

Conformance, Compliance, and Validation: An ADaM Team Lead's Perspective

John Troxell, Accenture, Berwyn, PA

ABSTRACT

Conformance, compliance, and validation checks have become a hot topic. In 2008 as I became ADaM Team Lead, I observed that SDTM computer validation checks had not been developed by CDISC. I believe that CDISC alone should define what constitutes CDISC compliance and, specifically, the algorithms for testing those aspects of compliance that are checkable by computer. Compliance rules, in my view, are naturally part of the standard. The ADaM Team published the first version of the ADaM validation checks in 2010, based on the ADaM Implementation Guide published in 2009. Updated versions have been released to reflect subsequent ADaM publications. However, all is not yet perfect. This paper provides a suggested map of the space comprising various aspects of regulatory agency conformance, CDISC compliance, and computer validation checks. To facilitate discussion, the paper also attempts to define vocabulary to refer to various components of the space. The paper looks at the chain of things that have to go right with validation checks, starting from the standards themselves, through rule definition and implementation. Finally, the paper discusses recent efforts toward improving the accuracy of computer validation checks.

INTRODUCTION

An ideal software tool that examines a set of analysis datasets for compliance to the CDISC ADaM standard would identify real problems, and not produce erroneous and misleading error messages. Validation software is a bit like a teacher. If the error messages are valid, they help educate users about some things that they might be doing wrong. If the error messages are erroneous however, then not only is time wasted, but the wrong actions might be taken in response, and the wrong lessons might be learned about the CDISC standard.

The goal of this paper is to lay a foundation for understanding, clear communication, and improvement efforts. It is important to acknowledge that everyone involved, including but not limited to CDISC volunteers, regulatory agencies, and software developers, has the same goals in mind: correct and instructive error messages, and compliant clinical data interchanges.

While some of the ideas and terms in this paper are my own, many were originated by others. In some cases, I am just the messenger.

ORIENTATION: A MAP AND A VOCABULARY

While there is growing awareness that there are issues with computer checks of adherence to ADaM and other CDISC standards, communication about conformance, compliance, and validation may be difficult because these words are somewhat open to interpretation. The words carry baggage and are commonly used in different ways by different people.

A necessary starting point in identifying root causes and improving the situation is a better vocabulary, so that in discussions about the issues, the speaker and the listener are on the same wavelength.

Even before better vocabulary is defined, however, we need to identify what are the things for which we need the clear vocabulary. On the next page, I present an entirely unofficial, suggested map of the space under discussion. While I think the map is not a bad starting point, my purpose in presenting it is to stimulate further progress on space definition and vocabulary development.

MAPPING THE SPACE

For discussion, I offer this completely unofficial map of the space about which we need clearer vocabulary.

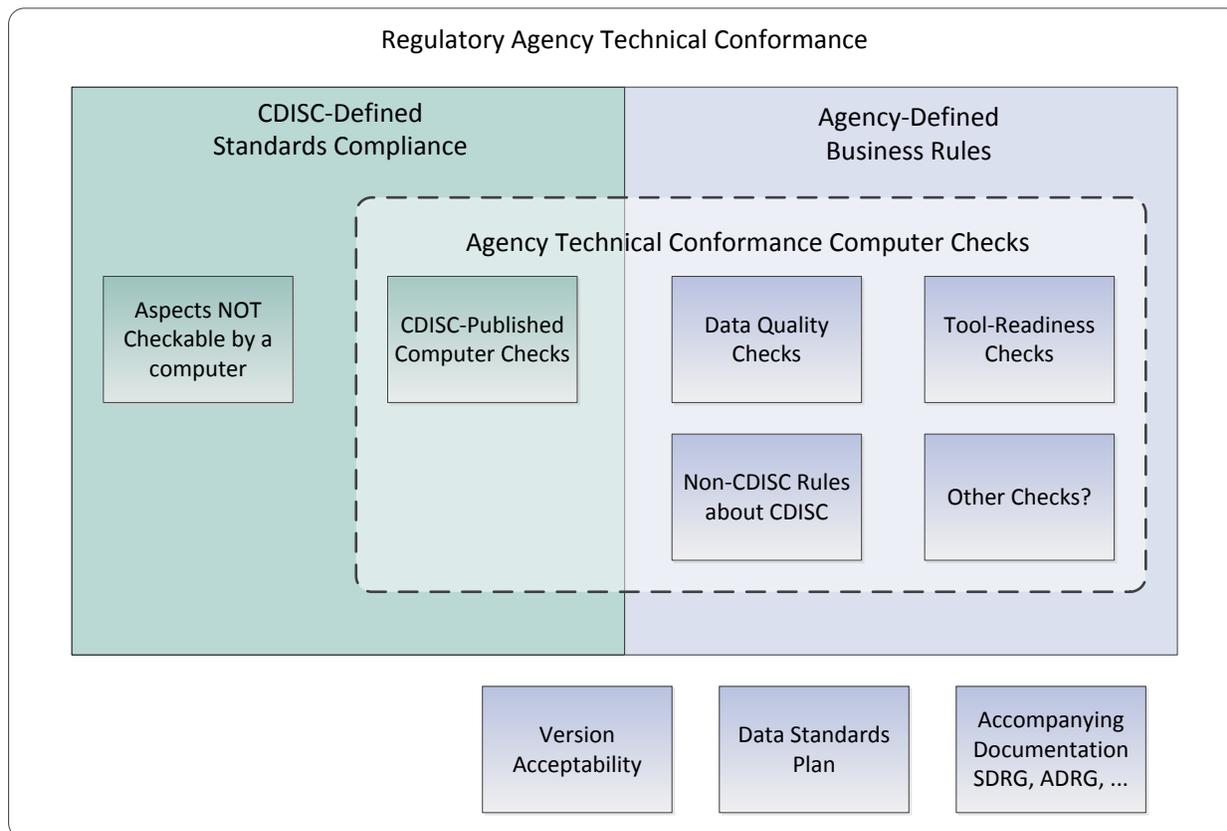


Figure 1. Unofficial Map of the Technical Conformance, CDISC Compliance, and Computer Validation Space

Some of the important components of this space are:

- **Regulatory Agency Technical Conformance.** Defined in an agency's Technical Conformance Guide. Encompasses all items in the map.
- **CDISC-Defined Standards Compliance.** Compliance with CDISC standards as defined by CDISC in the published CDISC standards documents, whether or not explicitly listed in separate sets of rules. Some aspects of CDISC compliance are currently checkable by computer. Other aspects of compliance are NOT currently checkable by computer, and may never be.
- **Aspects NOT checkable by a computer.** Those aspects of CDISC standards compliance as defined by CDISC that are not currently checkable by computer.
- **CDISC-Published Computer Checks.** Currently, CDISC has published rules for computer checks for ADaM compliance. Publication of checks for other standards may follow.
- **Agency Technical Conformance Computer Checks.** Checks run by a regulatory agency, including CDISC-Published Computer Checks, and Agency-Defined Business Rules.
- **Agency-Defined Business Rules.** Computer checks of regulatory agency technical conformance defined by the agency. Does not include CDISC-published computer checks.
- **Data Quality Checks.** Checks having to do with the quality of collected data. Not about CDISC standards.
- **Tool-Readiness Checks.** Checks having to do with whether the data support review tools.
- **Non-CDISC Rules about CDISC.** Checks about compliance to CDISC standards invented by others than CDISC and that are used by the agency. May or may not be consistent with the CDISC standards.

VOCABULARY

It is unlikely that we will ever have simple one-word descriptions of each of the components of this space. I would recommend that at least we all attempt to use more adjectives to try to clarify what we are talking about. A few more words used to describe something is a small price to pay for successful communication in improvement efforts.

I have used certain vocabulary in the map, but the matter of vocabulary is not yet settled.

For example, “conformance” is used by some to refer to both agency technical conformance, and adherence to CDISC standards. In attempt to distinguish between agency technical conformance and one of its components, adherence to the CDISC standards, I have used the word “compliance” to refer to adherence to the standard. However, it has been noted that the word “compliance” may bear a heavy connotation in other contexts at a regulatory agency. Should the word “adherence” be used?

I have chosen to resist as far as possible using the word “validation,” as I find it misleading in the context of CDISC compliance. For after all, even if computer checks return no error messages, it does not mean that the data are CDISC-compliant. That is because, as we have already seen, there are aspects of standards compliance that are not currently checkable by computer. Therefore, absence of error messages does not mean that the data are “valid”; they cannot be “validated” by computer.

Computer checks of CDISC standards compliance, in this sense, are “invalidation checks”, since they may demonstrate that the data do not adhere to the standard; but they cannot prove that the data do adhere to the standard (at least for ADaM and SDTM data).

ROLES

In what follows, I refer to certain actors as follows.

CHECK SPECIFIER

The person or entity who defines the computer check. For the published CDISC ADaM-specified computer checks, the check specifier is the CDISC ADaM Team. For the currently-implemented SDTM computer checks, the check specifier was a software vendor, or an agency, not CDISC.

PROGRAMMER; SOFTWARE DEVELOPER

The person or entity who develops a software tool that implements the checks specified by the check specifier.

THE CHAIN OF THINGS THAT HAVE TO GO RIGHT

There is a chain of things that have to go right in order for error messages about compliance to a CDISC standard to be correct. The chain is:

1. The CDISC standard must be clear and internally consistent;
2. The check specification must not conflict with the standard;
3. The check specification must be clear and complete enough for a pure programmer;
4. The check specification must be understood by the programmer;
5. The check must be implemented correctly by the programmer;
6. The content of the error message must be correct.

Failure at any point in the chain can result in invalid error messages. Currently, I think that failures are occurring at each of these points.

CAUSES OF ERROR MESSAGES THAT CONFLICT WITH THE ADAM STANDARD

This section discusses some of the causes of breakdowns at each of the steps in the chain.

ADaM may be more challenging from a computer checks perspective than SDTM. ADaM is mainly about types of generic, extensible containers for analysis-dependent and analysis-ready observed and derived data and metadata. In contrast, SDTM addresses what particular medical domain data to put into which corresponding medical-domain-labeled SDTM container.

Nevertheless, the causes of incorrect error messages discussed here apply in principle to computer checks for any CDISC data standard. There are many possible underlying causes of an incorrect error message. Among them are the following.

1. THE CDISC STANDARD DOCUMENT MAY NOT BE CLEAR ENOUGH

Sometimes, try as we may, a CDISC standard document may not be completely clear, especially to every possible audience. One of the primary reasons that ADaM Implementation Guide (ADaMIG) v1.1 was developed was to clarify portions of ADaMIG1.0 and improve internal consistency.

Clarification can involve more than just expressing something clearly; sometimes it also involves debate and resolution within the ADaM Team about the concept being clarified. That is because a discussion about how to clarify the text may reveal that different people understand the concept differently. In such a case, the concept itself must be clarified before the team can write a clearer description of it.

2. THE COMPUTER CHECK SPECIFICATION ITSELF CONFLICTS WITH THE CDISC STANDARD DOCUMENT

When specifying computer checks, it can be tempting to be aggressive in interpreting statements in the source document. Examples in a CDISC standard document that may have been intended to illustrate a possible solution to a given scenario might be interpreted by the check specifier as the mandatory, universal way to do something. Rules may be stated about a variable name that apply in the context of the concept for which the variable was invented, but are not universally applicable to all variables whose names share similar name fragments. Overly aggressive hunts for computable rules may result in specification of a check that is not always valid.

CDISC standards documents are written for many audiences, and the computer checks specifier or programmer was either never envisioned by the authors as a member of the audience, or has not been high on the list, at least so far. Therefore, there are statements made in the documents that were implicitly written for a certain audience, and that might at first glance appear to be computable, but upon further reflection, they might be limited only to certain situations, or were intended more as advice or recommended practice than prescription.

This cause is more prevalent with checks about CDISC standards that are specified by others than the CDISC Team who wrote the document. But it can apply even when the checks are specified by the CDISC Team itself. Those who specify CDISC compliance checks bear great responsibility. Checks should not be specified by those who are not true experts in the standard.

In particular, when a party other than CDISC wishes to specify a check about compliance to a CDISC standard, the best approach is to bring the idea to the relevant CDISC team, rather than to promote the check unilaterally. At the very least, the impression should not be given that the check was specified by CDISC, or that others than CDISC are authorities on CDISC compliance.

3. THE COMPUTER CHECK SPECIFICATION IS NOT CLEAR AND COMPLETE ENOUGH FOR A PURE PROGRAMMER

Computer check software developers were not the main audience implicitly assumed by authors of the CDISC documents. It can be that a computer check software developer has limited to no personal experience of the CDISC standard in question, or might not have a particular type of clinical trials expertise assumed by authors of the source document or specifiers of the check.

Statements in the ADaMIG or other CDISC document are often made in the context of a background that has been explained elsewhere in the document. For example, in ADaM, even for required variables, nulls are allowed, unless otherwise specified. Statements about allowable missing values might not be specified in a check specification, but are part of the context in which the check was written by the specifier. That knowledge may have been implicitly assumed by the check specifier, rather than explicitly stated in the specification of a check to which it applies.

A computer check specification may not include information, for example, about what to do in certain data scenarios, such as missing values. Practical specifics may have been glossed over, intentionally or unintentionally, by the specifiers of the check. However, a programmer needs to know. The result can be that the software developer has to guess what to do.

4. THE COMPUTER CHECK SPECIFICATION MAY BE MISINTERPRETED BY THE PROGRAMMER

This cause is far more likely when the check is specified incompletely and when the programmer does not have an assumed background. But also, there is, at least currently, a responsibility on the part of the software developer, to read the whole underlying source document, and possibly its associated documents, not just the rules specifications. For example, information about permissibility of missing values might be found somewhere else in the documents than in a compliance check specification, or in the section of the document to which the check specification points as its source.

5. THE CHECK IS NOT IMPLEMENTED CORRECTLY BY THE PROGRAMMER

Even assuming all the prior steps in the chain have been accomplished flawlessly, there is the possibility that there is a bug in the software.

In addition, the programmer may find seeming commonalities among the check specifications, and attempt to generalize, in order to develop subroutines and functions that exploit these seeming commonalities for reasons of programming efficiency. It is possible that this generalization into algorithms useful across many checks might not always be valid, particularly when other steps in the chain are not flawlessly executed.

6. THE CONTENT OF THE ERROR MESSAGE IS NOT CORRECT

Even assuming that all the prior steps in the chain have been accomplished flawlessly, and there is indeed an error, there is the possibility that the text or content of the error message is not correct.

In my opinion, the content of the message should be specified by the check specifier. The error message should identify the check specifier, and that standard message content should be implemented by the by the software developer. I think there is some room for value-added additional explanation on the part of the software developer, but only if clearly identified as not originating from the specifier of the check.

IMPROVING THE CURRENT SITUATION

This section presents some ideas about how to improve the situation that have been contributed by many sources. Besides myself, these sources include other members of the ADaM Leadership Team (ALT), and other parties, both within and outside of CDISC, both before and during the discussions of this topic at the March 2016 CDISC Intrachange. Not all of them are agreed upon, but some are currently underway. I have tried to organize them by responsible entity, but some of them may span entities. I have not included all ideas that have been contributed.

IMPROVEMENTS: EVERYONE

- Use better vocabulary about the conformance, compliance and validation space. The unofficial, suggested map of the space presented here, based partly on vocabulary put forward by an agency, was discussed during a session about validation at Intrachange. It seemed to strike a chord among the attendees. Not surprisingly, as the discussion has just gotten started, no agreement has yet been reached on vocabulary.
- In the meantime, understand the space, and use clear descriptions and adjectives to define what we are talking about. Don't use vague words without clarification, like "validation". We will all benefit if we all make an effort to be clear just exactly what it is that we are talking about.

IMPROVEMENTS: CDISC TEAMS

- Clarify the CDISC Standards as much as possible. Ongoing.
- CDISC Teams should define the CDISC compliance computer checks. ADaM Team has done this. SDS Team has released a draft for public comment for SDTM. CDISC did agree to this at a prior Intrachange.
- It may be helpful if the format for the specification of CDISC compliance computer checks is the same across CDISC Teams. This cross-team discussion has not yet taken place.
- Creation of test data for computer checks (both pass and fail examples) would help software developers.
- Those who write computer check specifications should not be overly aggressive in extracting rules from the standards, in order not to specify checks that are not universally appropriate. For instance, examples in the documents should not be over-interpreted, since not all examples illustrate universal rules.
- Senior members of the ADaM team, including ALT, must participate in decisions about ADaM compliance check specifications. ALT has started reviewing the current ADaM computer checks.
- Compliance check specifications must be clear and complete enough for a pure, CDISC-naïve programmer. This is just my idea; no discussion has occurred.
- It may be worth revisiting the <error / no error> binary nature of the current ADaM compliance check specifications, and move towards a continuum of note, warning and error. There could be considerable value in a more graduated approach. There is no consensus on this yet.
- CDISC could consider adding a Define-XML dataset metadata subclass attribute. A subclass attribute would permit tools to know more about a dataset. For example, if the class of a dataset is BDS, and the subclass were

TTE, then a tool could know that it is appropriate to fire checks having to do with the time-to-event variables; and similarly for the class OCCDS and subclasses ADAE and ADCM. This idea is under discussion; the examples and values here are purely for illustration of the concept and are hypothetical.

- The ADaM Team's process for tracking and managing issues and suggestions regarding compliance checks should be formalized and strengthened, including making use of the tools available from CDISC for issue tracking and management.
- CDISC should respond expeditiously and reliably to the issues and suggestions about CDISC compliance checks provided by software developers. This is agreed.

IMPROVEMENTS: COMPUTER CHECK SOFTWARE DEVELOPERS

- Programmers should try hard to understand the standard, not just read the check specifications. Read the whole standard. Don't assume everything you need to know is on the same page. Ask the authoring CDISC team when in doubt. It may be a long time before the computer check specifications assume zero knowledge about the standards.
- Software developers should submit ideas about computer checks of CDISC compliance to CDISC, collaborate with CDISC, and not release their own checks about CDISC compliance unilaterally.
- Software developers should not state or give the impression that they are experts in the standards, and specifically state that any advice they give about CDISC standards compliance is not authoritative.
- Error messages should identify the specifier of the computer check that resulted in the error message. For example, CDISC (which team), Agency (which one), software developer (name of company). This wish for clarification on the source of error messages has widespread support.
- Error messages produced by software implementing computer checks of compliance to a CDISC standard should include a standard error message defined by the check specifier. Additional non-standard text added by a tool developer should be clearly delineated as originating from the developer, not CDISC. This is just my idea so far; no discussion has occurred.

IMPROVEMENTS: REGULATORY AGENCIES

- Agencies should submit ideas for computer checks of CDISC compliance to CDISC, collaborate with CDISC, and not release their own checks about CDISC compliance unilaterally.
- When requesting that software developers implement checks, involve the relevant CDISC team. This may prevent agency-originated checks that conflict with the CDISC standards.
- Agencies should strive to clarify which of their requirements are about CDISC compliance, and which are not.

IMPROVEMENTS: COMPUTER CHECK SOFTWARE USERS

- Issue reports by users and discussions about computer check software should refer to a specific software and software version.
- All implementers and consumers of the CDISC standards have an obligation to understand the standards and not rely exclusively on software tools for validation. As has been discussed, passing the computer checks does not prove compliance. Manual review is essential. If in any doubt, expert advice should be sought.

CONCLUSION

Currently, there are issues with computer checks of adherence to ADaM and other CDISC standards that result in invalid and misleading error messages. The goal of this paper was to lay a foundation for understanding, clear communication, and improvement efforts. It is important to acknowledge that everyone involved, including but not limited to CDISC volunteers, regulatory agencies, and software developers, has the same goals in mind: correct and instructive error messages, and standards-compliant clinical data interchanges.

ACKNOWLEDGMENTS

The author wishes to thank the ADaM team and others in CDISC, regulatory agencies, and software developers for their ideas and efforts toward achieving the common goal of accuracy and clarity in error messages about compliance with CDISC standards. In addition, the author wishes to thank his colleagues Sandra Minjoe and Fred Wood for their helpful review comments.

CONTACT INFORMATION

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

John Troxell
The Data Standards Consulting Group
Accenture
Berwyn, PA
john.troxell@accenture.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.