baseline

In Table 5, Nadir would be from row 3 (value 21 is the smallest value). If we call that Nadir value the baseline row, then we can use the standard variables BASE and CHG to derive change from Nadir, as shown in Table 6.

Table 6: Analysis of Nadir Instead of Baseline

| --SEQ | AVISIT | AVAL | ABLFL | BASE | CHG |
|-------|---------------|------|-------|------|-----|
| 101 | Pre-Treatment | 25 | | 21 | 4 |
| 102 | Week 1 | 23 | | 21 | 2 |
| 103 | Week 2 | 21 | Y | 21 | 0 |
| 104 | Week 3 | 24 | | 21 | 3 |

However, this simple solution works only when there is not a need for a standard change from baseline. If you also need the standard change from baseline plus change from Nadir, you would either need to use BASETYPE to distinguish between the different Baselines, as shown in Table 7, or split this into two different datasets, such as in Table 6 and Table 8.

Table 7: Analysis of Nadir Plus Baseline in One Dataset

| --SEQ | AVISIT | AVAL | ABLFL | BASE | CHG | BASETYPE |
|-------|---------------|------|-------|------|-----|-----------|
| 101 | Pre-Treatment | 25 | Y | 25 | 0 | PRE-TREAT |
| 102 | Week 1 | 23 | | 25 | -2 | PRE-TREAT |
| 103 | Week 2 | 21 | | 25 | -4 | PRE-TREAT |
| 104 | Week 3 | 24 | | 25 | -1 | PRE-TREAT |
| 101 | Pre-Treatment | 25 | | 21 | 4 | NADIR |
| 102 | Week 1 | 23 | | 21 | 2 | NADIR |
| 103 | Week 2 | 21 | Y | 21 | 0 | NADIR |
| 104 | Week 3 | 24 | | 21 | 3 | NADIR |

Table 8: Analysis of Baseline (no Nadir)

| --SEQ | AVISIT | AVAL | ABLFL | BASE | CHG |
|-------|---------------|------|-------|------|-----|
| 101 | Pre-Treatment | 25 | Y | 25 | 0 |
| 102 | Week 1 | 23 | | 25 | -2 |
| 103 | Week 2 | 21 | | 25 | -4 |
| 104 | Week 3 | 24 | | 25 | -1 |

Notice that Table 6, which has only Nadir content (no Baseline analysis) and Table 8, which has only Baseline content (no Nadir analysis), the variable BASETYPE is not used. Instead, dataset-level metadata would explain which dataset is used for Baseline or Nadir analysis.

In Table 7, data for both analyses are shown, with one rows 1-4 used for the standard Baseline analysis, and rows 5-8 used for Nadir analysis. However, this means it basically duplicates all the input rows, and can be confusing for a reviewer. This brings us to the second Nadir option, adding columns.

Create Columns for Nadir Analysis

Ignoring BDS rules for a minute, wouldn't it be helpful to have Nadir information in columns, such as shown in Table 9?

Table 9: Nadir Content in Columns

| --SEQ | AVISIT | AVAL | ABLFL | BASE | CHG | NADIRFL | NADIR | NADRCHG |
|-------|---------------|------|-------|------|-----|---------|-------|---------|
| 101 | Pre-Treatment | 25 | Y | 25 | 0 | | 21 | 4 |
| 102 | Week 1 | 23 | | 25 | -2 | | 21 | 2 |
| 103 | Week 2 | 21 | | 25 | -4 | Y | 21 | 0 |
| 104 | Week 3 | 24 | | 25 | -1 | | 21 | 3 |

If this type of dataset is analysis-ready and reviewer-friendly, might it be worth “breaking” the BDS rule? Here you have to weigh the fundamental principle of being analysis-ready vs. the BDS rule of when to add rows vs. columns.

If you decide to go this route, your next decision needs to be whether to call the dataset BDS or ADAM OTHER. Calling it BDS means that automated tools designed for BDS will work with it. I would recommend that at the very least you call it BDS when running any automated checks, so that those checks can look for any other potential issues within the dataset.

EXAMPLE 3: AVALC FOR LISTINGS

Variables AVAL and AVALC are described in Section 3.3.4 CDISC Notes of ADaMIG v1.1 ([Reference 4](#)).

Table 10: Variable Descriptions for AVAL and AVALC

| BDS Variable | Variable description from CDISC Notes |
|--------------|--|
| AVAL | Numeric analysis value described by PARAM. |
| AVALC | Character analysis value described by PARAM. |

The CDISC Notes for AVALC also states:

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.

Later in this same section, additional text further describes how to handle numeric AVAL and character 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.

In other words, do not simply dump --STRESC into AVALC and --STRESN into AVAL, regardless of parameter. Instead, determine, by parameter, how you will analyze AVAL and/or AVALC, and whether you will retain --STRESC for listings or traceability. Commonly, your derivations might look something like:

- For all parameters analyzed as numeric, populate AVAL. You might need parameter-specific imputation rules when source data isn't numeric, such as when --STRESC contains a value like "BLQ" or "<2".
- For all parameter analyzed as categorical, populate AVALC. You may also consider populating AVAL in order to sort AVALC for table display. This is common with questionnaires and other scales where the text strings in AVALC need to be presented in some sort of order (such as "Low", "Medium", and "High") which is not alphabetical.

One reason I've been given for populating AVALC, even on parameters analyzed as numeric, is so that all content is in one variable for the production of listings. Variable --STRESC works for this need when all rows have a value, but not in other cases (such as when ADaM parameter BMI is derived from height and weight and doesn't exist in the SDTM data). However, AVAL and AVALC are analysis values, and listings

are not analyses! If you want to include a variable like this, solely for use only in listings, you are allowed to create your own variable, such as LISTVAL.

Unlike the first 2 examples, there is no need to break a BDS rule here. Instead of misusing AVALC, simply add a variable that will make listing generation more straightforward.

EXAMPLE 4: CREATING A VARIABLE FROM SUPPQUAL CONTENT

As described in Section 3.2 of OCCDS v1.0 ([Reference 6](#)), “For traceability when copying variables from SUPPQUAL, it is recommended to use variable names that exactly match the corresponding SUPPQUAL.QNAM values.”

In most cases when bringing in content from SUPPQUAL, using SUPPQUAL QNAM as the variable name is a simple way to facilitate traceability. Exceptions to consider are when:

1. The QNAM is a standard ADaM variable name and means something else (for example, when there is a QNAM of TRTP)
2. The QNAM ends in a standard ADaM suffix that means something else (for example, a suffix of DT, where there could not possibly be a numeric date in SUPPQUAL) and would be confusing.
 - o Note that standard ADaM suffix values are listed in tables in ADaMIG v1.1 ([Reference 4](#)) Section 3.1.
3. The QNAM is for a concept that would make it subject to a fragment rule used for creating your own ADaM variable name (for example, a flag with values Y/N that doesn't have a QNAM suffix of “FL”)

In these cases, you will need weigh the ADaM fundamental principle of traceability against the ADaM naming rules, and determine which to follow.

My personal recommendation is that when the QNAM is a standard ADaM variable or ends in a standard ADaM fragment (numbers 1 and 2 above), then follow the ADaM naming rules, assign a different variable name, and rely on metadata to provide traceability. This will avoid overlap with any ADaM variables of the same name used in the study, and prevent confusion on seeing a standard ADaM variable name or fragment being used for a non-standard purpose.

However, my recommendation for when the variable concept would make it subject to a fragment rule for creating your own ADaM variable name (number 3 above), is to use the QNAM as the ADaM variable name. There is precedence for this, since some variables copied from SDTM do not follow the ADaM content rules (for example, variable AESER is a Y/N flag but doesn't end in FL), yet we don't change those variable names when copied to ADaM.

If you choose traceability over naming rules, you will want to address any Pinnacle-21 or other validation findings with text such as “SDTM QNAM value was used as the variable name in order to provide traceability.”

EXAMPLE 5: INTERMEDIATE DATASET PRIOR TO TIME-TO-EVENT

When the first ADaMIG v1.0 ([Reference 3](#)) was published in 2009, the ADaM team included a couple time-to-event variables in the BDS section. In 2012, the ADaM Time-to-Event document was released, with more variables and examples, and it suggested that an intermediate dataset might be helpful for analysis.

In 2016, the Therapeutic Area User Guide (TAUG) for Breast Cancer included an example with an intermediate dataset designed as BDS, as shown in Table 11 (not all columns shown). Then in 2017, the TAUG for Prostate Cancer included an example of an intermediate dataset that was not BDS, as shown in Table 12 (not all columns shown).

Table 11: Intermediate Dataset from Breast Cancer TAUG

| ASEQ | ASTDT | ASTDY | PARQUAL | PARAMCD | AVALC | ANL01FL |
|------|-----------|-------|--------------|----------|------------|---------|
| 1 | 2013DEC29 | -4 | PROTOCOL | DISPOSIT | RANDOMIZED | |
| 2 | 2013DEC30 | -2 | INVESTIGATOR | ASSESS | PD | Y |
| 3 | 2013DEC31 | -1 | CENTRAL | ASSESS | SD | Y |
| 4 | 2014JAN01 | 1 | PROTOCOL | DISPOSIT | TREATMENT | Y |
| 5 | 2014JAN21 | 20 | INVESTIGATOR | ASSESS | SD | Y |
| 6 | 2014JAN22 | 22 | CENTRAL | ASSESS | SD | Y |

Table 12: Intermediate Dataset from Prostate Cancer TAUG

| ASEQ | ADT | ADTDESC | ADTDESCD | ADY |
|------|-----------|--|----------|-----|
| 1 | 03MAR2014 | Date of Randomization | RANDDT | 1 |
| 2 | 15OCT2014 | Change in Anti-Cancer Therapy | RXCHGDT | 227 |
| 3 | 15SEP2014 | Date of Last Tumor Assessment with No PD | LNOPDDT | 197 |
| 4 | 03DEC2014 | Date Last Known Alive | LSTALVDT | 276 |
| 5 | 01NOV2014 | Date of Analysis Cut-off | CUTOFFDT | 244 |

Each of the TAUG intermediate datasets uses BDS variable ASEQ along with some sort of BDS study date and study day. The Breast Cancer TAUG also uses BDS variables PARAM (not shown), PARAMCD, and AVALC, but the Prostate Cancer TAUG does not.

After the release of the Breast Cancer TAUG, the ADaM team had a lengthy discussion about whether BDS is the best option for this intermediate dataset, since:

1. This intermediate dataset is not being analyzed directly. Its sole purpose is to collect information to make the derivation of the time-to-event parameters more straightforward and traceable. Since it isn't being used for analysis, it doesn't necessarily need to be in the BDS structure.
2. PARAM and PARAMCD are used to describe AVAL/AVALC, but really the variable we want to describe is the analysis date. If we don't need the dataset in a BDS structure, then we don't need to force content into required BDS variables PARAM, PARAMCD, and AVAL/AVALC.

The Prostate Cancer TAUG, the most recent TAUG to use an intermediate dataset prior to a time-to-event dataset, used variable ADTDESC (Analysis Date Description) to describe the date. Variable ADTDESCD serves a similar purpose to PARAMCD, as the short code version of ADTDESC. Variables PARAM and PARAMCD are not included in the Prostate Cancer TAUG's intermediate dataset, so it is definitely not BDS.

Making use of an intermediate dataset certainly helps with the fundamental principle of traceability, and both of these TAUGs meet that principle. However, if the intermediate dataset itself isn't being analyzed, then there doesn't seem to be a need to force it into a BDS structure. Use any standard variable names, such as ASEQ and ADT, whenever the concept applies, but it is perfectly fine to call the dataset structure ADAM OTHER.

Wrapping up this discussion, it is worth a reminder that these two TAUGs are simply examples. They are not ADaM requirements, and you should not feel compelled to follow them if your situation is a little different. However, they do make a nice starting point!

CONCLUSION

The ADaM fundamental principles of providing clear and unambiguous communication, providing traceability between the analysis data and its source data, and being directly analyzable may sometimes be at odds with BDS and OCCDS rules. We saw some examples where following BDS or OCCDS rules would cause you to break a fundamental principle. In these cases, it seems far more important to follow

the fundamental principles and ensure the dataset is actually ADaM, than to ensure it is of standard structure BDS or OCCDS.

When you must break a BDS or OCCDS rule, I recommend the following:

1. Follow as much of ADaM as possible, especially standard naming conventions.
2. Determine whether to call your dataset BDS, OCCDS, or ADAM OTHER. A good rule of thumb (but certainly not a requirement) might be to first try calling it BDS or OCCDS, then run validation rules from a tool such as Pinnacle 21 (*Reference 8*) and review the results.
 - If many checks fire, then call it ADAM OTHER.
 - If not, call it a standard structure (BDS or OCCDS, whichever was used to run the validation rules).
3. Regardless of the decision made in Step 2, describe in the Analysis Data Reviewer's Guide (ADRG) how your dataset differs from the standard ADaM structure.

REFERENCES

1. ADaM v2.1: CDISC. 2009. Accessed March 29, 2021. <https://www.cdisc.org/standards/foundational/adam/adam-v2-1>.
2. ADaM Conformance Rules v3.0: CDISC. 2020. <https://www.cdisc.org/standards/foundational/adam/adam-conformance-rules-v3-0>.
3. ADaMIG v1.0: CDISC. 2009. Accessed March 29, 2021. <https://www.cdisc.org/standards/foundational/adam/adamig-v1-0>.
4. ADaMIG v1.1: CDISC. 2016. Accessed March 29, 2021. <https://www.cdisc.org/standards/foundational/adam/adamig-v1-1>.
5. ADaMIG v1.2: CDISC. 2019. Accessed March 29, 2021. <https://www.cdisc.org/standards/foundational/adam/adamig-v12>.
6. ADaM BDS for Time-to-Event Analysis. 2012. Accessed March 31, 2021. <https://www.cdisc.org/standards/foundational/adam/adam-basic-data-structure-bds-time-event-tte-analyses-v1-0>.
7. OCCDS v1.0: CDISC. 2016. Accessed March 29, 2021. <https://www.cdisc.org/standards/foundational/adam/adam-structure-occurrence-data-occds-v1-0>.
8. Pinnacle 21 Validator: Pinnacle 21. Accessed March 30, 2021. <https://www.pinnacle21.com/products/validation>.
9. Therapeutic Area User Guide for Breast Cancer: CDISC. 2016. <https://www.cdisc.org/standards/therapeutic-areas/breast-cancer>.
10. Therapeutic Area User Guide for Prostate Cancer: CDISC. 2017. <https://www.cdisc.org/standards/therapeutic-areas/prostate-cancer>.

CONTACT INFORMATION

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

Sandra Minjoe
MinjoeSandra@prahs.com

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