

Avoiding Sinkholes: Common Mistakes During ADaM Data set Implementation

Richann Watson, Experis; Karl Miller, inVentiv Health

ABSTRACT

The ADaM Implementation Guide was created in order to help maintain a consistency for the development of analysis data sets in the pharmaceutical industry. However, since its inception we have seen issues with guideline non-conformance which can impede this development process and carry impacts that are felt down-stream in subsequent processes. When working with ADaM data sets, non-compliance and other issues are likely the number one source for numerous hours of re-work. Not only creating unnecessary additional work for the data sets themselves, but also for reports, compliance checks, the Analysis Data Reviewers Guide (ADRG), etc. all the way down to the ISS/ISE processes. Considering this breadth of impact, one can see how devastating these sinkholes can be. Like any sinkhole, there is a way out of it but it is a long, tedious process that will consume a lot of resources and it is always better to avoid the sinkhole entirely. This paper will assist you in creating compliant ADaM data sets, provide the reasoning on why you should avoid these sinkholes, all of which will help minimize re-work and likely eliminate the need for additional work.

OUTLINE

1. Creating an ADaM data set just for the creation of listing. There is no need to create an ADaM data set for listings. Listings can be based off of SDTM domains. There is no need to have a one to one relation for SDTM to ADaM. ADaM data sets are determined based on analysis needs so it is possible for multiple SDTM domains to feed into one ADaM data set or for an ADaM data set to be created from other ADaM data sets.
2. ADaM data sets are not SDTM +. Just because the variable PARAMCD/PARAM and AVAL are put in a data set does not make it a BDS structure. You cannot just join a parent SDTM domain with the supplemental qualifier and change a variable name to PARAMCD, PARAM and AVAL or just add those variables and call it BDS. There are specific rules that govern the creation of a BDS structure.
3. Creating variables instead of new parameters. There are specific rules governing when a variable can be created. If variable does not adhere to those rules and falls in one of the other rules that indicate it should be a new row, then it should be created as a new timepoint or a new parameter. For example, don't create a new variable that converts AVAL and BASE to a different unit and then use the new variables to create another variable that calculates the CHG.
4. Adding timed baseline values into ADSL. Need to avoid circular reference. There is no rule that states all the results on a table must come from the same data set. If there is no need for baseline values other than to summarize baseline characteristics (e.g., baseline weight, baseline height), then do not add to ADSL. It is fine for the table to reference both ADSL and ADVS. Avoid the extra work. Reasoning for not adding is that baseline values will be based on the record where ABLFL = 'Y' so ADVS would need to be created prior to ADSL, but to create ADVS you would need to have ADSL to get the start of treatment (TRTSDT) to determine baseline. However, if these baseline values are needed for other types of analyses (i.e., covariates, subgrouping), then they can be added to ADSL but they would need to be based off of SDTM.VS and cross checks would need to be put into place to verify that the baseline values in ADSL match the baseline values in ADVS.
5. In BDS, AVAL and AVALC are both populated but there is not a 1-1 match within a PARAMCD. If both variables are populated then there should be a 1-1 match within the parameter. There is no requirement that they both be populated if they are in the data set. It is fine for either AVAL to be populated or AVALC to be populated. If the result contains a BQL or AQL and a formula is

implemented to convert these values to a numeric value, then AVALC should not be populated with the BQL/AQL result. Instead –STRESC should be brought over to show data point traceability and AVALC is left null so that AVAL can be populated using the formula. Reason for this is that if AVALC is populated with '<xx' or '>xx' and AVAL is some numeric value, then there could be another record within that parameter that has the same numeric value and AVALC would be that numeric value thus breaking the 1-1 match rule. Typically either one or the other variable is populated and not both. So if parameter is supposed to be a numeric value populate AVAL and retain –STRESC to keep the character version. If the parameter is qualitative then populate AVALC. The only time AVAL should be populated for a qualitative example is when trying to order the values. For example, URBC can have values of:

AVALC
1-5
6-9
10-15
TNTC

AVALC (ordered)
1-5
10-15
6-9
TNTC

AVAL	AVALC
1	1-5
2	6-9
3	10-15
4	TNTC

6. Partial dates and times imputed need to keep the imputation flag to indicate level of imputation. This shows the value was imputed and will not be found in the SDTM domain. Otherwise reviewer may look for that specific date or time and not find it.
7. Adverse events that go across multiple periods in a cross-over study that need to be analyzed in every period that the AE was ongoing (not just when it became TEAE). Seen people add new variables to account for all the different periods which can make for very confusing. Record can be split to count in different periods. See PharmaSUG paper from 2015.
8. Difference between PARAMTYP and DTYPE. PARAMTYP is used to indicate a parameter did **not previously exist**. PARAMTYP can **only** take on the value of 'DERIVED'. DTYPE is to indicate that a specific derivation was used to create that particular record. DTYPE can be used in conjunction with PARAMTYP if a new record using the indicated algorithm needs to be populated. Need example to illustrate the difference.
9. SMQ and CQ – trying to populate more than one SMQ/CQ in the same variable. Only one SMQ/CQ per set of variables. See PharmaSUG paper from 2017.
10. PARAM should fully describe AVAL. That means it needs to include units, location, specimen, fasting status, methodology or any other type of qualifier that is used to distinguish it from a similar test. For example, Glucose does not tell us if this is Serum, Plasma or Urine glucose. Does not tell us if this is mg/dL or mmol/L. Does not tell us if this is Fasting or not. To fully describe AVAL, PARAM should be something like 'Serum Fasting Glucose (mg/dL)'.
11. Improper use of parameter-level and record-level flags. Parameter-level flags identify a parameter that are typically included in the populate analysis but is to be excluded for a particular type of analyses or for a particular subject within the population. Exclusion could be due to wanting to keep only parameters that align with the baseline record (e.g., subject has vitals taken and can be in different positions but only want to keep the parameters that match the baseline position, so all other vital parameters not matching baseline are excluded for that subject). Record-level flags identify a specific record that would typically be eligible for assessment with the population but for some reason it is being excluded. Exclusion could be due to the record occurring after a specific event.
12. There should be only one baseline record per parameter, per baseline definition, per subject. If analysis calls for multiple baseline definitions then either split the data so that each data set contains a different baseline definition (name data set name and label so it is clear it pertaining to a specific baseline definition) or create a new set of records and use Basetype to identify the records for each baseline definition.

13. Improper use of ANLzzFL, CRITy/CRITyFL, AVALCATy. If sufficient data to select correct record don't add new variables just to add variables. CRITy/CRITyFL should be based on values on the same record. It should not be looking at other records to determine CRITy/CRITyFL and result should be binary (may not have a 1-1 with AVAL/AVALC). AVALCATy should look at results on same record and should be mutually exclusive. AVALCATy should have a 1-1 to AVAL/AVALC within a PARAMCD. See PharmaSUG paper from 2015.
14. Using OTHER structure just because. OTHER is not the 'go to'. It is the last resort. If the data will fit in BDS or OCCDS then those structures should be used. Determine what the analysis needs are and don't create a data set just to create it.
15. Maintain traceability. Don't create an intermediate data set and reference it but never actually submit it. Intermediate data sets are great to help gather all information in one place especially when complex computations are involved. If intermediate data set is needed then it needs to be submitted. Need to maintain traceability via metadata and/or data point.
16. Same name, same meaning, same value concept. Variables with same name as SDTM variables must have same meaning and same value. Cannot re-derive.
17. Create a new variable that already is defined in ADaM IG.
18. Need to follow V5 transport rules. Variable name no longer than 8 characters. Value no longer than 200 characters.
19. Not referencing TLF shells when developing ADaM data sets. Need to know what is being analyzed.
20. Not using the ORG coding variables when integrating so that coding from original analysis is kept and the re-coding of terms is stored in the coding variables. Should not re-code