

ADaM Compliance Starts with ADaM Specifications

Trevor Mankus, Kent Letourneau, PRA Health Sciences

ABSTRACT

As of December 17th, 2016, the FDA and PMDA require that all new studies included in submissions have their analysis data sets created in compliance with the CDISC ADaM standards. The possibility of having your submission not accepted heightens the importance of having an effective process for ensuring that you are following this standard. At PRA Health Sciences, this process starts with a review of the metadata in our ADaM specifications. This paper gives background on this topic and describes the process by which we create data sets that are fully compliant with the ADaM standards. We discuss that determining ADaM compliance cannot only be done with available validation software tools and that a human component is needed. We will also discuss the benefits of beginning this review on the ADaM specifications rather than on the data sets.

INTRODUCTION

Assessing ADaM compliance is not a simple task. In today's fast paced and timeline-centric landscape, finding areas where companies can identify and build efficiencies in their existing processes which reduce development costs is a primary focus. Readily available validation software is typically used at the end of programming efforts. Any data anomalies which lead to non-compliance with the CDISC standards are typically met with resistance due to the large amount of re-work required to adequately correct the non-conformance. What if we change our mindset and begin looking at ADaM compliance earlier on in the process?

The CDISC ADaM Team released validation checks to help assess ADaM compliance. Version 1.3 of the validation check document contains 319 rules. Each validation check is able to be implemented with software to test the rules defined within the ADaM Implementation Guide v1.0, Data Structure for Adverse Events, and The ADaM Basic Data Structure for Time-to-Event Analyses. Therefore, it is also safe to assume that at least some of the validation rules defined by the CDISC ADaM Team can be tested against ADaM *metadata* only.

PROGRAMMING PROCESS AND WORK FLOW

Typically, the programming process for producing tables, listings, figures, and a clinical study report begins with mapping the collected data into SDTM. After the SDTM data sets are finalized, best practice is to run them through validation software to identify areas of CDISC non-compliance. If there are issues which need correcting, the SDTM data sets are revised to address the error(s). If there are no issues, then ADaM programming may begin. Likewise, once ADaM data sets are finalized they also go through validation software and any identified areas of concern are addressed. This traditional process flow is outlined in Figure 1.

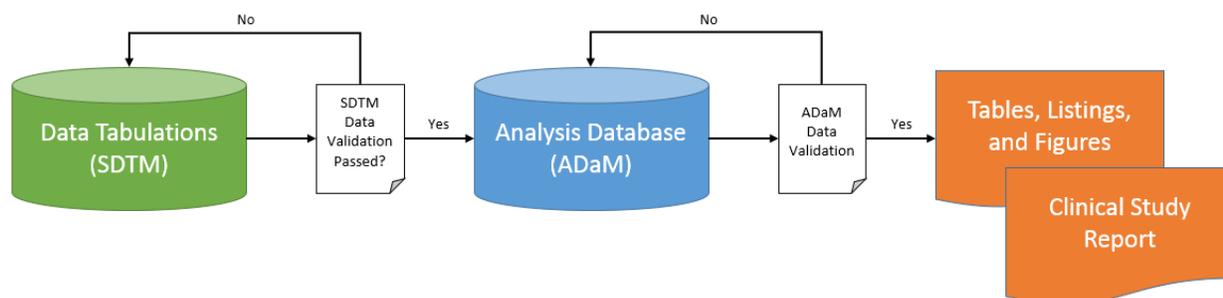


Figure 1. Traditional Work Flow

The primary issue with this traditional work flow is that ADaM validation via software tools can only be done once all of the ADaM data sets are finalized. If there are any findings during validation, then the programming team needs to go back and address these issues. This often leads to a large amount of re-work for two reasons: (1) issues may exist for ADSL which leads to all other non-ADSL data sets having to be recreated; and (2) programming of tables, listings, and figures which has started may be impacted if the structure of the ADaM data sets are being modified.

One solution to this issue is to introduce a new step in the process flow. While it might seem like adding an additional step to the overall process can be detrimental to the overall time required, it actually can be beneficial by reducing the number of potential findings at the ADaM validation stage once all data sets are finalized. This new step involves the review of ADaM specifications specifically focusing on CDISC compliance (see Figure 2).

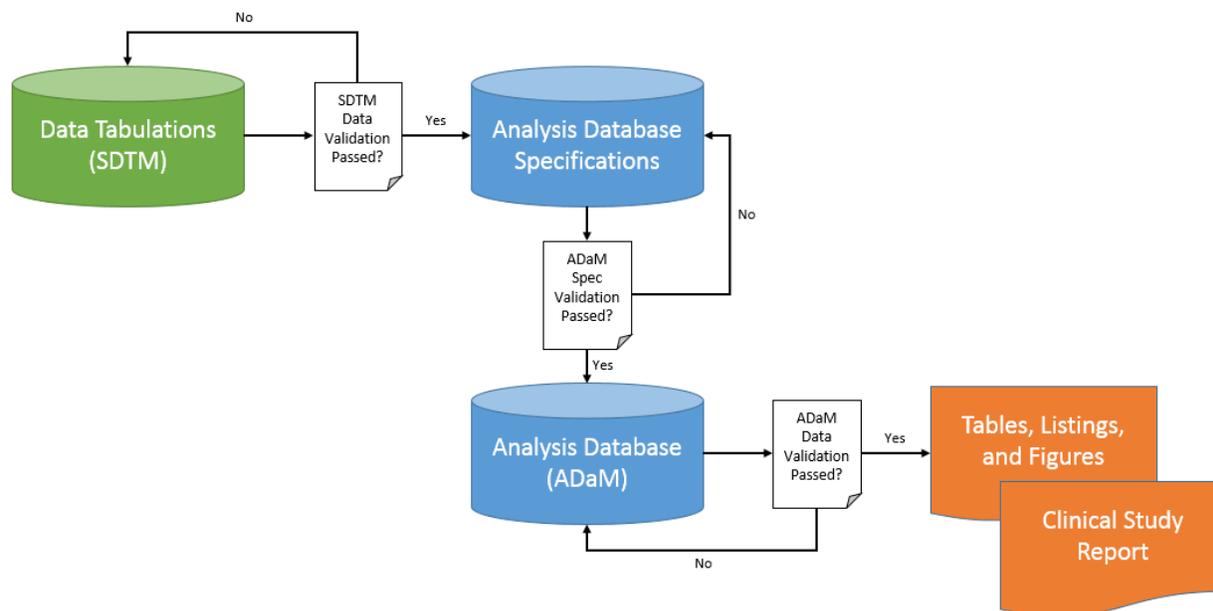


Figure 2. Revised Work Flow

By adding a review of the ADaM specs, you can begin to check for compliance against a subset of the CDISC ADaM rules earlier in the work flow which should result in fewer findings at the final review once the ADaM data sets are finalized.

IDENTIFYING WHICH CHECKS ARE BASED ON METADATA VS. DATA

As stated previously, the CDISC ADaM Team currently maintains a library of 319 validation rules. These rules are organized by ADaM Structure Groups (ALL, ADSL, ALL:SDTM, BDS, etc.), Functional Groups (controlled terminology, metadata, present/populated, etc.), and ADaM Variable Groups (general, ADSL, ADAE, flag variables, etc.). Some of these checks are executed by looking at the *values* within the ADaM dataset (see Table 1), but many of them only look at the *structural metadata* (see Table 2).

We can begin to execute the checks which are based on structural metadata before any ADaM data sets are created simply by comparing them against the ADaM specifications.

ADaM Structure Group	Functional Group	ADaM Variable Group	Machine-Testable Failure Criteria
ADSL	Controlled Terminology	Population Indicator(s)	SAFFL is present and has a value that is not Y or N
BDS	Value Consistency	Flag Variables	BASETYPE is present, BASE is populated, and BASE is not equal to AVAL where ABLFL is equal to "Y" for

			a given value of PARAMCD and Basetype for a subject
ALL	Value Consistency	Timing Variables	The value of a variable with a suffix of SDT is greater than the value of a variable with the same root and a suffix of EDT

Table 1. ADaM Rules Based on Data Values

ADaM Structure Group	Functional Group	ADaM Variable Group	Machine-Testable Failure Criteria
ALL	Metadata	General	The length of a variable name exceeds 8 characters
ALL	Metadata	General	The length of a variable label is greater than 40 characters
ALL	Metadata	Timing Variables	A variable with a suffix of DT does not have a SAS Date format
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
BDS	Metadata	Treatment Variables	A variable with a prefix of TRTAG has y fragment appended after TRTAG that is not a single-digit integer [1-9]

Table 2. ADaM Rules Based on Metadata

Identifying which ADaM rules can be programmatically tested against your ADaM specifications depends on what type of metadata is collected in the specifications themselves. At a minimum, ADaM specifications should include variable name, label, type, length, display format, controlled terminology, and a derivation or programming comment. If these metadata fields are included in the ADaM specifications, then we found that about 139 out of the 319 (~44%) checks can be validated against just based on the metadata alone which can provide a substantial jump start on ADaM validation.

THE ADAM SPECIFICATION REVIEW PROCESS

Introducing a new step to the programming process flow introduces a risk of interrupting or delaying study timelines unless the defined process is streamlined for efficiency. Streamlining the process includes automating the request, building tools to help complete the task, and providing a means of communication that is effective enough to share the nuances of the study between the study team and the reviewer.

At PRA Health Sciences, our process is built around the Microsoft SharePoint platform. This solution provides us the flexibility we need to ensure the system meets our requirements but also allows us to maintain it ourselves within our department.

REQUESTING A REVIEW

The process begins with the study team lead requesting a review by one of our Subject Matter Experts (SMEs). This request is made via a custom SharePoint application we built. The request form (see Display 1) is completed online and submitted electronically.

PART 1 Request Your Review <i>(Completed by PRA data standards users)</i>	
Study ID:	<input style="width: 100%;" type="text"/>
Study Phase:	<input style="width: 100%;" type="text"/>
Study Indication:	<input style="width: 100%;" type="text"/>
Databasics Time Code:	<input style="width: 100%;" type="text"/> Time spent reviewing ADaM specifications will be billed to your project.
Requested By:	<input style="width: 100%;" type="text"/> Enter your PRA username
ADaMIG Version:	<input style="width: 100%;" type="text"/>
Spec Name and Location:	<input style="width: 100%;" type="text"/> Name and network location to ADaM specification (e.g., \\NA1SASFILE1\...)
SDTM Data Location:	<input style="width: 100%;" type="text"/> Network location of the study SDTM data, if available (e.g., \\NA1SASFILE1\...) <input type="checkbox"/> Unblinded / Restricted Additional user access may be required
Date Available:	<input style="width: 100%;" type="text"/> Enter the date that the ADaM specification will be available for review
Date Completed By:	<input style="width: 100%;" type="text"/> Enter the date you need to have feedback returned

Display 1. SharePoint Review Request Form

Once the request form is submitted, an automated email alert is generated by the SharePoint system and immediately sent to our internal Subject Matter Expert distribution list. This allows our SMEs to get notified of the request so they can assess their time and availability.

ASSIGNING A SUBJECT MATTER EXPERT (SME)

Based on the information provided in the request, our team of SMEs then decides who will complete the review. Some of the information provided helps us decide who is best suited to review the ADaM specifications for that study team. Other information on the request may be used during the review itself. See Table 3 for more information on how each field is used.

Field(s)	Description
Study ID, Study Phase, Study Indication	These fields are used to assign a SME which has experience in the study phase or indication.
ADaMIG Version	This field lets the SME know which version of the ADaM Implementation Guide was used so they know what rules should be followed.
Date Available, Date Completed By	These dates help us assign a SME who is available to support the request within the allotted timeframe.
SDTM Data Location	Allows our SME to include checks which are based on the SDTM:ADaM relationship.

Table 3. Request Form Fields

Once a SME is assigned, the ADaM specification review begins.

USING SAS® TO SUPPORT THE ADAM SPECIFICATION REVIEW

In order to build a macro to validate ADaM specifications, we needed at least two libraries of metadata: (1) the CDISC ADaM metadata and (2) the study-specific metadata from the ADaM specification. A third

library (SDTM data) is also useful as it enables additional validation based on the relationship between SDTM and ADaM metadata. Figure 3 shows the different input metadata libraries.

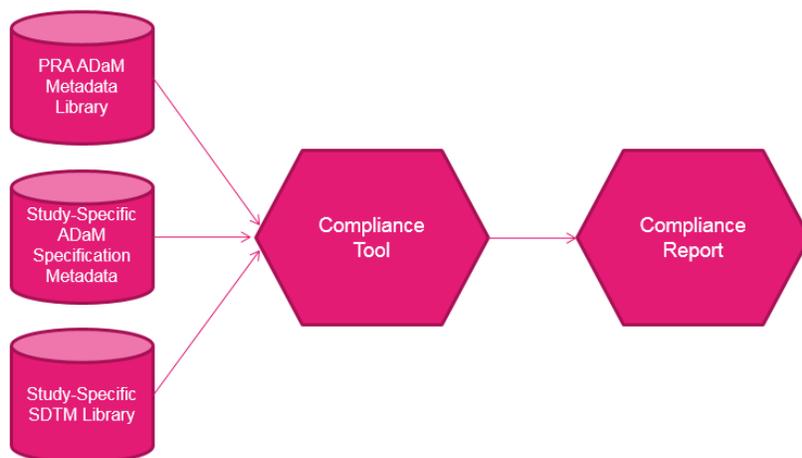


Figure 3. Compliance Tool Process Flow

For the purposes of this paper, the PRA ADaM Metadata Library in Figure 3 is the ADaMIG metadata in Excel format. This acts as the “baseline” which we compare the study-specific ADaM metadata against.

Programming Code and Logic

Once all of the required metadata libraries are established, the SME can make a SAS® program which contains a call to the validation macro. A sample macro call might look like this:

```

%g_sADaM_DDTValidator(projid    = PRA-TEST-001,
                      admslib   = \\na1sasfile1\PRA\PRA-TEST-001\ADaM\Data,
                      admsnm    = adms,
                      template  = \\ADaM DDT Review\ADMS-DDT.xlsx,
                      adsl      = ADSL,
                      occds     = ADAE ADCM ADMH,
                      bds       = ADEG ADLB ADVS,
                      sdtmlib   = \\na1sasfile1\PRA\PRA-TEST-001\SDTM\Data,
                      out       = \\Applications\ADaM DDT Review);
  
```

The PROJID macro parameter is used for naming the report file and header/footer information embedded in the output. The next two macro parameters are used to provide information about where the study-specific ADaM specification metadata can be found. At PRA Health Sciences, the ADaM specification is converted into a SAS data set (adms.sas7bdat) as part of our normal programming process. We found that this data set could additionally be used to support this process. The TEMPLATE macro parameter is used to provide information about where the standard ADaM metadata can be found. As previously mentioned, this is essentially the ADaMIG converted into Excel. The ADSL, OCCDS, and BDS macro parameters are used to group the ADaM data sets defined in the specification into data set classifications. ADaM has three structures: ADSL, BDS, and OCCDS. Different ADaM compliance rules exist for each data set class. The SDTMLIB macro parameter provides the path of where the SDTM data can be found (if available). This is important because there are ADaM validation rules which are specific to the relationship between SDTM and ADaM. For example, ADaM adheres to a principle of harmonization known as “same name, same meaning, same values”. This means if a variable exists in both SDTM and ADaM, portions of the variable level metadata should not be modified. Lastly, the OUT macro parameter provides a path of where the permanent report file will be saved once generated.

The macro itself contains checks for all of the ADaM rules which are based on metadata only. Here is an example of a check which compares the variable data types between the ADaM specification and the ADaMIG:

```

if ^missing(DDT_Name) and strip(upcase(Type))^=strip(upcase(DDT_Type)) then do;
  %add_error(id=PRA3003, dataset=Dataset, variable=Name, value=Type, expected=DDT_Type);
end;

```

Here is an example of a check which compares the variable labels across SDTM and ADaM for variables which exist in both the SDTM data and the ADaM specification:

```

%if &sdtmlib^= %then %do;
  if ^missing(SDTM_Label) and strip(Label)^=strip(SDTM_Label) then do;
    %add_error(id=PRA0001, dataset=Dataset, variable=Name, value=Label, expected=SDTM_Label);
  end;
%end;

```

Summarizing the Results

The validation utility writes out a report file in the location specific in the OUT macro parameter. The report file is a Microsoft Excel formatted workbook. To create it, we use ODS TAGSETS.EXCELXP to wrap the PROC REPORT code. The ExcelXP tagsets allow you to customize the way the output file is formatted to make it very user friendly. Our report file has embedded titles, footnotes (including Page x of y), printable worksheets, auto-filter applied to the header rows, and frozen rows to help users quickly and easily navigate through their findings.

Our report file contains three worksheets. The first is a summary of the compliance findings. It provides a count of how many instances of a given validation rule were violated. See Display 2 for an example of what this worksheet looks like.

	A	B	C	D
1	PRA Health Sciences			
2	PRA Project ID: PRA-TEST-001			
3	Global Data Standards ADaM Specification QC Report v1.0			
4	ADSL: ADSL OCCDS: ADAE ADCM ADMH BDS: ADEG ADLB ADVS			
5	Generated by MankusTrevor on 28SEP2016 at 12:23 (EST)			
6	Report File: \\Applications\ADaM DDT Review\PRA-TEST-001_GDS_ADaM_Spec_Validation_Report_28SEP2016.xlsx			
7	Configuration File: GDS_ADaM_Validation_Checks_v1.3.xlsx			
8				
9	Summary of Findings			
10				
11	Rule Information			
12	ID	Description	Category	Issues Found
13	PRA0001	A variable is present in ADaM with the same name as a variable present in SDTM but the variables do not have identical labels	Metadata	5
14	PRA0003	A variable is present in ADaM with the same name as a variable present in SDTM but the variables do not have identical lengths	Metadata	14
15	PRA0009	Labels for ADaM variables do not match the standard labels for ADaM variables listed in the implementation guide that cannot be modified (with the exception of 1) variables whose names contain indexes y, xx, or zz and (2) variable labels with asterisks (*) and ellipses (...) indicated for sponsor appropriate text)	Metadata	83
16	PRA0022	A variable is present with the same name as a variable present in ADSL but the variables do not have identical labels	Metadata	1
17	PRA0023	A variable is present with the same name as a variable present in ADSL but the variables do not have identical formats	Metadata	4
18	PRA0024	A variable is present with the same name as a variable present in ADSL but the variables do not have identical lengths	Metadata	1
19	PRA1001	A variable with a suffix of FL has a value that is not Y, N or null	Controlled Terminology	434
20	PRA1003	COMPLFL is present and has a value that is not Y or N	Controlled Terminology	1
21	PRA1007	SAFFL is present and has a value that is not Y or N	Controlled Terminology	1
22	PRA1017	A variable with a suffix of RFL has a value that is not Y or null	Controlled Terminology	22
23	PRA1018	A variable with a suffix of PFL has a value that is not Y or null	Controlled Terminology	22
24	PRA2003	SITEID is not present within ADSL	Presence	1

Display 2. Summary Worksheet in Compliance Report Output

The next worksheet in the report file provides a more detailed look (see Display 3). This worksheet shows an item level summary of each rule that fired during validation. Each row includes the offending dataset, variable, value from the ADaM specification, and expected value it is comparing against (if applicable to that rule). Users can filter on any of the column headers to quickly and easily navigate down to a specific item if they desire.

ADaM Compliance Starts with ADaM Specifications, continued

ID	Description	Category	Dataset(s)	Variable(s)	Value	Expected
13	A variable is present in ADaM with the same name as a variable present in SDTM but the variables do not have identical labels	Metadata	ADAE	AECAT	Category of Adverse Event	Category for Adverse Event
14	A variable is present in ADaM with the same name as a variable present in SDTM but the variables do not have identical labels	Metadata	ADAE	AEENDTC	Analysis End Date/Time	End Date/Time of Adverse Event
19	A variable is present in ADaM with the same name as a variable present in SDTM but the variables do not have identical labels	Metadata	ADAEMQ	AECAT	Category of Adverse Event	Category for Adverse Event
20	A variable is present in ADaM with the same name as a variable present in SDTM but the variables do not have identical labels	Metadata	ADAEMQ	AEENDTC	Analysis End Date/Time	End Date/Time of Adverse Event
25	Labels for ADaM variables do not match the standard labels for ADaM variables listed in the implementation guide that cannot be modified (with the exception of 1) variables whose names contain indexes y, xx, or zz and (2) variable labels with asterisks (*) and ellipses (...) indicated for sponsor appropriate text	Metadata	ADCM	ANL01FL	Analysis Flag - Original Data Structure	Analysis Flag 01 <insert text>
26	Labels for ADaM variables do not match the standard labels for ADaM variables listed in the implementation guide that cannot be modified (with the exception of 1) variables whose names contain indexes y, xx, or zz and (2) variable labels with asterisks (*) and ellipses (...) indicated for sponsor appropriate text	Metadata	ADCM	ONTRTFL	On-Treatment Flag	On Treatment Record Flag
27	Labels for ADaM variables do not match the standard labels for ADaM variables listed in the implementation guide that cannot be modified (with the exception of 1) variables whose names contain indexes y, xx, or zz and (2) variable labels with asterisks (*) and ellipses (...) indicated for sponsor appropriate text	Metadata	ADCM	PREFL	Pre-Treatment Flag	Pre-treatment Flag
28	Labels for ADaM variables do not match the standard labels for ADaM variables listed in the implementation guide that cannot be modified (with the exception of 1) variables whose names contain indexes y, xx, or zz and (2) variable labels with asterisks (*) and ellipses (...) indicated for sponsor appropriate text	Metadata	ADEG	ADTM	Analysis Date/Time	Analysis Datetime

Display 3. Details Worksheet in Compliance Report Output

In Display 3, row #13 shows that the label for ADAE.AECAT is incorrect. The ADaM specification has “Category of Adverse Event”, but the correct label should be “Category for Adverse Event”. Row #28 shows that the label of ADEG.ADTM is incorrect. The ADaM specification has “Analysis Date/Time”, while the correct label per the ADaMIG is “Analysis Datetime”.

The final worksheet in the output file is called Non-Standard Metadata (see Display 4). This worksheet shows all of the variables in the ADaM specification which are not defined in the ADaMIG. This tab requires a manual review, as two things need to be checked: (1) has a custom variable been created for a purpose which should have instead used a standard variable name; and (2) is the name of a custom variable adhering to the variable naming rules and conventions defined in the ADaMIG.

Dataset Class	Dataset	Variable	Label	
40	OCCURRENCE DATA STRUCTURE	ADAE	ANTHYPFL	Antihypertensive Flag
78	OCCURRENCE DATA STRUCTURE	ADAE	MENIMPFL	Pre-existing Mental Impairment Flag
101	OCCURRENCE DATA STRUCTURE	ADAE	TRNPRFL	Treated but no Procedure Flag
218	OCCURRENCE DATA STRUCTURE	ADCM	ANTHYPFL	Antihypertensive Flag
263	OCCURRENCE DATA STRUCTURE	ADCM	HYPERFL	Arterial Hypertension Flag
265	OCCURRENCE DATA STRUCTURE	ADCM	MENIMPFL	Pre-existing Mental Impairment Flag
272	OCCURRENCE DATA STRUCTURE	ADCM	PRSPTRFL	Protocol Specified Treatment Flag
289	OCCURRENCE DATA STRUCTURE	ADCM	TRNPRFL	Treated but no Procedure Flag
298	BASIC DATA STRUCTURE	ADEG	ANTHYPFL	Antihypertensive Flag
333	BASIC DATA STRUCTURE	ADEG	HYPERFL	Arterial Hypertension Flag
335	BASIC DATA STRUCTURE	ADEG	MENIMPFL	Pre-existing Mental Impairment Flag
357	BASIC DATA STRUCTURE	ADEG	TRNPRFL	Treated but no Procedure Flag
770	OCCURRENCE DATA STRUCTURE	ADMH	ANTHYPFL	Antihypertensive Flag
802	OCCURRENCE DATA STRUCTURE	ADMH	HYPERFL	Arterial Hypertension Flag
804	OCCURRENCE DATA STRUCTURE	ADMH	MENIMPFL	Pre-existing Mental Impairment Flag
832	OCCURRENCE DATA STRUCTURE	ADMH	TRNPRFL	Treated but no Procedure Flag
903	SUBJECT LEVEL ANALYSIS DATASET	ADSL	ANTHYPFL	Antihypertensive Flag
945	SUBJECT LEVEL ANALYSIS DATASET	ADSL	HYPERFL	Arterial Hypertension Flag
947	SUBJECT LEVEL ANALYSIS DATASET	ADSL	MENIMPFL	Pre-existing Mental Impairment Flag
968	SUBJECT LEVEL ANALYSIS DATASET	ADSL	TRNPRFL	Treated but no Procedure Flag

Display 4. Non-Standard Metadata Worksheet in Compliance Report Output

Display 4 shows custom variables which end in *PFL and *RFL, which in ADaMIG v1.1 have a meaning of parameter-level flags and record-level flags. In this study, these variables are neither parameter-level or record-level. This may be something which was done in error by the study team which they wish to fix.

This worksheet can also be used to look for common trends within a sponsor or compound which could possibly be used to help further define their standard. For example, if we find that study teams are coming up with a different custom variable name for the same purpose, it may be worth standardizing so it is consistently named for all studies moving forward.

ADAM COMPLIANCE THAT IS NOT MACHINE TESTABLE

While software tools can be extremely helpful in assessing ADaM compliance, they need to be supplemented somehow. ADaM compliance is more than just machine testable logic. It requires expertise and manual review of metadata and data. Below are some examples of items which should be reviewed manually because they are not machine testable:

- Each ADaM data set has a Structure defined which specifies the proper structure of the data set.
- Each ADaM data set has Key Variables defined which specify the variables needed to identify a unique record in the dataset.
- Variable derivations and comments are formatted properly for the Define-XML standard.
- ASEQ is present and defined appropriately to support data point traceability for ADaM data sets which are used as the source for other ADaM data sets.
- SRCDOM, SRCVAR, and/or SRCSEQ are present and defined appropriately to support data point traceability for BDS datasets which contain records from more than one source.

Verifying the ADaM data sets are structured in such a way to support metadata traceability and data point traceability is likely the primary fundamental principle which is not machine testable. Assessing compliance means putting yourself in the position of the statistical reviewer themselves and ensuring the data you plan to present is clearly understandable. Traceability facilitates transparency, which is an essential component in building confidence in a result or conclusion. Given the definition, it is not easily machine testable because of the complex scenarios which are required to support the planned analyses.

CONCLUSION

In summary, waiting until ADaM data sets are finalized before beginning to assess CDISC compliance often leads to re-work which is inefficient in both time and cost. You can actually begin to assess compliance on a subset of the ADaM rules which are based on structural metadata only much earlier on in the programming work flow.

Developing an automated ADaM specification review process not only helps reduce the re-work and increase efficiencies, but also allows you to begin building metrics so you can look at trending non-compliance amongst your organization. Being able to identify these trending issues provides insight into areas where your team may benefit from additional training or supplemental standards material.

If the ADaM specification process is built and executed properly, your organization should start seeing a decrease in the amount of time spent programming each ADaM dataset (due to less re-work) and also begin to see less findings come out of the validation software which they may be using on a regular basis. Ultimately, high quality ADaM specifications lead to more efficient programming, higher quality ADaM data sets, and proper define.xml deliverables.

Lastly, it is important to remember that ADaM compliance is more than just machine testable logic. It requires expertise and manual review. Simply passing a validation report does not mean your ADaM datasets are compliant.

REFERENCES

CDISC Analysis Data Model Team, 2009, "Analysis Data Model (ADaM) v2.1", available on CDISC website at <<http://www.cdisc.org/standards>>

CDISC Analysis Data Model Team, 2009, "Analysis Data Model (ADaM) Implementation Guide v1.0", available on CDISC website at <<http://www.cdisc.org/standards>>

CDISC ADaM Compliance Sub-Team, 2015, "CDISC ADaM Validation Checks v1.3", available on CDISC website at <<http://www.cdisc.org/standards>>

CONTACT INFORMATION

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

Trevor Mankus
PRA Health Sciences
MankusTrevor@prahs.com

Kent Letourneau
PRA Health Sciences
LetourneauKent@prahs.com