

Exploring Common CDISC ADaM Conformance Findings

Trevor Mankus, Pinnacle 21

ABSTRACT

Having analysis data that is compliant with the CDISC ADaM standard is critically important for the regulatory review process. ADaM data are required to be provided in both FDA and PMDA submissions because the data allows those agencies to better understand the details of the performed analyses and reproduce the results for further validation. Validation of ADaM data is a primary focus for regulatory agencies so they can begin their review of the results.

This presentation will review some of the more commonly occurring validation rules which were found across all our customer data packages which were validated using our automated software and discuss potential reasons for why these rules fired.

INTRODUCTION

Getting started with validation of ADaM data can be as simple as running our automated software or sponsor-specific tools which identify areas of non-compliance. In actuality, the most challenging of assessing ADaM conformance is reviewing the output from these tools and understanding what corrective actions need to be taken for each finding.

Across industry, companies are starting to build tools that enable the tracking of metrics across their studies and compounds. One important metric that should be in the forefront of each company's interest is the frequency of each conformance rule across all validation efforts. Tracking this metric can enable companies to strategically focus their efforts in areas that would benefit from better training material, improvements for existing processes, and even help to identify areas where a standard process could be introduced.

At Pinnacle 21, we have the capability to pool all our customer's validation findings together and identify the most commonly occurring ADaM conformance rules across all their data packages being submitted for validation. This paper documents the most frequently occurring findings, discusses the potential reason for why the issue(s) exists, provides suggestions for corrective action that can be taken to resolve the finding, as well as provides examples of how to document these findings in the Analysis Data Reviewer's Guide.

CONFORMANCE RULE MESSAGE TYPE

The CDISC ADaM Conformance Rules v2.0^[1] package introduces a new concept called "Message Type". Each rule is assigned a message type of either "Error", "Warning", or "Note". The definitions of each message type are:

- Error: Check is a requirement based on the text from the ADaM IG
- Warning: Check meets the following criteria:
 - Concept was/will be clarified in a later version of the ADaM IG
 - May not apply to your study
 - Strong recommendations from CDISC
 - Dictionary variables with expected relationships
- Note: Data quality checks which may require additional investigation to ensure correctness of ADaM data set

Based on these definitions, it is good practice to ensure all rules which are classified as "Error" are resolved as they are based on a clearly documented rule that can be found in the ADaM IG. Rules that

are classified as “Warning” should be carefully reviewed to identify any appropriate action to take, if any. Some rules are marked as “Warning” simply because the text found in the ADaM IG was not clear enough to justify classifying it as an “Error”. Lastly, rules that are classified as “Note” should be reviewed to ensure that the data is correct and that no derivations were performed incorrectly.

Any remaining rules that can be not cleaned up should be described in the ADRG in Section 6.2. Note that this paper contains *examples* of suggested text that can go into this section; the text that you use may differ based on your individual requirements.

CT2002: VARIABLE VALUE NOT FOUND IN EXTENSIBLE CODELIST

This controlled terminology rule was found in ~70% of data packages. In many cases, the rule triggered for the variable RACE in ADSL. In other cases, the rule triggered for the variable DTYPE across all other BASIC DATA STRUCTURE data sets.

CASE 1: ADSL.RACE

The ADaM Implementation Guide mandates that the race of a subject is a required variable in ADSL and that it must be identical to DM.RACE (i.e., no values may be altered). ADaM adheres to a principle of harmonization known as “same name, same meaning, same values”. The SDTM Implementation Guide has the RACE variable subject to controlled terminology found in the *non-extensible* codelist named RACE (C74457).

CL.C74457.RACE	Race (RACE)	text Extensible: No	C74457
	AMERICAN INDIAN OR ALASKA NATIVE		C41259
	ASIAN		C41260
	BLACK OR AFRICAN AMERICAN		C16352
	NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER		C41219
	NOT REPORTED		C43234
	UNKNOWN		C17998
	WHITE		C41261

Display 1. List of Values Found in the RACE (C74457) Codelist

This is where things get interesting. Even though the RACE codelist is classified as non-extensible, the SDTM IG describes the following scenarios:

- If multiple races are collected, then the value of RACE should be “MULTIPLE”, and the additional information will be included in the Supplemental Qualifiers data set.
- If race was collected via an “Other, specify” field then the value of RACE should be “OTHER” and specify the details in SUPPDM.
- If the subject refuses to provide race information, the value of RACE could be “UNKNOWN”.

Pinnacle 21 handles these exceptions by classifying the RACE codelist as extensible in their terminology configuration file^[2] and grades this finding as a Warning as opposed to an Error.

The proposed solution to clean up this validation message is to review the standard values found in the codelist and replace the value in DM.RACE with the most suitable match. For example, if the value for

DM.RACE is “BLACK/AFRICAN AMERICAN” then replace it with “BLACK OR AFRICAN AMERICAN” as per the codelist. Note that this change must be made in the Demographics (DM) SDTM domain since this is where the RACE variable is originally created. If there is no suitable matching value, you should describe this in Section 6.2 of your Analysis Data Reviewer’s Guide (ADRG) with a brief, non-technical explanation of the findings.

Dataset(s)	Diagnostic Message and/or Check ID	Severity	Count/Issue Rate	Explanation
ADSL	CT2002: Variable value not found in extensible codelist	Warning	2	Both subjects selected “Other, specify” on the CRF. Per the SDTM IG, DM.RACE=“OTHER”. Details of race can be found in SUPPDM.

Table 1. Example of Issues Summary Table Found in Section 6.2 of ADRG

In addition to describing this message in the ADRG, you can even consider submitting a request to the CDISC CT team to have the standard codelist amended^[3].

CASE 2: BDS.DTYPE

The DTYPE (Derivation Type) variable is to be used to identify records within a given parameter where the analysis values found in AVAL/AVALC is determined differently than the other analysis values within the parameter. 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. In addition to DTYPE, the metadata for AVAL (or AVALC) will provide further information on the details of any algorithm or statistical method used to derive or impute these values. Per the ADaM IG, the DTYPE variable is subject to controlled terminology found in the extensible codelist named DTYPE (C81224).

When this message is found during data validation, the following actions should be taken:

- Verify the value in your data is the most appropriate based on the CDISC Submission Values found within the DTYPE codelist. For example, if you are imputing a missing value based on the worst observed value carried forward, DTYPE should be populated with “WOCF” (C81199) and not “WORST” or “WORST OBSERVATION CARRIED FORWARD”.
- Check that the value in your data matches the CDISC Submission Value, paying close attention to case and formatting.
- If you’ve reviewed the value in your data and determined it is most appropriate, document this in the Reviewer’s Guide. In addition to describing this message in the ADRG, you can even consider submitting a request to the CDISC CT team to have the standard codelist amended^[3].

Most importantly, try to avoid setting DTYPE equal to a value of “DERIVED”. This does not help describe the method used for populating the analysis value. The presence of a value in DTYPE indicates that AVAL/AVALC was derived; the value found within DTYPE should describe *how* the analysis value was derived.

Dataset(s)	Diagnostic Message and/or Check ID	Severity	Count/Issue Rate	Explanation
ADEFF	CT2002: Variable value not found in extensible codelist	Warning	27	DTYPE set to "SUM" which is not a part of the DTYPE codelist in the 2017-03-31 ADaM terminology package. A value of "SUM" indicates that AVAL/AVALC was derived by adding values from prior observed records. See Section 3.6: Imputation/Derivation Methods for more details.

Table 2. Example of Issues Summary Table Found in Section 6.2 of ADRG

AD0047: REQUIRED VARIABLE IS NOT PRESENT

This rule was found in ~51% of data packages. In many cases, the rule triggered for BDS data sets because there was no record-level treatment variable found. In ADaM IG v1.0 the TRTP variable is required in a BDS data set; however, in ADaM IG v1.1 at least one treatment variable is required in a BDS data set. This requirement is satisfied by any of the subject-level or record-level treatment variables (e.g., TRTxxP or TRTP).

In other cases, our metrics showed that required variables were missing from the Adverse Events Analysis Dataset (ADAE). Throughout the ADaM Structure for Occurrence Data (OCCDS) document, variables are defined with Core=Conditionally Required with the condition being "Required for Adverse Event data". For example, variables such as --DECOD, --BODSYS are required in an ADaM dataset for adverse events but are not required in other non-AE data sets. In addition, the OCCDS document includes Table 3.2.8.1 - Adverse Event Descriptive Variables which defines the AESER (Serious Event) variable as Core=Required.

For this validation message, the most appropriate corrective action to take would be to review the requirements in the applicable standard document and update the ADaM data set so that all required variables are present.

This rule is notorious for producing false-positive messages due to a bug in Pinnacle 21 Community that incorrectly identified OCCDS and ADAM OTHER data sets as BDS data sets. Pinnacle 21 Community v2.2.0 and older versions of Pinnacle 21 Enterprise identified data sets as BDS if any of PARAM, AVAL, ADT, ASTDT, or AENDT variables are present. However, ASTDT and AENDT variables are not good candidates for this task because they can be used in both OCCDS and BDS. This has since been fixed in our methodology for identifying data set structure.

ADaM Structure	Pinnacle 21 Prototype Criteria
ADSL	Data set explicitly named "ADSL"
ADAE	Data set explicitly named "ADAE"
BDS	Any of the following variables present: PARAMCD, PARAM, AVAL, AVALC, CNSR, CNSDTDSC, EVNTDESC
OCCDS	Any of the following variables present: AETERM, AEDECOD, MHTERM, MHDECOD, CETERM, CEDECOD, DSTERM, DVTERM, HOTERM, CMTRT, EXTRT, EXTRT, SUTRT, PRTRT
ADAM OTHER	Data set with name beginning in "AD" that is not classified elsewhere

Table 3. Criteria Used by P21 to Determine ADaM Structure

AD0018: VARIABLE LABEL MISMATCH BETWEEN DATASET AND ADAM STANDARD

This rule was found in ~50% of data packages. Section 3.1.6 (Additional Information about Section 3) of ADaM IG v1.1 states the following:

- Variable labels specified in the ADaM and ADaM IG are required with two exceptions:
 - Descriptive text is allowed at the end of labels of variables whose names contain indexes “y” or “zz”; and
 - 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.

For this validation message, the most appropriate corrective action to take would be to review the variable label found in your data against the variable label specified in the ADaM or ADaM IG document. Special care should be taken to ensure casing and formatting is followed so that the values are an exact match. If the errant label is for a variable which meets one of the exceptions described above, then you should review the value of the label up to the point where producer-defined text can be introduced to ensure it matches the standard label. For example, ANL01FL can have a label of “Analysis Flag 01 – Thromb Event” because the producer-defined text appears *after* the standard label text (e.g., Analysis Flag 01).

In some cases, this rule may issue a false-positive in Pinnacle 21 where descriptive text is added at the end of labels of variables whose names contain indexes of “y” or “zz”. This is something we have identified and are working on fixing in a future release of the validator. In such instances, you should describe your findings in the ADRG.

Dataset(s)	Diagnostic Message and/or Check ID	Severity	Count/Issue Rate	Explanation
ADAE	AD0018: Variable label mismatch between dataset and ADaM standard	Error	1	Variable label matches the ADaM standard up until producer-defined text was added, which is allowed per ADaM IG v1.1 - Section 3.1.6. ANL01FL (Analysis Flag 01 – Thromb Event)

Table 4. Example of Issues Summary Table Found in Section 6.2 of ADRG

AD0124: INCONSISTENT VALUE FOR PARCAT WITHIN A UNIQUE PARAMCD

This rule was found in ~24% of data packages. Of these, most of the issues were found in Laboratory Data or Pharmacokinetic Parameter ADaM data. Many producers of ADaM struggle with this rule because the relationship between ADaM variables PARCATy and PARAM is different than that of SDTM variables –CAT and –TEST which is what they may be more familiar with.

In SDTM, the –CAT variable is used to sub-divide or qualify –TEST values further. For example, you may have:

LBTESTCD	LBTEST	LBCAT
GLUC	Glucose	URINALYSIS
		CHEMISTRY

Table 5. Sample SDTM.LB Domain Data

The example above is valid in the SDTM.LB domain; however, it does not translate well into ADaM category concepts. In ADaM, the PARCATy variable is intended to group PARAM values into categories.

It is important to remember that PARAM to PARCATy must have a many-to-one mapping; any given PARAM may be associated with at most one level of PARCATy.

❌ Incorrect usage of PARAM:PARCAT1		✅ Correct usage of PARAM:PARCAT1	
PARAMCD	PARCAT1	PARAMCD	PARCAT1
GLUC	Urinalysis	UGLUC	URINALYSIS
	Chemistry	GLUC	CHEMISTRY

Table 6. Examples Showing Incorrect and Correct Usage of PARAM:PARCATy

In the sample data above, the producer had to modify the value in PARAMCD to make GLUC unique for Urinalysis and Chemistry. In this instance, they opted to amend a “U” prefix to the existing value of PARAMCD when LBCAT=URINALYSIS. Note that the decision of how to do this is left up to the producer of ADaM and is not managed by CDISC.

CONCLUSION

The examples described throughout this paper highlight the importance of having the ability to report metrics on ADaM conformance across your organization. Understanding commonly occurring areas of non-compliance with the standard can help your organization better develop training materials. In addition, it provides insight into what type of in-house tools can be developed to enforce adherence to the standards especially in areas where the non-conformance can be easily avoided. It is important to note, however, that assessing conformance with the CDISC ADaM standard is more than just passing validation checks. Part of the ADaM standard is governed by principles and defined by concepts which are not machine-testable using automated checks. Data sets must be reviewed by knowledgeable and capable individuals who have a firm understanding of the ADaM standard and core concepts.

Documenting the reason for any remaining conformance findings in the ADRG is just as important as reviewing your data. The ADRG is one of the most helpful and important documents for a regulatory reviewer when they receive the submission package. Be sure you provide the reviewer with comments which are helpful and provide insight into why any remaining issues are present. Doing so can help ensure a timely review and show that you have thoroughly reviewed your data against the ADaM standard.

REFERENCES

- [1] CDISC ADaM Conformance Subteam. 2019. “CDISC ADaM Conformance Rules v2.0.” Published February 20, 2019. <https://www.cdisc.org/standards/foundational/adam>
- [2] Pinnacle 21. 2019. “CDISC Terminology for Validation.” Accessed January 23, 2019. <https://www.pinnacle21.com/downloads/cdisc-terminology>
- [3] National Cancer Institute. 2017. “CDISC Terminology.” Accessed January 23, 2019. <https://www.cancer.gov/research/resources/terminology/cdisc>

CONTACT INFORMATION

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

Trevor Mankus
 Pinnacle 21
tmankus@pinnacle21.com
www.pinnacle21.com