Paper CD07

“Compliance” for Analysis Data

Chris Decker, d-Wise Technologies, Raleigh, NC, USA
Randall Austin, GlaxoSmithKline, Research Triangle Park, NC, USA

ABSTRACT

Dictionary.com defines compliance as “the degree of constancy and accuracy with which a person follows a prescribed procedure”. Over the years the clinical research community and other stakeholders have developed computer-testable validation criteria to ensure that “CDISC data” meet the criteria outlined in the CDISC SDTM model. These SDTM validation criteria weren’t developed and maintained by the same team which developed the model, so the interpretation of requirements, as expressed in published validation checks, sometimes differs depending on who developed the checks and for what purpose they were developed. These differences introduce confusion and challenges as users attempt to validate their data.

With the release of the ADaM Implementation Guide, the CDISC ADaM team chose to take a different approach. The team decided to define and publish its own set of machine-readable standard checks for the ADaM model, thus creating one authoritative source for ADaM validation criteria. This paper discusses the significance of compliance within clinical research, what ADaM validation implies and the ADaM Team’s process to define compliance checks for analysis data.

INTRODUCTION

The CDISC Analysis Data Model (ADaM) Team published the ADaM Implementation Guide (IG) 1.0 in December, 2009. The team recognized that if the data model is to be accepted as a standard within the pharmaceutical industry, it must be adhered to in a consistent manner from one company to the next. This is especially true since transparency, supported by predictable data representation, forms the core of the ADaM model.

Therefore a business need exists to clearly define expectations for what “an ADaM dataset” and “an ADaM regulatory submission” mean. Subjective criteria must be separated from objective criteria. What is minimally-acceptable versus “nice to have” must be clarified.

The ADaM 2.1 and ADaM IG 1.0 go a long way towards explaining these expectations but they do so in textbook fashion. The experience of SDTM adoption shows that clinical research organizations, as well as software and technology vendors, crave explicit implementation rules written in the form of straightforward, software-codeable, business requirements. If such rules are not available, users will take the initiative to develop their own rules. It is no surprise that the result may be rules that are somewhat fit for the purposes of the developer rather than clearly and accurately extracted from the published implementation guide. Since the ADaM Team is the best interpreter of its own intentions, it should shoulder the responsibility of authoring the implementable business requirements that the industry needs. By doing so, it retains control of what it means to be compliant with ADaM. So even before the ADaM IG was formally published, the team began work on checks to validate compliance.

The team’s primary objective was to develop a list of checks that are:

- objective (unambiguous and opinion-free)
- “machine-readable” (written in a manner which can be translated into a software expression)
- directly supported with references to the IG
- software neutral
- freely available to everyone who wants to implement ADaM

This paper discusses the significance of compliance within clinical research, what ADaM validation implies and the ADaM Team’s process to define compliance checks for analysis data.

WHAT IS COMPLIANCE?

The term compliance has become a blanket statement for a wide range of activities and process-deliverables within clinical trial research. It can imply different things depending on which particular activities or deliverables are being addressed.

In the software world, “compliance” is evaluated for all the process components which form a system, including a validated installation; the set of procedures to maintain the system; and all the supporting documentation that covers the requirements, design, implementation and ongoing maintenance. So in that usage, compliance is more about the process around the development and implementation of the software than in the technical aspects.

Within the clinical trial process, “compliance” is often associated with validation or verification and can be divided into two distinct areas. The first concerns the procedures that are used to manage the flow of information. This includes extracting data from a source system (e.g. EDC or CDMS), transforming the data into analysis-ready data, and generating tables, listings and figures. Each of these steps has its own nuances and therefore each company has its own unique process for ensuring compliance with standard procedures.

The second and most critical area subject to compliance is the data itself. Data structures must be defined according to specific criteria and the quality of the data itself must be verified. This is the type of compliance which has been addressed by CDISC.

COMPLIANCE VERSUS VALIDATION

The terms “compliance” and “validation” are used (and often misused) interchangeably within the industry. Webster’s Dictionary defines them as:

- Compliance: the act of complying with a regimen or official requirements
- Validation: the act of demonstrating that a procedure, process, and activity will consistently lead to the expected results

These two definitions are related in practice so it is understandable that there is confusion about their usage. This is particularly true when they may have different nuances depending on whether we are referring to software, processes or data, or the relationship between those components. For the purposes of this paper, we use the term “compliance” to describe whether the data meet the overall intended goals of the CDISC ADaM guidelines and “validation” to describe the specific technical exercise of checking against individual criteria.

CDISC COMPLIANCE

The history of compliance within CDISC has been a long and winding road. One of the first models developed by CDISC was ODM (Operational Data Model), which is very different from the other CDISC models. ODM is a XML based specification with very clear and machine readable rules. An ODM file can be checked as a well-formed XML file and validated against the ODM specific schema. This validation process is machine readable as it checks the structure of the ODM against the rules defined within the schema. The example below describes a rule defined within the schema for the TransactionType attribute. It must be a string, can only have the values listed and has a maximum length of 7.

```
<xs:simpleType name="TransactionType">
   <xs:restriction   base="xs:string">
     <xs:enumeration value="Insert"/>
     <xs:enumeration value="Update"/>
     <xs:enumeration value="Remove"/>
     <xs:enumeration value="Upsert"/>
     <xs:enumeration value="Context"/>
     <xs:maxLength   value="7"/>
   </xs:restriction>
</xs:simpleType>
```

Using this schema, software can validate that the ODM file follows these rules. In the example above, if the TransactionType attribute has a length of 10 characters or a value of “Change”, the validation software would generate an error when checking the file.

This type of specification is much different than the rest of the CDISC models. The challenge with the other models, such as SDTM and ADaM, is that they exist as PDF documents and not well-defined machine readable specifications that can be tested automatically against a specification document. Therefore defining validation rules for a model such as SDTM becomes a post-hoc manual effort, requiring subjective interpretation of the underlying logic within the Implementation Guide’s text.

In the case of SDTM, the initial published Implementation Guide did not include a set of validation rules. Instead, a set of rules were developed by a commercial vendor for use by the FDA as part of a Cooperative Research and Development Agreement (CRADA) to validate and review data in the SDTM format. Since the SDTM validation criteria weren't developed and maintained by the same team which developed the model, the interpretation of requirements, as expressed in the published validation checks, sometimes differs depending on who developed the checks and for what purpose they were developed. In addition, in the time since the initial set of checks was published, vendors as well as drug development companies have developed their own interpretations of the rules. These differences introduce confusion and challenges as users attempt to validate their data.

To proactively avoid these issues, the ADaM team decided to define and publish validation checks for their own model.

WHAT IS ADAM COMPLIANCE?

ADaM represents both a philosophy (focused on ensuring traceability of analytical results back through the methodology that was employed and the data that were used) and a set of rules which are designed to establish commonality in the way data are described. Both the philosophy and the rules must be met in order for data to be "ADaM compliant".

TRACEABILITY PLUS DATA CONVENTIONS

It is possible to prepare data which strictly follow the naming conventions, value relationships and other rules set forth in the ADaM documentation while failing to adequately "tell the story" of how analyses were conducted. On the other hand, the traceability aspect of ADaM may be well developed, with data collection, transformation and analysis fully described in an interconnected way, but the data are stored in a nonstandard format. While it would be nice to give points for sincerity and effort, neither of these scenarios meet the basic overall ADaM goal. Half-done is not done.

This does not mean that every ADaM-compliant data package must be identical to every other one. As the Analysis Data Model (ADaM) V2.1 points out, "The design of analysis datasets is generally driven by the scientific and medical objectives of the clinical trial." So one analysis dataset will differ from another based on the requirements of the analyses they support. However differences must still follow predictable patterns and the overall objectives (traceability and commonality) must still be met.

SUBJECTIVE VERSUS OBJECTIVE

Compliance with ADaM has both a subjective and objective component. The subjective elements are, by their nature, difficult to retrospectively assess in a systematic way. For example, ADaM 2.1, section 3 states, "Analysis datasets and their associated metadata should facilitate clear and unambiguous communication." This is inherently subjective since what is "clear and unambiguous" to one person may be "dense as dirt" to another. Similarly, ADaM IG 1.0, section 2.1 states "Analysis datasets should have a structure and content that allow statistical analyses to be performed with minimal programming." In this case, the definition of "minimal" is open to interpretation depending on someone's programming skills or their expectation of what it means for a dataset to be "analysis ready". Yet even though they are not rigorously quantifiable, principles such as these do create a reasonable expectation of what an ADaM data package should include. Based on these criteria, it may be difficult to describe what a successful ADaM implementation contains, but it should be quite easy to recognize a failed one.

With the publication of ADaM IG 1.0, we now have an extensive list of explicit, objective rules to go along with the overarching principles. These definitional guidelines include things such as:

- "ADSL contains one record per subject, regardless of the type of clinical trial design."
- "The names of date imputation flag variables end in DTF, and the names of time imputation flag variables end in TMF."
- "For subject-level character population flag variables: N = no (not included in the population), Y = yes (included). Null values are not allowed."

Compliance with many of these objective criteria can be assessed simply by examining the data. Others are situational rules which require additional knowledge to evaluate. Examples of the latter include:

- "A character indicator variable is required for every population that is defined in the statistical analysis plan."
- "ANLzzFL is a conditionally required flag to be used in addition to other selection variables when the other selection variables in combination are insufficient to identify the exact set of records used for one or more analyses."

Solely by looking at a dataset, how can we tell if a population specified in the analysis plan has been left out? How can we know what the selection variables are for a specific analysis subset, and whether they are sufficient? We must have knowledge of the analysis plan and the statistical methods which were used to fully evaluate these types of requirements.

THE GOAL OF ADAM COMPLIANCE CHECKS

The ADaM Compliance Team's fundamental goal was to create a list of compliance checks which can be implemented via common software. This was accomplished by focusing on objective criteria which can be assessed solely from datasets. For lack of a better term, we refer to these as "machine-readable checks." The question of "which machine" and "which software" isn't explicitly addressed. Instead our goal was to create generic statements that can be easily coded into any software language. SAS is commonly used within clinical research, but isn't the only software being used. And with the introduction of XML, HL7 standards and cutting edge technology, the landscape may change. With all this in mind, the checks were developed to be technology-neutral.

DEFINING THE ADAM VALIDATION RULES

This section describes the step-by-step process the ADaM team followed to define the first set of validation checks for the ADaM 2.1 document and the ADaM 1.0 IG.

TEAM PROCESS

The ADaM Team decided to define compliance rules even as the ADaM IG was being developed. The ADaM model is complex. The subjective component of ADaM and the requirement for well-defined metadata are challenging to evaluate. However the objective data requirements are fundamental and can be clearly defined. Rather than leaving an unmet need for industry-wide validation criteria, the team decided to define what is clearly understood. The rest can be addressed sometime in the future when it becomes possible to do so.

The team began developing the rules during the review cycle of the IG. However the model was continuously evolving during this review period and most of the ADaM team members were busy reviewing and updating the IG itself. This made validation rules a low priority and the team put the project on hold. After the IG was finalized at the end of 2009, the team restarted the validation project and a small sub-team was formed to create an ADaM compliance document.

EXTRACTING RULES

The first step was to perform a thorough review of the ADaM documents to identify any text which could be translated into a rule. The sub-team divided the various sections of the document among the members and began documenting the rules.

TO TEST OR NOT TO TEST?

The sub-team ended up with over 350 rules from this first effort. At first glance this seemed a bit overwhelming. However as the sub-team began the review process they quickly realized that not all checks could be implemented with a machine; many were clearly subjective and so required human interpretation. Examples of machine-testable and non-testable checks are:

- *Non-testable: Analysis datasets and their associated metadata should facilitate clear and unambiguous communication*
- *Testable: All ADaM variable labels must be no more than 40 characters in length*

The sub-team made the decision early in the process to only keep checks that could be tested by a machine and therefore implemented by common software. The sub-team performed a second review eliminating the rules which were non-testable. For example, ADaM states, "*For compound criterion rows, the value of PARAMTYP must be 'DERIVED'".* We can check whether PARAMTYP is equal to DERIVED but there isn't an automated way to evaluate whether a row involves compound criteria. So rules like this were removed from the list.

As the sub-team worked through this process, many rules were identified which were based on common sense but could not be directly referenced in the ADaM documents. An example of such a common-sense rule is:

> *For all TRTxx variables there must be a set of variables named such that "xx" iterates continuously and sequentially from 1 to highest number.*

This rule is logical but it is not explicitly listed within the ADaM IG. While it could be useful to include rules such as this, the sub-team decided to only include rules which were clearly documented within ADaM. Even though we did not include these implied rules in the final validation document, we did document them for the ADaM Team to consider when the IG is updated in the future.

CLARITY OF RULES

The sub-team discussed what the general tone and grammatical structure of the checks should be. One choice was to have both a short description of the logic as well as a software code fragment. We decided to only include the description of the logic to support the team's software-neutral approach.

Many of the rules in the ADaM IG describe relationships between variables and their valid values. The nuances of the requirements can be complex. If these are expressed as a single expression which includes all the requirements, the machine-readable rule would be complex and potentially unclear. The team chose to write clear, specific checks which deal with the individual pieces of a complex rule, even though this results in more checks. A common example is the flag-variable rule from the ADaM IG, which requires the value of numeric flag variables (*FN) to translate directly to the value of character flag variables (*FL):

> **FN and *FL must be a one-to-one mapping*

It may be possible to check this condition, and all that it implies, all at once in a single complicated code statement. However, while the nuances may be implied, they aren't explicitly defined. Therefore, this example from the IG was split into five separate checks:

- There is more than one value of a variable with a suffix of FN for a given value of a variable with the same root name and a suffix of FL
- There is more than one value of a variable with a suffix of FL for a given value of a variable with the same root name and a suffix of FN
- A variable with a suffix of FL is equal to Y and a variable with the same root and a suffix of FN is not equal to 1
- A variable with a suffix of FL is equal to N and a variable with the same root and a suffix of FN is not equal to 0
- A variable with a suffix of FL is equal to null and a variable with the same root and a suffix of FN is not equal to null

Note that the sub-team wrote the checks "in the negative". This means if the check is met, it fails validation. This is consistent with how data checks are often written for software, data management systems and other clinical systems. So in the example above, if the data meet any of the conditions, it fails ADaM validation.

Each rule is clearly defined and can be checked with a straightforward code statement. Taken as a group, they validate compliance with the IG.

"ADAM IN A BOX"

ADaM requires analysis data to be consistent with SDTM data, and its metadata must be properly represented in define.xml. The sub-team had lengthy discussions about whether the scope of the ADaM validation checks should include cross-model checks.

Examples of ADaM requirements that relate to SDTM include:

- *If SDTM character variables are converted to numeric variables in ADaM datasets, then they should be named as they are in the SDTM with an "N" suffix added. If necessary to keep within the 8-character variable name length limit, the last character may be removed prior to appending the N.*
- *ADSL variables STUDYID, USUBJID, SUBJID, SITEID, AGE, AGEU, SEX, RACE, ARM must match SDTM variable in metadata and values.*

The first example requires us to know whether a variable is intended to be a numeric representation of another variable. This thought process cannot be tested with a machine so falls outside the scope of this version of the checks. However the second example can be tested if the SDTM data are available. In general, the sub-team limited checks against SDTM to those that could be clearly and simply defined. The sub-team ended up with a handful of rules involving SDTM that included verifying:

- identical metadata across variables with the same name (e.g. STUDYID)
- values of USUBJID and SUBJID match values in the SDTM DM domain
- values of SRCDOM reference a valid SDTM domain

The issue becomes more complex with respect to validating ADaM against define.xml. Requirements include:

- *The ordering of the variables within a dataset should match the order of the variables presented in the define file*
- *An ADaM variable described in define.xml must be included in the dataset*

While it is possible to perform these types of checks and they would be useful, they cannot be done solely with ADaM datasets. Introducing them would greatly increase the complexity of the checks and expand the scope of the project to encompass compliance with all interrelated CDISC models. In the end, the team deferred most of these cross-model issues to a future version.

RULE METADATA

A common feature of published CDISC SDTM checks is that they are assigned severity levels. For example, one SDTM validation tool rates the significance of a rule as low, medium or high, based on a subjective assessment of whether the error can potentially affect the use of the data. We originally set out to assign severity levels in a similar fashion. For example, the ADaM rule, "When merging data from ADSL into other analysis datasets, only those fields relevant to these analysis datasets should be included," probably is of low overall importance with respect to the validity of the data and the ability to recreate analyses. We also talked about features which the data “must have” and “should have” versus those that are “nice to have.” However once the list of rules had been pared down to ones that are strictly objective, machine-testable and solely supported by the available data, we found that everything that remained represented black-and-white, pass/fail criteria. Within this list, data are either compliant with the ADaM guidance or not. If users choose to deviate from the ADaM rules, the resulting datasets may be useful for their purposes but they are not ADaM compliant. Therefore the idea of severity was not included in this version of the checks.

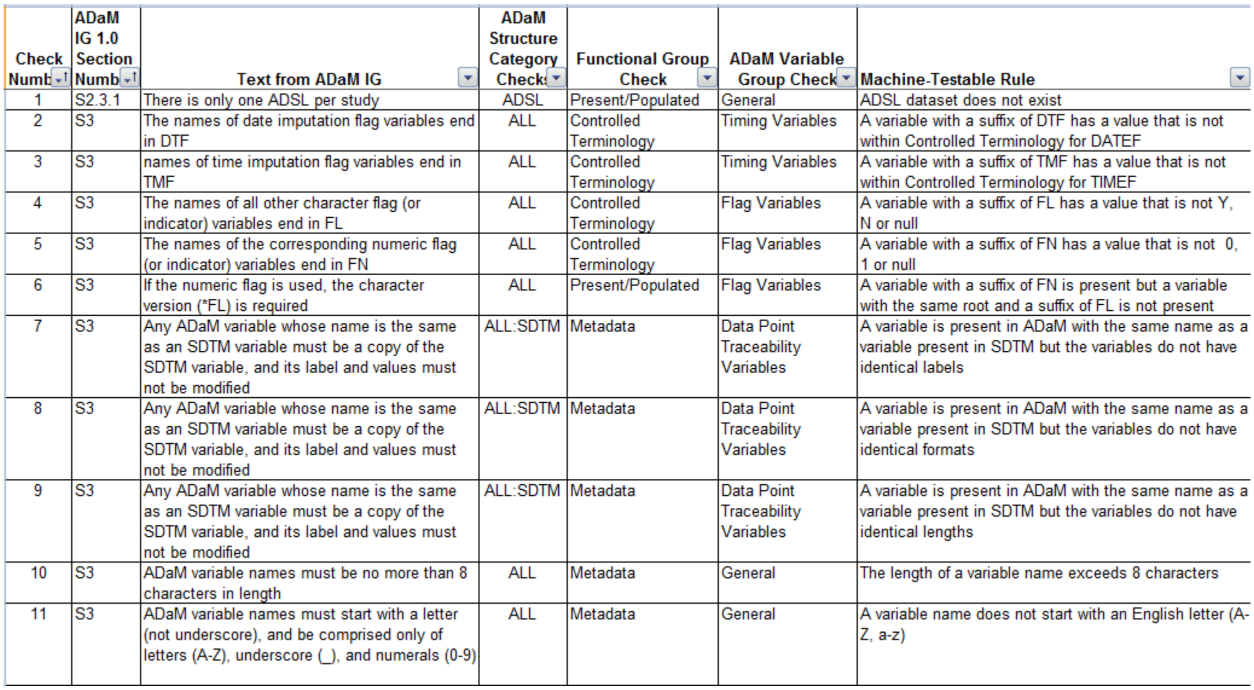
Although the checks weren’t assigned severity levels, the sub-team did recognize the value of putting checks into useful categories. Several schemes were discussed, and the sub-team settled on three types of categorizations:

- ADaM Structure Category: groups checks based on the ADaM structure that the requirement relates to. Categories include checks that apply to all ADaM structures, ADSL only, BDS only, or the relationship of one structure to another (e.g. ADSL:BDS).
- Functional Group: groups checks based on the nature of the requirement. Categories include metadata, consistency of values, the presence of a variable, controlled terminology and valid values.
- ADaM Variable Group: groups checks based on the type of variable covered by the requirement, based on the section titles in the IG. Examples include Study Identifiers, Timing Variables and Treatment Variables.

These different categorizations allow users and software developers to focus on specific types of validation rules, including or excluding checks which aren’t of interest to them at a particular stage of development. For example, users may choose to ignore Controlled Terminology checks if data haven’t been “cleaned” yet.

REVIEW CYCLE

Once the sub-team agreed on the final scope and content, a spreadsheet was created which included all the checks and associated metadata. This was reviewed by both the full ADaM Team and the CDISC Technical Leadership Committee. The sub-team evaluated the feedback from those groups and incorporated it as appropriate. A snapshot of the spreadsheet is shown below.



Check Numb	ADaM IG 1.0 Section Numb	Text from ADaM IG	ADaM Structure Category Check	Functional Group Check	ADaM Variable Group Check	Machine-Testable Rule
1	S2.3.1	There is only one ADSL per study	ADSL	Present/Populated	General	ADSL dataset does not exist
2	S3	The names of date imputation flag variables end in DTF	ALL	Controlled Terminology	Timing Variables	A variable with a suffix of DTF has a value that is not within Controlled Terminology for DATEF
3	S3	names of time imputation flag variables end in TMF	ALL	Controlled Terminology	Timing Variables	A variable with a suffix of TMF has a value that is not within Controlled Terminology for TIMEF
4	S3	The names of all other character flag (or indicator) variables end in FL	ALL	Controlled Terminology	Flag Variables	A variable with a suffix of FL has a value that is not Y, N or null
5	S3	The names of the corresponding numeric flag (or indicator) variables end in FN	ALL	Controlled Terminology	Flag Variables	A variable with a suffix of FN has a value that is not 0, 1 or null
6	S3	If the numeric flag is used, the character version (*FL) is required	ALL	Present/Populated	Flag Variables	A variable with a suffix of FN is present but a variable with the same root and a suffix of FL is not present
7	S3	Any ADaM variable whose name is the same as an SDTM variable must be a copy of the SDTM variable, and its label and values must not be modified	ALL:SDTM	Metadata	Data Point Traceability Variables	A variable is present in ADaM with the same name as a variable present in SDTM but the variables do not have identical labels
8	S3	Any ADaM variable whose name is the same as an SDTM variable must be a copy of the SDTM variable, and its label and values must not be modified	ALL:SDTM	Metadata	Data Point Traceability Variables	A variable is present in ADaM with the same name as a variable present in SDTM but the variables do not have identical formats
9	S3	Any ADaM variable whose name is the same as an SDTM variable must be a copy of the SDTM variable, and its label and values must not be modified	ALL:SDTM	Metadata	Data Point Traceability Variables	A variable is present in ADaM with the same name as a variable present in SDTM but the variables do not have identical lengths
10	S3	ADaM variable names must be no more than 8 characters in length	ALL	Metadata	General	The length of a variable name exceeds 8 characters
11	S3	ADaM variable names must start with a letter (not underscore), and be comprised only of letters (A-Z), underscore (_), and numerals (0-9)	ALL	Metadata	General	A variable name does not start with an English letter (A-Z, a-z)

Once the internal CDISC review cycle was complete, the sub-team wrote a document to explain the context of the validation spreadsheet and how it should be used. It also includes the scope and a description of the categories.

A package which included the ADaM Validation Checks and a comment spreadsheet was published for public review the first week of July 2010. Comments were received up to the end of August and are now under review. The goal is to publish the first production version of the AdaM Validation Checks by the end of September 2010.

The sub-team realizes that the first release is limited to only the clearly defined machine-readable checks and does not take into account the subjective nature of ADaM compliance or the more complex checks across models. Once it is published, work will begin on additions, improvements and clarifications.

CONCLUSION

People who create and work with analysis data have often suggested that defining standards for analysis data is very difficult if not impossible. While ADaM falls short of solving all the diverse analysis needs within the industry, it is a good foundation that can help us begin to define more structured analysis data. With the release of ADaM version 1.0, the industry had a foundation of best practices to implement analysis standards. However, they did not really have an implementable, unambiguous model. With the release of a more mature version 2.1 as well as a supporting Implementation Guide, a concrete model now exists that can be implemented.

By developing over 170 machine-readable validation checks for ADaM, the team has shown that the model is practical, predictable and ready to be implemented within your organization today. It is not just a collection of philosophical guidelines but is a well-defined set of rules and data structures. One caveat is that ADaM still offers a significant amount of flexibility and is expandable, as needed. So as people implement ADaM within their organizations, they'll want to add additional validation checks to cover their specific extensions.

ADaM compliance not only includes the validation checks published by the ADaM team but also encompasses all the other components that are needed for well-formed analysis data. Specifically within ADaM, complete and accessible metadata is required to have compliant analysis data.

REFERENCES

- ADaM Implementation Guide 1.0, December, 2009. www.cdisc.org.
- ADaM 2.1 document, December, 2009. www.cdisc.org.

ACKNOWLEDGMENTS

The authors would like to acknowledge the sub-team who made significant contributions of time and expertise to delivering a quality product in a timely fashion. The compliance sub-team included ADaM team members Randall Austin, Sandy Chang, Chris Decker, Nate Freimark, Monika Kawohl, Geoff Mann, Kim Minkalis, Terek Peterson, Jack Shostak, and Dave Smith.

CONTACT INFORMATION

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

Author Name: Chris Decker
Company: d-Wise Technologies
Address: 4020 Westchase Blvd Suite 527 Raleigh, NC 27607
Work Phone: 919-600-6234
Email: cdecker@d-wise.com
Web: www.d-wise.com

Author Name: Randall Austin
Company: GlaxoSmithKline
Address: P.O. Box 13398 Research Triangle Park, NC 27709
Work Phone: 919-483-1379
Email: randall.r.austin@gsk.com