

Abstract

Writing ADaM programming specification has become a perfunctory task in regulatory submissions. The programming specification documentation serves as the primary source for ADaM programming. Define.xml, and Study Data Reviewer's Guide. ADaM programming for FDA submission should meet submission requirements and follow CDISC ADaM guidelines. The OpenCDISC Validator is a very useful tool to check data compliance with CDISC models such as SDTM, ADaM and Define.xml. However the tool requires datasets as one of its inputs, which are normally available at later stage of programming cycle. Sometimes it is too late and/or costly to fix the errors identified by the tool at the later stage of programming cycle. This paper introduces a preventive approach to check ADaM metadata compliance with CDISC ADaM guidelines at an earlier stage even before actual ADaM dataset programming thereby avoiding the waste of time and resources for correction of ADaM metadata at a later stage. It also automatically ensures the consistency of variable attributes between ADaM datasets and the define files, which guarantees technical accuracy and operational efficiency.

ADaM Programming Specification Template

1.1.1 ADSL: Subject Level Analysis Dataset

Variable	Variable Label	Type	Length	Controlled Terms or Formats	Origin	Role	Comments	Cat
STUDYID	Study Identifier	Char	20	Controlled	Supplied	Identifying	Constant Value "ABC1234567890"	Req
SITEID	Study Site Identifier	Char	20	Controlled	Supplied	Identifying	Constant Value "CD1234567890"	Req
SEQ	Sequence Number	Num	8	Controlled	Supplied	Identifying	Constant Value "12345678"	Req
AGE	Age	Num	8	Controlled	Derived	Analysis	Calculated	Req
WEIGHT	Weight	Num	10	Controlled	Derived	Analysis	Calculated	Req
DOB	Date of Birth	Date	8	Controlled	Derived	Analysis	Calculated	Req
SEX	Sex	Char	1	Controlled	Derived	Analysis	Calculated	Req
DOSE	Dose	Num	10	Controlled	Derived	Analysis	Calculated	Req
DOSEID	Dose Identifier	Char	4	Controlled	Derived	Analysis	Calculated	Req
DOSEFLG	Dose Flag	Char	1	Controlled	Derived	Analysis	Calculated	Req
DOSEIDR1	Dose Identifier	Char	4	Controlled	Derived	Analysis	Calculated	Req
DOSEFLGR1	Dose Flag	Char	1	Controlled	Derived	Analysis	Calculated	Req

Figure 1. An example of Individual ADaM Programming Specification in Word Format – ADSL.DOC

How to Write Domain Information Table in ADaM Programming Specifications

Checking Number	Checking	Requirement	Key Words
1	Description	1. Non-Missing 2. length <= 40	
2	Unique Identifier Variables	Non-Missing	
3	Structure	Non-Missing	
4	General Class	Non-Missing	Special Purpose Interventions Events Findings

How to Write Variable Information Table in ADaM Programming Specifications

Checking Number	Checking	Requirement	Key Words	Source	Corresponding Rule ID in OpenCDISC	Checking Number in ADaM Validation Checks
5	Variable Name	1. Length <= 8; 2. Start with a letter, comprised of letters (A-Z), underscore (_), and numerals (0-9).		CDISC Requirement	AD1006	13 14 15
6	Variable Label	Length <= 40		CDISC Requirement	AD0016	16
7	Type	Non-Missing	Char, Num	CDISC Requirement		
8	Length	Non-Missing for character variables Length <= 200 for character variables		CDISC Requirement		17
9	Controlled Terms or Formats	1. If the Controlled Terms are given: YES:NO; C1Y; C2N; Or AVISITN (AVISIT) (1) 950 = Baseline (2) 1001 = Day 1 (3) 1029 = Week 4 for a pair of code-decode variables. 2. If no Controlled Term Name is provided, then assume it the same as the Variable Name. 3. If the Formats are given: a. if the Variable Name ends with DT, then Variable Label must contain 'Date'. Type = 'Num', Role = 'Timing', and format = 'YYMMDD10'. b. if the Variable Name ends with TM, then Variable Label must contain 'Time'. Type = 'Num', Role = 'Timing', and format = 'TIME5'. c. if the Variable Name ends with DHM, then Variable Label must contain 'Date/Time' or 'DateTime'. Type = 'Num', Role = 'Timing', and format = 'DATE TIME20'. 4. If the Variable Name ends with DM, then Variable Label must contain 'Date/Time' or 'DateTime'. Type = 'Num', Role = 'Timing', and format = 'DATE TIME20'.	1. Vertex Requirement 2. Vertex Requirement 3. CDISC Requirement	For Datetime Variables: AD0041 AD0042 AD0043 AD0058 AD0059 AD0060	For Datetime Variables: 41 42 43 58 59 60	
10	Origin	Non-Missing		CDISC Requirement		
11	Role	Non-Missing	Identifier Topic Timing Grouping Qualifier Result Qualifier Synonym Qualifier Reserved Qualifier Variable Qualifier Selection Analysis	Vertex Requirement		
12	Comments	Non-Missing for Origin Derived (at the FINAL run)		Vertex Requirement		
13	Care	Non-Missing	Req Cond Perm	CDISC Requirement		

General Consistency Checking Rules vs. Metadata Related Rules

- Functional groups defined by CDISC ADaM validation checks
 - Value Consistency checking
 - Metadata checking
 - Present/Populated checking
 - Controlled Terminology checking
 - Valid Value checking

Comparison with OPENC DISC Validation Rules

OPENC DISC Validation	Preventive Approach for ADaM Metadata Validation
<ul style="list-style-type: none"> Check at the very end of ADaM programming activities Most of CDISC ADaM Validation Checks: Value Consistency + Metadata + Present/Populated + Controlled Terms + Valid Value Source: ADaM dataset + define.xml 	<ul style="list-style-type: none"> Check at any stage of ADaM programming activities Focus on Metadata Related Rules: ADaM Metadata + Variable Presence + Part of Controlled Terms Source: ADaM specs ONLY Unique Checking: <ul style="list-style-type: none"> Compliance of Domain Information Compliance between Domain Information and Variable Information Key Words Checking for Specs Existence Checking of Decoded Variables

Programming Flow Chart

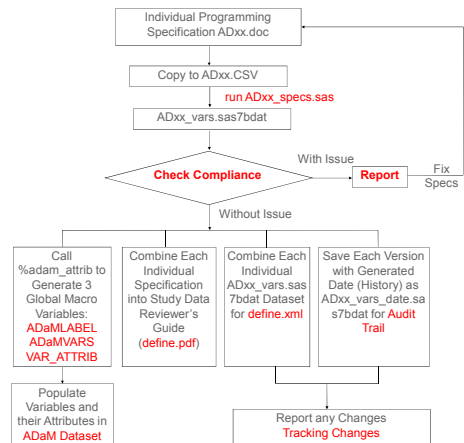


Figure 2. Flow Chart for Automatic Compliance Checking of CDISC ADaM Metadata

A SAS Macro Automates Compliance Checking of Metadata Against CDISC Standards

A SAS Macro %read_specs

- Retrieves domain information and variable information from ADaM programming specifications.
- Performs ADaM compliance checking with CDISC requirements and FDA submission requirements
- Outputs non-compliance reports if any

A Preventive Approach for Automatic Checking of CDISC ADaM Metadata to Detect Noncompliance

Xiangchen (Bob) Cui¹, Min Chen¹, Tathabbai Pakalapati²,

¹Statistical Programming, Vertex Pharmaceuticals, Cambridge, MA 02139

²Clinical Research Services, Cytel Inc., Cambridge, MA 02139

Examples of Non-Compliance Reports

1. Non-Compliance ADSL Domain

Domain	Error Type
ADSL	Missing Domain General Class
General Class key word: Special Purpose/Interventions/Events/Findings	

2. Non-Compliance ADSL Variables

Variable	Error Type
HCVOTYPE	VAR Label > 40 Chars
O_TMDIAG	VAR Name > 8 Chars
TRTSDT	Missing Type
	TYPE key word: Char Num
	All *DT variables must be numeric
CMPEATFL	Missing Core
	CORE key word: Eqg Cond Perm
DISCFN	DISCFN variable is present but DISCFL variable is not present
STUDVID	STUDVID variable is not in the dataset ADSL
AGEU	AGEU variable is not in the dataset ADSL
FASFN	FASFN variable is present but FASFL variable is not present in the dataset ADSL
*FL	No variable that ends in FL is in the dataset ADSL

3. Non-Compliance ADSL Key Variables

Variable	Error Type
TRTCD	Unique Identified Variable TRTCD is Defined in Domain Information Table, but Not Defined in Variable Information Table

4. Duplicate Variables defined in ADSL

Variable	Variable Number
SEX	9
	92

5. Non-Compliance Report for Decoded

Variable	Error Type
PHASEFN	Variable PHASEFN does not have a paired variable PHASEFL in the specification ADLB.DOC

Output Two Datasets After All Compliance Checking Are Passed

1. Individual ADaM Data Domain Information – ADSL_DOMAIN

RUNORDER	DOMAIN	DESCRIPTION	STRUCTURE	KEYS	PURPOSE	CLASS	PATH	RELPATH	CLASSRO	REPEATING	ISREFERENC
1	ADSL	Subject-Level Analysis Data	One recor	USUBID	Analysis	Special				1 No	No

2. Individual ADaM Data Variable Information – ADSL_VARS

DOMAIN	VARNUM	VARIABLE	LABEL	TYPE	LEN	ORIGIN	TERM	CODELIST	ROLE	COMMENT	CORE	MANDAT	paired
ADSL	5	AGE	Age	Num	8	DM	age		Record Qualifier	Equals to DM age	Req	Yes	
ADSL	6	AGEGR1	Pooled Age Group 1	Char	20	Derived			Record Qualifier	<=45, if age <= 45; >45 and <=65, if 45 < age <= 65; >65, if age > 65; Note: Decode variable for AGEGRM.	Perm	No	
ADSL	7	AGEGRIN	Pooled Age Group 1 (N)	Num	8	Derived	AGEGRIN	(1)1 = <=45 (2)2 = >45 and <=65 (3)3 = >65	Synonym Qualifier	Category derived if age non-missing; Equals: 1, if age <=45; 2, if 45 < age <= 65; 3, if age > 65	Perm	No	AGEGR1
ADSL	8	AGEU	Age Units	Char	8	DM	ageu	YEAR	Variable Qualifier	AGEU = DM.AGEU	Req	Yes	

Conclusions

This paper introduces a macro-based comprehensive approach for automatic compliance checking of ADaM programming specification with CDISC ADaM Standards. It can detect any non-compliance between ADaM metadata and CDISC ADaM Guidelines prior to the generation of actual ADaM datasets, and therefore ensure the proper ADaM dataset structure earlier in the programming cycle. The guidelines for writing modularized ADaM programming specification are also provided, as well as compliance checking rules.

Compared with OpenCDISC, our macro-based approach is powerful for checking the compliance of ADaM metadata. As a preventive approach, our tool makes it possible for ADaM programming specifications to be finalized at very early stage of the programming cycle, and thus minimizes the chance to errors and avoids the unnecessary efforts for verification of the compliance at the later stage. The high quality of the submissions can be achieved in a cost-effective and efficient way.

References

- “CDISC SDTM/ADaM Pilot Project, Project Report”- <http://www.cdisc.org/stuff/contentmgr/files/0/df91a087c6df43275288267c9fe92180/misc/sdtdamadpilotprojectreport.pdf>
- Xiangchen (Bob) Cui, Min Chen. “An Innovative ADaM Programming for FDA Submission: Streamline the Programming Specification Creation, Compliance Checking, ADaM Derivation, Version Control, and define.xml Generation”, PharmSUG, May 2012.
- CDISC Analysis Data Model Team. “Analysis Data Model (ADaM) Implementation Guide”. December 2009. <http://www.cdisc.org/adam>
- CDISC Analysis Data Model (ADaM) Team. “CDISC ADaM Validation Checks”, July 2012. <http://www.cdisc.org/adam>
- Xiangchen (BoB) Cui, Min Chen. “Automatic Consistency Checking of Controlled Terminology and Value Level Metadata between ADaM Datasets and Define.xml”, SAS Global, April 2012.

Acknowledgments

Appreciation goes to Cindy Song for her review and comments

Contact Information

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

Name: Xiangchen Cui, Ph.D.
Company: Vertex Pharmaceuticals, Inc.
E-mail: xiangchen_cui@vrtx.com

Name: Min Chen, Ph.D.
Company: Vertex Pharmaceuticals, Inc.
E-mail: min_chen@vrtx.com

Name: Tathabbai Pakalapati
Company: Cytel Inc.,
E-mail: tathabbai_pakalapati@vrtx.com