PhUSE EU Connect 2018

Paper SI04

Optimization of the traceability when applying an ADaM Parallel Conversion Method

DEBRUS Roxane, Business & Decision Life Sciences, Brussels, Belgium

ABSTRACT

One of the methods to create CDISC ADaM datasets is the "Parallel Conversion", which uses the Legacy Analysis Datasets and the sponsor's specific study specifications as a source to generate the ADaM Datasets. The validation of analysis data is done by reproduction of key TFLs based on the ADaM datasets. The main challenge in this type of conversion is generating the traceability in Define-XML between SDTM and ADaM. In the past years, Business & Decision Life Sciences (BDLS) generated 2 SAS macros to improve the validation procedure. The first describes the characteristics (name, label, type, length) and the actual content of each variable from the SDTM and ADaM libraries. The second reads the metadata in the ADaM Define-XML and verifies its correctness by comparing it with the variables and their content in SDTM and ADaM. In this presentation, we will share a practical example to demonstrate how we automated the validation of the traceability between SDTM and ADaM by using the developed macros. We will also present the challenges encountered, the improvements done and also the future options we plan to develop.

INTRODUCTION

Following the Clinical Data Interchange Standards Consortium (CDISC), the Analysis Data Model (ADaM) defines dataset and metadata standards that support efficient generation, replication and review of clinical trial statistical analyses, providing traceability among analysis results, analysis data, and data represented in the Study Data Tabulation Model (SDTM). The purpose of ADaM is to provide a framework that enables analysis of the data, while at the same time allow reviewers and other recipients of the data to have a clear understanding of the data's lineage, all the way from collection to analysis to results. ADaM is one of the required standards for data submission to the Food and Drug Administration (FDA) in the U.S. and Pharmaceuticals and Medical Devices Agency (PMDA) in Japan.

ADAM CONVERSION

As illustrated on Figure 1, there are multiple ways to generate CDISC ADaM datasets, however the most common process is to generate it directly by using the SDTM datasets as a source, called "Linear Conversion". The second method to create CDISC ADaM datasets is the "Parallel Conversion", where both SDTM and ADaM are created independently, utilising the Legacy Analysis Datasets and the sponsor's specific study specifications (mapping) as a source to generate the ADaM Datasets.

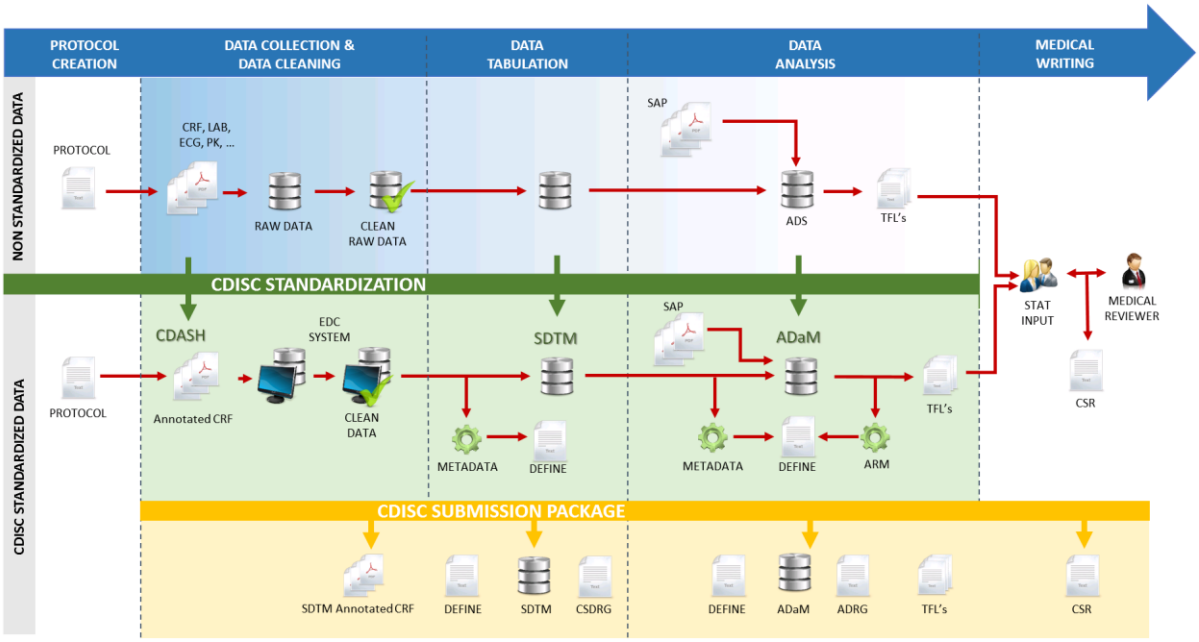


Figure 1 – Global CDISC Standardization Flow

An important component of a regulatory review is an understanding of the provenance of the data (i.e. traceability of the sponsor's results back to the CRF data). Traceability permits an understanding of the relationships between the analysis results (tables, listing and figures in the study report), analysis datasets, tabulation datasets and source data.

Traceability enables the reviewer to accomplish the following:

- Understand the construction of analysis datasets
- Determine the observations and algorithm(s) used to derive variables
- Understand how the confidence interval or the p-value was calculated in a particular analysis
- Relate counts from tables, listings and figures in a study report to the underlying data

Based on reviewer experience, establishing traceability is one of the most problematic issues associated with any data conversion [4]. If the reviewer is unable to trace study data from the data collection of subjects participating in a study to the analysis of the overall study data, then the regulatory review of a submission may be compromised. The FDA does not recommend a particular approach to legacy clinical study data conversion, but rather explains the issues that should be addressed so that the converted data is traceable and adequate to support review.

PARALLEL ADAM CONVERSION METHOD

Figure 2 below illustrates the parallel ADaM Conversion method with an independent SDTM and ADaM conversion scenario. The Study Data Technical Conformance Guide (v4.1, March 2018) identified the following issues that can occur when this approach is implemented:

1. Limited ability to determine location of collected CRF variables in the converted SDTM data unless the legacy aCRF is re-annotated.
2. Limited traceable path from SDTM to the legacy analysis data.
3. Limited ability to replicate/confirm legacy analysis datasets (i.e. analysis variable imputation or derived variables) using SDTM datasets.
4. Limited ability to confirm derivation of intermediate analysis datasets or custom domains.
5. Limited traceable path from SDTM to the ADaM datasets.
6. Limited ability to replicate ADaM datasets (i.e. analysis variable imputation or derived variables) using SDTM datasets.
7. Limited traceable path from ADaM to the Tables, Figures and the Clinical Study Report (CSR).
8. Difficulty in understanding the source or derivation methods for imputed or derived variables in integrated/pooled data, supplemental qualifiers and related records.

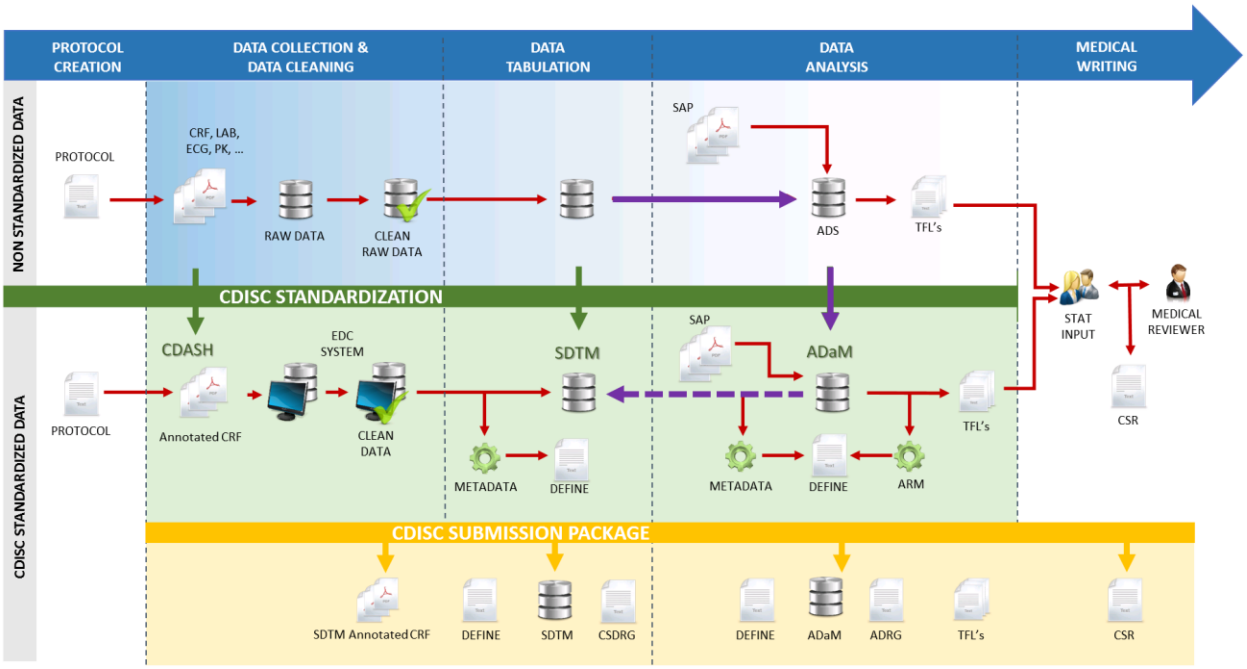


Figure 2 – Parallel CDISC ADaM Standardization Flow

In order to reduce the risk of a traceability issue while applying the ADaM parallel conversion method, the statistical team at BDLS has developed 2 macros over the last couple of years to use in addition to the standard operational procedures to ensure that some major principles are correctly applied during the CDISC conversion. These are automated checks that will scan the datasets and validate not only the structure but also the metadata and the content of the ADaM datasets generated from the Legacy Analysis Datasets, compared to the SDTM datasets.

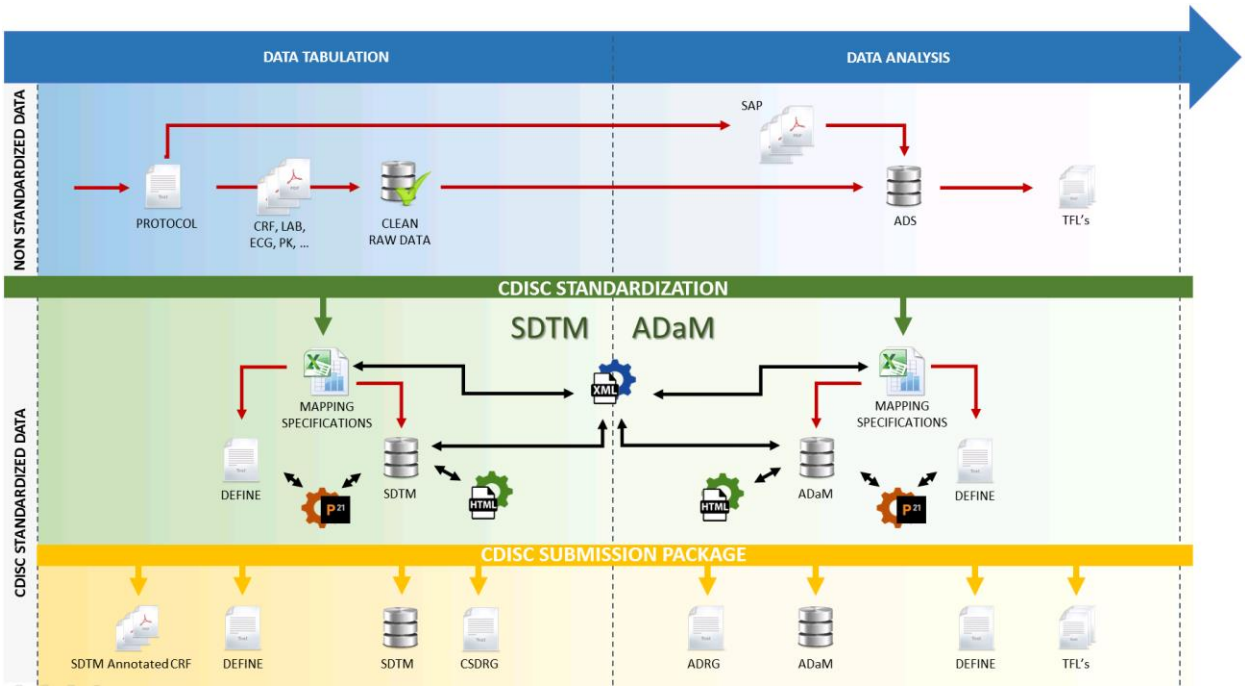


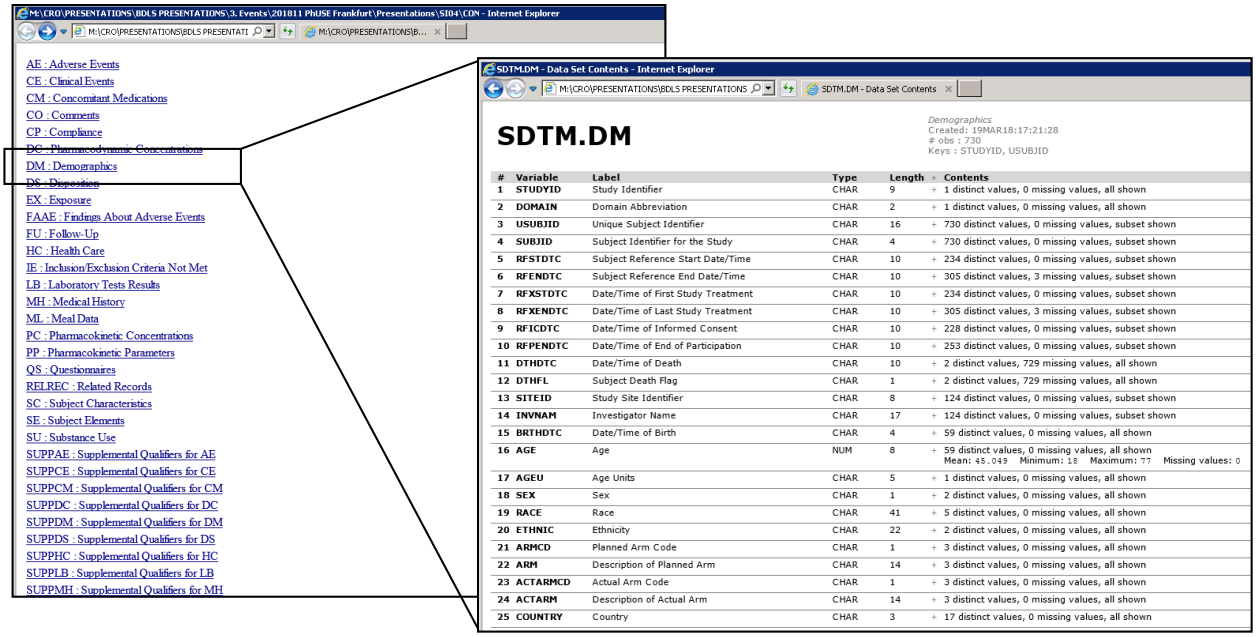
Figure 3 – Automated Validation Steps

As illustrated in the Figure 3, the content macro is run directly on the datasets and produces a set of HTML reports per library. It can be applied separately for SDTM and ADaM. However, the SDTM-ADaM compare macro needs to be applied simultaneously on both SDTM and ADaM, mapping and datasets and will check 8 major principles:

- A variable present in ADaM is indicated as a copy of a variable present in SDTM but the variables do not have identical metadata attributes
- A variable present in ADaM is indicated as a copy of a SDTM variable but the SDTM variable does not exist in SDTM variable metadata
- A variable present in ADaM is indicated as a derived variable but the same SDTM variable name exist in SDTM variable metadata
- SDTM.EX is present and neither ADSL.TRTSDT or ADSL.TRTSDTM are present
- The variable referenced in the computational algorithm of the ADaM variable does not exist in SDTM or ADaM variable metadata
- A variable present in ADaM is indicated as a copy of a variable present in SDTM but the values are not identical
- The values of USUBJID are not present in SDTM.DM
- SRCDOM has a value that is not a SDTM dataset, ADaM dataset name or null

THE CONTENT MACRO

This macro reads any type of libraries, whether it is CDISC compliant data (SDTM, ADaM, SEND, etc.) or not (Raw data, Analysis Dataset, etc.) and produces a series of HTML files containing first a global description of all the datasets stored in the specified library and the main characteristics and actual content of each variable.



For each variable, the report gives the main programming characteristics (i.e. variable name, label, type and length) and main content characteristics (i.e. number of distinct values and number of missing values for the character variables, with additional mean, minimum and maximum values for the numeric variables).

In case a variable is completely empty and all values are missing, the content description is coloured in red in order to draw the reviewer's attention. Depending on the variable, the decision to remove it or not can be taken.

25	AESCONG	Congenital Anomaly or Birth Defect	CHAR	1	– All values are missing

By clicking on the "+", more detailed information related to the content can be displayed:

- For **categorical variables**, the number of distinct values and the complete list of values contained in the dataset is displayed with its respective frequency.

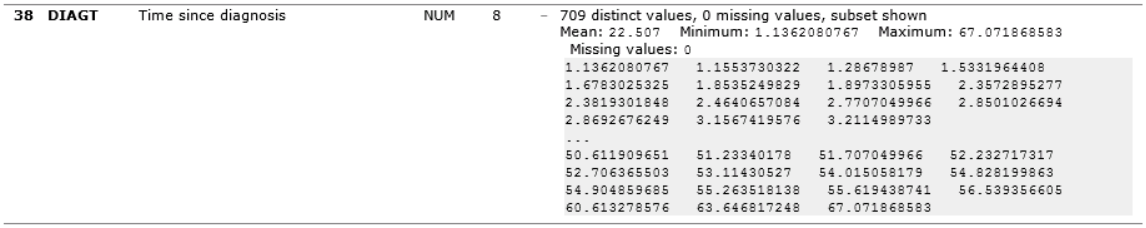
18	SEX	Sex	CHAR	1	– 2 distinct values, 0 missing values, all shown "F" (389x) "M" (341x)
19	RACE	Race	CHAR	41	– 5 distinct values, 0 missing values, all shown "ASIAN" (23x) "BLACK OR AFRICAN AMERICAN" (17x) "MULTIPLE" (2x) "NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER" (1x) "WHITE" (687x)

In case there are more than 60 distinct values, only a subset will be displayed. The selection is done in order to represent the value distribution as much as possible. For example, for DM.RFSTDTC, there are 234 distinct values while the subset display 78 distinct values, and it will be the 1st, the 4th, the 7th, 10th, 13th, etc. after sorting all distinct values by alphabetic order. In this case, the frequency isn't shown.

5	RFSTDTC	Subject Reference Start Date/Time	CHAR	10	– 234 distinct values, 0 missing values, subset shown "2015-10-06" "2015-10-14" "2015-10-19" "2015-10-22" "2015-10-27" "2015-11-03" "2015-11-09" "2015-11-12" "2015-11-17" "2015-11-23" "2015-11-30" "2015-12-03" "2015-12-08" "2015-12-11" "2015-12-16" "2015-12-21" "2015-12-25" "2015-12-30" "2016-01-06" "2016-01-11" "2016-01-14" "2016-01-19" "2016-01-22" "2016-01-27" "2016-02-01" "2016-02-04" "2016-02-09" "2016-02-12" "2016-02-17" "2016-02-22" "2016-02-25" "2016-03-01" "2016-03-04" "2016-03-09" "2016-03-14" "2016-03-17" "2016-03-22" "2016-03-28" "2016-04-01" "2016-04-06" "2016-04-11" "2016-04-14" "2016-04-19" "2016-04-25" "2016-04-28" "2016-05-03" "2016-05-09" "2016-05-12" "2016-05-18" "2016-05-23" "2016-05-26" "2016-06-02" "2016-06-07" "2016-06-13" "2016-06-16" "2016-06-21" "2016-06-27" "2016-07-01" "2016-07-06" "2016-07-11" "2016-07-14" "2016-07-19" "2016-07-22" "2016-07-27" "2016-08-01" "2016-08-04" "2016-08-09" "2016-08-12" "2016-08-17" "2016-08-22" "2016-08-25" "2016-08-30" "2016-09-02" "2016-09-07" "2016-09-12" "2016-09-19" "2016-09-22" "2016-09-27"

- For **discrete variables**, each value contained in the dataset is presented with its frequency. Furthermore, the mean, minimum and maximum value and number of missing values are given for numerical fields.

- For **continuous variables**, the number of distinct values is presented, with the mean, minimum and maximum values and number of missing values. Furthermore, the list of values contained in the dataset will be displayed. In case there are more than 60 distinct values, only a subset is displayed. The selection is made in order to illustrate the complete distribution and shows the top 15 and bottom 15 values after sorting all distinct values by ascending order. In this case, the frequency isn't shown.



HOW TO USE THE MACRO

The standard way to call the macro is to define the name of the selected library (or its path) as single parameter (if the library hasn't been assigned yet). As a result, the macro will generate the standard output defined, containing the global description file and one description file per dataset contained in the library selected:

```
%LIB_QC_contents_html(SDTM);

%LIB_QC_contents_html(W:\PROJECT-X\STUDY-0001\SDTM);
```

Options:

Select one or several dataset(s) to be run. This can be useful in case the output needs to be used during the pre-production phase and refreshed after applying some small updates.

```
%LIB_QC_contents_html(SDTM, select=DM AE);
```

Activate the format on the data to check the correctness of the format and not the underlying value. By default this parameter is set to "No".

```
%LIB_QC_contents_html(SDTM, APPLY_FORMATS=Yes);
```

Activate strip_fmts_except_DT = Y so that any variable having a format starting with "TOD", "DATE", "TIME", "BEST", "B8601D", "B8601T", "E8601D", "E8601T", "YYMMDD" are preserved. This maintains the readability of the date format while excluding any other not required formats.

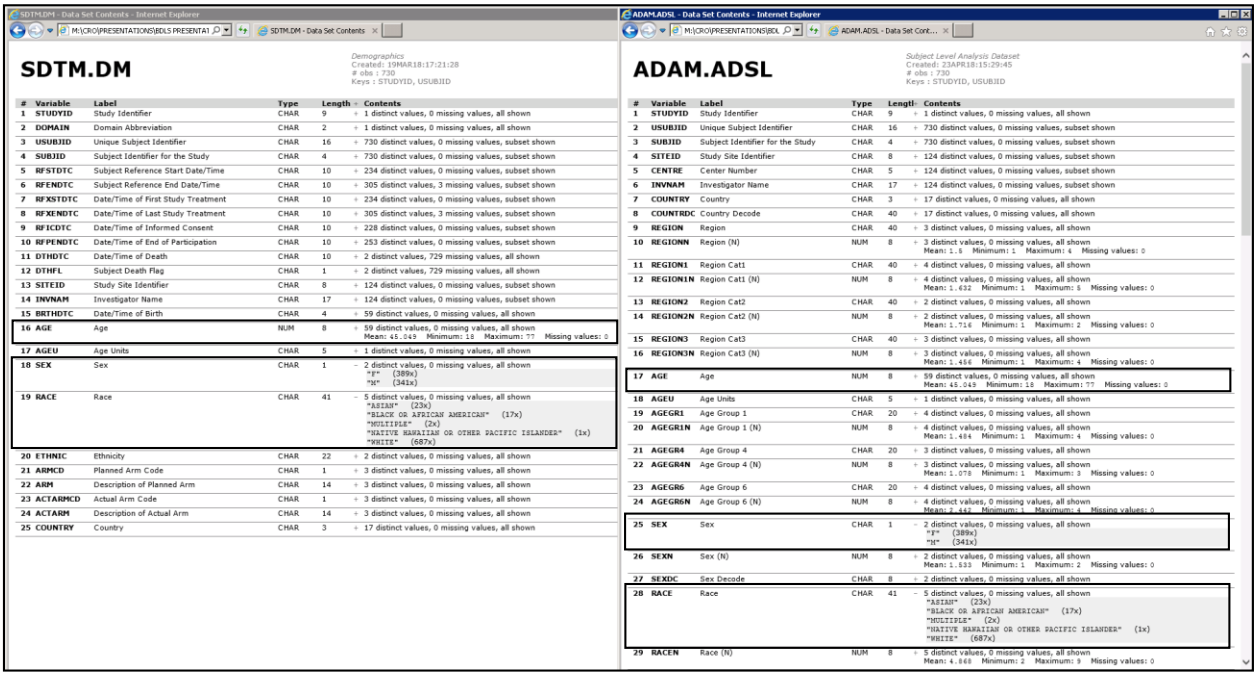
```
%LIB_QC_contents_html(SDTM, select=AE, strip_fmts_except_DT=Y);
```

The options can be combined and called simultaneously to tailor the output depending on the needs:

```
%LIB_QC_contents_html(ADAM, select=ADAE ADSL, APPLY_FORMATS=Yes, FMTLIB=
W:\PROJECT-X\STUDY-0001\ADAM\Reference Data);
```

PRACTICAL EXAMPLE 1

When the macro is run on different libraries, it allows some easy consistency checks to be done. For example ADSL.AGE, ADSL.RACE and ADSL.SEX should be exactly equivalent in values and characteristics to DM.AGE, DM.RACE and DM.SEX respectively. By putting both outputs next to each other, the reviewer can easily verify that this requirement has been fulfilled.



SDTM.DM

Demographics
Created: 19MAR18:17:21:28
obs : 730
Keys : STUDYID, USUBJID

#	Variable	Label	Type	Length	Contents
1	STUDYID	Study Identifier	CHAR	9	+ 1 distinct values, 0 missing values, all shown
2	DOMAIN	Domain Abbreviation	CHAR	2	+ 1 distinct values, 0 missing values, all shown
3	USUBJID	Unique Subject Identifier	CHAR	16	+ 730 distinct values, 0 missing values, subset shown
4	SUBJID	Subject Identifier for the Study	CHAR	4	+ 730 distinct values, 0 missing values, subset shown
5	RFSTDTC	Subject Reference Start Date/Time	CHAR	10	+ 234 distinct values, 0 missing values, subset shown
6	RFENDTC	Subject Reference End Date/Time	CHAR	10	+ 305 distinct values, 3 missing values, subset shown
7	RFXSTDTC	Date/Time of First Study Treatment	CHAR	10	+ 234 distinct values, 0 missing values, subset shown
8	RFXENDTC	Date/Time of Last Study Treatment	CHAR	10	+ 305 distinct values, 3 missing values, subset shown
9	RFICDTC	Date/Time of Informed Consent	CHAR	10	+ 228 distinct values, 0 missing values, subset shown
10	RFPENDTC	Date/Time of End of Participation	CHAR	10	+ 253 distinct values, 0 missing values, subset shown
11	DTHDTC	Date/Time of Death	CHAR	10	+ 2 distinct values, 729 missing values, all shown
12	DTHFL	Subject Death Flag	CHAR	1	+ 2 distinct values, 729 missing values, all shown
13	SITEID	Study Site Identifier	CHAR	8	+ 124 distinct values, 0 missing values, subset shown
14	INVNAM	Investigator Name	CHAR	17	+ 124 distinct values, 0 missing values, subset shown
15	BRTHDTC	Date/Time of Birth	CHAR	4	+ 59 distinct values, 0 missing values, all shown
16	AGE	Age	NUM	8	+ 59 distinct values, 0 missing values, all shown Mean: 45.049 Minimum: 18 Maximum: 77 Missing values: 0
17	AGEU	Age Units	CHAR	5	+ 1 distinct values, 0 missing values, all shown
18	SEX	Sex	CHAR	1	- 2 distinct values, 0 missing values, all shown "F" (389x) "M" (341x)
19	RACE	Race	CHAR	41	- 5 distinct values, 0 missing values, all shown "ASIAN" (23x) "BLACK OR AFRICAN AMERICAN" (17x) "MULTIPLE" (2x) "NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER" (1x) "WHITE" (687x)
20	ETHNIC	Ethnicity	CHAR	22	+ 2 distinct values, 0 missing values, all shown
21	ARMCD	Planned Arm Code	CHAR	1	+ 3 distinct values, 0 missing values, all shown
22	ARM	Description of Planned Arm	CHAR	14	+ 3 distinct values, 0 missing values, all shown
23	ACTARMCD	Actual Arm Code	CHAR	1	+ 3 distinct values, 0 missing values, all shown
24	ACTARM	Description of Actual Arm	CHAR	14	+ 3 distinct values, 0 missing values, all shown
25	COUNTRY	Country	CHAR	3	+ 17 distinct values, 0 missing values, all shown

ADAM.ADSL

Subject Level Analysis Dataset
Created: 23APR18:15:29:45
obs : 730
Keys : STUDYID, USUBJID

#	Variable	Label	Type	Length	Contents
1	STUDYID	Study Identifier	CHAR	9	+ 1 distinct values, 0 missing values, all shown
2	USUBJID	Unique Subject Identifier	CHAR	16	+ 730 distinct values, 0 missing values, subset shown
3	SUBJID	Subject Identifier for the Study	CHAR	4	+ 730 distinct values, 0 missing values, subset shown
4	SITEID	Study Site Identifier	CHAR	8	+ 124 distinct values, 0 missing values, subset shown
5	CENTRE	Center Number	CHAR	5	+ 124 distinct values, 0 missing values, subset shown
6	INVNAM	Investigator Name	CHAR	17	+ 124 distinct values, 0 missing values, subset shown
7	COUNTRY	Country	CHAR	3	+ 17 distinct values, 0 missing values, all shown
8	COUNTRDC	Country Decode	CHAR	40	+ 17 distinct values, 0 missing values, all shown
9	REGION	Region	CHAR	40	+ 3 distinct values, 0 missing values, all shown
10	REGIONN	Region (N)	NUM	8	+ 3 distinct values, 0 missing values, all shown Mean: 1.5 Minimum: 1 Maximum: 4 Missing values: 0
11	REGION1	Region Cat1	CHAR	40	+ 4 distinct values, 0 missing values, all shown
12	REGION1N	Region Cat1 (N)	NUM	8	+ 4 distinct values, 0 missing values, all shown Mean: 1.632 Minimum: 1 Maximum: 5 Missing values: 0
13	REGION2	Region Cat2	CHAR	40	+ 2 distinct values, 0 missing values, all shown
14	REGION2N	Region Cat2 (N)	NUM	8	+ 2 distinct values, 0 missing values, all shown Mean: 1.714 Minimum: 1 Maximum: 2 Missing values: 0
15	REGION3	Region Cat3	CHAR	40	+ 3 distinct values, 0 missing values, all shown
16	REGION3N	Region Cat3 (N)	NUM	8	+ 3 distinct values, 0 missing values, all shown Mean: 1.464 Minimum: 1 Maximum: 4 Missing values: 0
17	AGE	Age	NUM	8	+ 59 distinct values, 0 missing values, all shown Mean: 45.049 Minimum: 18 Maximum: 77 Missing values: 0
18	AGEU	Age Units	CHAR	5	+ 1 distinct values, 0 missing values, all shown
19	AGEGR1	Age Group 1	CHAR	20	+ 4 distinct values, 0 missing values, all shown
20	AGEGR1N	Age Group 1 (N)	NUM	8	+ 4 distinct values, 0 missing values, all shown Mean: 1.484 Minimum: 1 Maximum: 4 Missing values: 0
21	AGEGR4	Age Group 4	CHAR	20	+ 3 distinct values, 0 missing values, all shown
22	AGEGR4N	Age Group 4 (N)	NUM	8	+ 3 distinct values, 0 missing values, all shown Mean: 1.078 Minimum: 1 Maximum: 3 Missing values: 0
23	AGEGR6	Age Group 6	CHAR	20	+ 4 distinct values, 0 missing values, all shown
24	AGEGR6N	Age Group 6 (N)	NUM	8	+ 4 distinct values, 0 missing values, all shown Mean: 2.442 Minimum: 1 Maximum: 4 Missing values: 0
25	SEX	Sex	CHAR	1	- 2 distinct values, 0 missing values, all shown "F" (389x) "M" (341x)
26	SEXN	Sex (N)	NUM	8	+ 2 distinct values, 0 missing values, all shown Mean: 1.533 Minimum: 1 Maximum: 2 Missing values: 0
27	SEXDC	Sex Decode	CHAR	8	+ 2 distinct values, 0 missing values, all shown
28	RACE	Race	CHAR	41	- 5 distinct values, 0 missing values, all shown "ASIAN" (23x) "BLACK OR AFRICAN AMERICAN" (17x) "MULTIPLE" (2x) "NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER" (1x) "WHITE" (687x)
29	RACEN	Race (N)	NUM	8	+ 5 distinct values, 0 missing values, all shown Mean: 4.848 Minimum: 2 Maximum: 9 Missing values: 0

PRACTICAL EXAMPLE 2

According to the ADaM IG, population flags shouldn't have a missing value and should be always equal to "Y" or "N".

In this example an issue has been identified where the safety population flag has 23 occurrences of missing values. This can easily be spotted during the quality process and corrected before further validation steps.

94	SAFFL	Safety Population Flag	CHAR	1	- 1 distinct values, 23 missing values, all shown **MISSING** (23x) "Y" (700x) "N" (7x)

In this example an issue has been identified where the safety population flag has an unexpected value and/or an unexpected characteristic (e.g. length of 2). It gives a high level overview of metadata and not only a variable level values review.

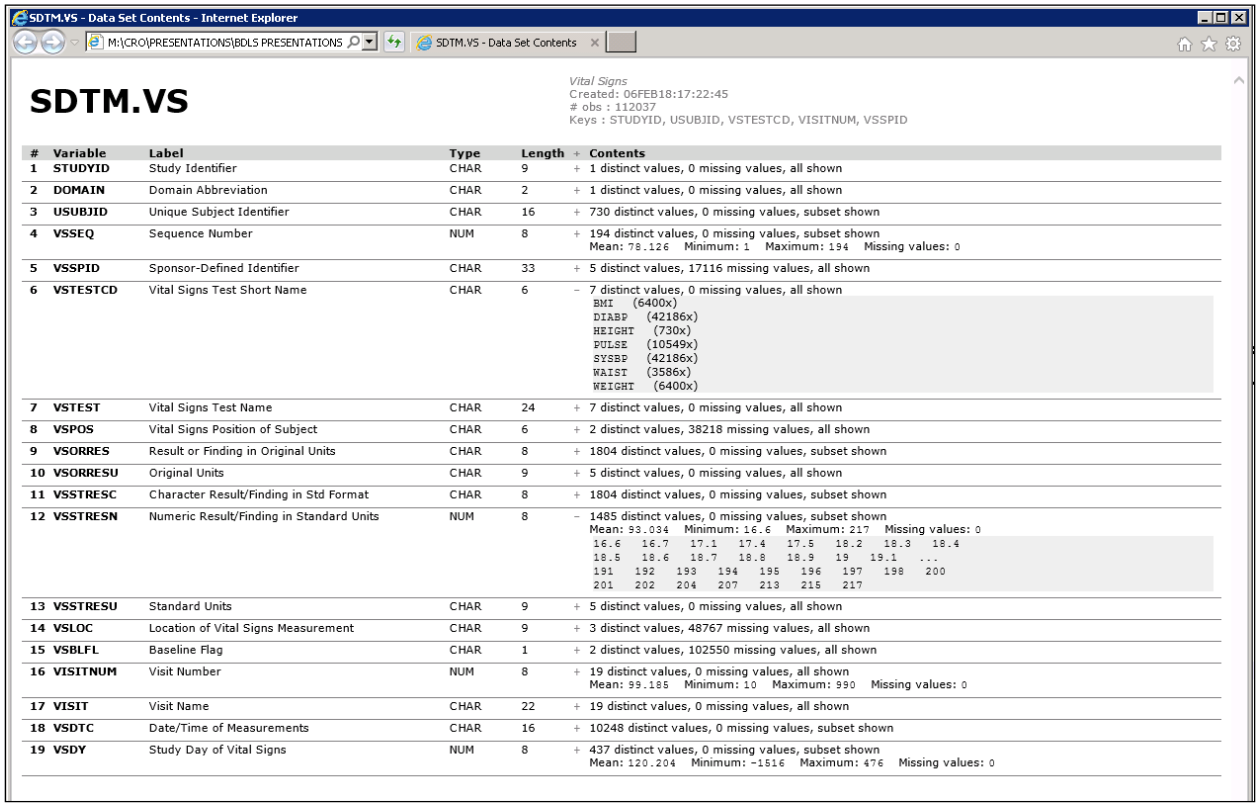
94	SAFFL	Safety Population Flag	CHAR	2	- 3 distinct values, 0 missing values, all shown "Y" (700x) "N" (7x) "NA" (23x)

ADVANTAGES

- ✓ Easy to produce: One macro to be run on any kind of data; creating one HTML output per dataset.
- ✓ Easy to integrate: Runs on different Statistical Analysis System (SAS) version: Base, Studio or DI Studio.
- ✓ Easy to share: Small output size - Size from 50 to 500Kb per file, 1 library contains up to 40-50 datasets (maximum size 10 MB).
- ✓ Easy to open: Opens with any web browser (Internet Explorer, Firefox, Chrome, Opera, etc.) - Does not require any installation or additional software.
- ✓ Easy to read: Gives simple, useful information - Easily understandable whatever the profile of the reader.

CHALLENGES AND IMPROVEMENT

No subset 'by variable' can currently be made in the output or in the macro on a particular variable value, which can make the review a little difficult, especially for Findings/BDS structured datasets. For example, in the SDTM.VS dataset, most of the time, there is more than one test performed (i.e. different value of VS.VSTEST / VS.VSTESTCD) and the result/finding in standard unit (i.e. VS.VSSTRESN / VS.VSSTRESC) value description can't be displayed by test name. The addition of a 'by variable' subset option could improve the level of detail in the output and increase the usability of these reports.



Another limitation is related to free text variables such as Deviation Term (DV.DVTERM), Adverse Event (AE.AETERM), Result or Findings in Original Unit (--ORRES), Concomitant Medication Term (CM.CMTERM) or Concomitant Medication Indication (CM.CMINDC), etc. where there is a high level of variability between the subjects. As previously explained, for character variables having more than 60 distinct values, the number of displayed value will be limited to a subset and there is no selection possible on which recorded value will be shown. Therefore it is not possible to check if truncations has been applied on the data. A suggestion for improvement would be to add the value of the longest record in the display for theses free-text variables.

Another option could be to allow the selection to be done not by alphabetical order but on the decreasing frequency of each distinct value or to sort the values by length before sorting alphabetically. Finally, in case there are too many distinct values to display, the selection could always keep the top 5 and bottom 5 values (both by length and by frequency), then apply a selection on the remaining values 'in the middle'.

The systematic review of the outputs for all variables is a boring and time-consuming task. Maybe some types of variables are more important to review than others? E.g. the numerical variables that have plausible ranges of values, to check for outliers? Would it be useful that the macro checks for and reports outliers (displays them in another colour), e.g. values distant from the mean by more than 2 or 3 times the standard deviation?

Another useful option would be to have the reports also in excel format, so that users could add comments and annotations to some variables and values (those most suspect of having issues) in order to keep track of their review more easily.

THE SDTM-ADAM COMPARE MACRO

This second macro reads the metadata in the ADaM specification contained in the Define-XML and verifies its correctness by comparing it with the variables and their content in SDTM and ADaM. The output produced by the macro is one unique XML workbook file readable with Excel, containing 6 different tabs: Summary, SDTM Overview, ADaM Overview, Exception Metadata, Exception Data, and Log Overview.

- **Summary Tab**

Project: PROJECT-X
Study: STUDY-0001
Summary: Overall number of findings
Generated: 10SEP2018

Check ID	Error Message	Total Number of Findings
101	A variable present in ADaM is indicated as a copy of a variable present in SDTM but the variables do not have identical metadata attributes	81
102	A variable present in ADaM is indicated as a copy of a SDTM variable but the SDTM variable does not exist in SDTM variable metadata	0
103	A variable present in ADaM is indicated as a derived variable but the same SDTM variable name exist in SDTM variable metadata	53
104	SDTM.EX is present and neither ADSL.TRTSDT or ADSL.TRTSDTM are present	0
105	The variable referenced in the computational algorithm of the ADaM variable does not exist in SDTM or ADaM variable metadata	410
201	A variable present in ADaM is indicated as a copy of a variable present in SDTM but the values are not identical	7319
202	The values of USUBJID are not present in SDTM.DM	0
203	SRCDOM has a value that is not a SDTM dataset name or null	0

In this first tab all the checks that have been run on the datasets and the total number of findings observed are listed. By default eight checks are run: five on the metadata (101 to 105) and three on the data itself (201 to 203). The explanation of each check with a practical example to illustrate is given in the Exception Tabs.

- **SDTM Overview**

Project: PROJECT-X
Study: STUDY-0001
SDTM Overview
Generated: 10SEP2018

Dataset Name	Dataset Label	Total Number of Rows	Date Created	Check #0101	Check #0102	Check #0103	Check #0104	Check #0105	Check #0201	Check #0202	Check #0203
AE	Adverse Events	22156	29JAN18:11:16:17	**21**	0	0	0	**194**	0	0	0
CE	Clinical Events	910785	22JUN18:11:12:36	**12**	0	0	0	**8**	0	0	0
CM	Concomitant Medications	808442	15MAY18:15:35:22	0	0	0	0	**2**	0	0	0
CO	Comments	3574	05JUN18:10:03:40	0	0	0	0	**139**	0	0	0
CP	Compliance	7763	29JAN18:11:46:36	**3**	0	0	0	**2**	0	0	0
DC	Pharmacodynamic Concentrations	17120	14MAR18:13:39:27	**2**	0	0	0	**4**	0	0	0
DM	Demographics	730	19MAR18:17:21:28	**15**	0	0	0	**1**	0	0	0
DS	Disposition	3666	29JAN18:11:34:20	0	0	0	0	**27**	0	0	0
EX	Exposure	6120	13FEB18:13:36:59	**3**	0	0	0	**2**	0	0	0
FAAE	Findings About Adverse Events	92589	17MAY18:09:33:53	0	0	0	0	0	0	0	0
FU	Follow-Up	280	29JAN18:11:47:34	0	0	0	0	**2**	0	0	0
HC	Health Care	7999	29JAN18:11:48:59	0	0	0	0	**1**	0	0	0
IE	Inclusion/Exclusion Criteria Not Met	9	29JAN18:11:52:45	0	0	0	0	**2**	0	0	0
LB	Laboratory Tests Results	12203144	03JUL18:13:07:33	**3**	0	0	0	**6**	0	0	0
MH	Medical History	15487	29JAN18:09:47:39	0	0	0	0	0	0	0	0
ML	Meal Data	9119	29JAN18:12:04:01	0	0	0	0	0	0	0	0

This second tab gives a more detailed overview of the results on a SDTM dataset level. The two first columns are used to identify the SDTM datasets that have been checked, while the third and the forth column are used to identify on which version of the SDTM datasets the checks have been run (number of records and creation date and time).

The remaining columns in this tab contain the number of findings per dataset for each check. When there is no finding for a check, the value is zero. When there is at least one finding, the number of findings is displayed (in bold and underlined). To ease the review of this report, there are hyperlinks on the title row and when clicking on a check number, it redirects the reader to the explanation of the check.

- **ADaM Overview**

This third tab gives a detailed overview of the results on ADaM dataset level. Identically to the previous tab, the two first columns are used to identify the ADaM datasets that have been checked, while the third and the forth columns are used to clearly identify on which version of the ADaM datasets the checks have been performed (number of records and creation date and time). The other part of this tab defines the number of findings by dataset and type of check.

As the checks are comparing SDTM and ADaM against each other, most of the findings will be populated in both SDTM and ADaM overview. However, as some findings may be related to a lack of connection between both dataset libraries some findings might be populated in only one of the overview.

Project: PROJECT-X
Study: STUDY-0001
ADaM Overview
Generated: 10SEP2018

Dataset Name	Dataset Label	Total Number of Rows	Date Created	Check #0101	Check #0102	Check #0103	Check #0104	Check #0105	Check #0201	Check #0202	Check #0203
ADAE	Adverse Events Analysis Dataset	22156	23JUL18:14:51:45	0	0	14	0	0	0	0	0
ADAESI	Other Adverse Events of Spec Interest	44051	23JUL18:15:11:58	0	0	0	0	11	0	0	0
ADCE	Clinical Events Analysis Dataset	39	23JUL18:15:07:51	0	0	0	0	1	0	0	0
ADCM	Concomitant Medications Analysis Dataset	14480	23JUL18:15:07:35	0	0	22	0	0	0	0	0
ADCP	Compliance Analysis Dataset	62180	23JUL18:15:08:13	0	0	0	0	0	0	0	0
ADDISP	Consent, Rando., Treat., Discont. Data	1460	23JUL18:15:20:35	0	0	0	0	3	0	0	0
ADDV	Protocol Violation Analysis Dataset	299	23JUL18:15:42:12	0	0	0	0	0	0	0	0
ADEX	Exposure Analysis Data	1474	23JUL18:15:19:35	0	0	0	0	0	0	0	0
ADHC	Health care ressource utilisation Data	8630	23JUL18:14:43:28	0	0	28	0	0	0	0	0
ADLB	Laboratory Test Analysis Data	453438	23JUL18:15:15:59	0	0	2	0	0	0	0	0
ADMH	Medical History Analysis Dataset	3438	23JUL18:15:07:15	0	0	23	0	0	0	0	0
ADSL	Subject-Level Analysis Dataset	730	23JUL18:14:06:32	0	0	0	0	2	0	0	0
ADTTE	Time to Event Analysis Dataset	45260	23JUL18:15:11:16	0	0	0	0	0	0	0	0
ADVS	Vital Signs Analysis Dataset	415563	23JUL18:14:48:39	0	0	0	0	1	0	0	0

- **Exception Metadata**

This tab gives very detailed information about each finding. It gives the reviewer all the information needed to identify the issue exactly, i.e. the SDTM and/or ADaM dataset(s) considered, the variable name, the programming characteristics (type, length, codelist), and the error message. Each error has a Row ID assigned to ease the communication between the team members.

Check #101: *A variable present in ADaM is indicated as a copy of a variable present in SDTM but the variables do not have identical metadata attributes*

Following the ADaM IG, "*Any variable in an ADaM dataset whose name is the same as an SDTM variable must be a copy of the SDTM variable, and its label, meaning, and values must not be modified. ADaM adheres to a principle of harmonization known as "same name, same meaning, same values.*"

Project: PROJECT-X
Study: STUDY-0001
Exception Metadata
Generated: 10SEP2018

Check ID	Study ID	Dataset Comparison	Variable Name	Variable Label	Variable Type	Variable Length	Codelist	Error Message	Row ID
101	STUDY-0001	ADAE vs AE	AECAT	Category for Adverse Event	char	100	AECAT	A variable present in ADaM is indicated as a copy of a variable present in SDTM but the variables do not have identical metadata attributes	7
101	STUDY-0001	ADAE vs AE	AEDECOD	Dictionary-Derived Term	char	200		A variable present in ADaM is indicated as a copy of a variable present in SDTM but the variables do not have identical metadata attributes	8
101	STUDY-0001	ADAE vs AE	AEHLGTCD	High Level Group Term Code	char	8		A variable present in ADaM is indicated as a copy of a variable present in SDTM but the variables do not have identical metadata attributes	9

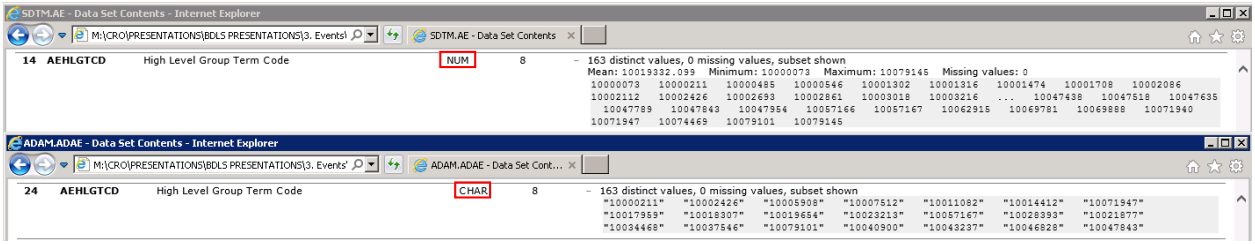
In order to understand the findings reported, both metadata should be put next to each other to compare their attributes. Therefore either the comparison is done by opening both mapping files and/or both define.xml files if they have already been produced.

Another option is to use a describe statement in a PROC SQL to extract directly the metadata from the dataset.

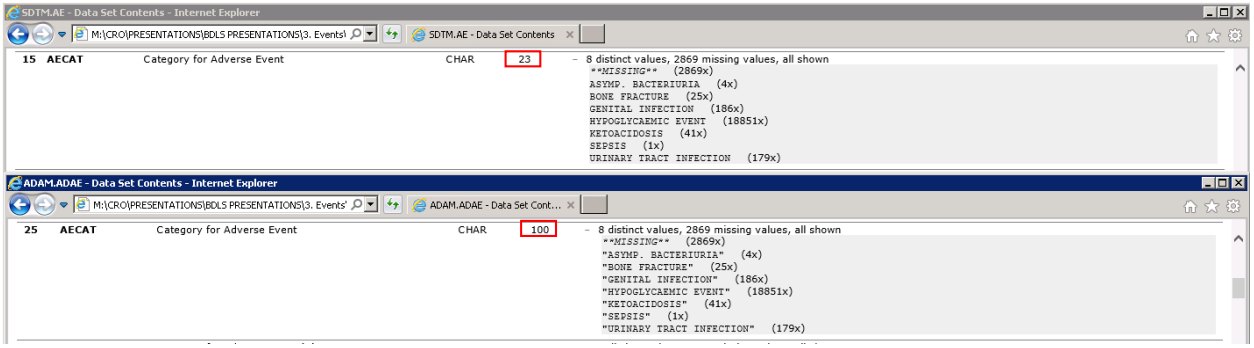
```
proc sql;
  describe table sdtm.ae;
  NOTE: SQL table SDTM.AE was created like:
  create table SDTM.AE( label='Adverse Events' bufsize=16384 )
  (
   …
   AEDECOD char(43) label='Dictionary-Derived Term',
   AEHLGTCD num label='High Level Group Term Code',
   AECAT char(23) label='Category for Adverse Event',
   …
  );
quit;
```

```
proc sql;
  describe table adam.adae;
  NOTE: SQL table ADAM.ADAE was created like:
  create table ADAM.ADAE( label='Adverse Events Analysis Dataset' bufsize=16384 )
  (
   …
   AEDECOD char(200) label='Dictionary-Derived Term',
   AEHLGTCD char (8) label='High Level Group Term Code',
   AECAT char(100) label='Category for Adverse Event',
   …
  );
quit;
```

At BDLS, we use the html outputs generated by the first macro presented in this paper (%LIB_QC_contents_html) for both datasets.



In this first example, the High Level Group Term Code (AEHLGTCD) is set as numeric variable in SDTM.AE while in the sponsor's Analysis Dataset (ADS), it has been recorded as a character variable and should be transformed as a numeric value during the ADaM conversion process. Because of this difference in the type of variable, the type of detail given in the content report is not the same.



This second example is a typical finding for this check as following SDTM IG, the length is set to a specific value, long enough to allow all the possible values to be entered, depending on the type of variable and at the end of the generation process, all variables should be trimmed and the length should be set to the length of the longest value. One of the possible actions to limit this kind of issue would be to trim the ADaM dataset as well, even if it is not mandatory. In theory, if the ADaM variable is an exact copy of the SDTM variable, it should be trimmed to the exact same length.

However, in the ADaM IG 1.1, a statement has been added compared to the ADaM IG v1.0, saying that variable length of SDTM variables can differ between SDTM and ADaM in order to optimize file size (e.g., trailing blanks may be removed). Therefore, each finding will need to be review individually to determine if an update is requested or not.

Check #102: *A variable present in ADaM is indicated as a copy of a SDTM variable but the SDTM variable does not exist in SDTM variable metadata*

These kind of findings can happen when a change in structure is done between two production runs in SDTM but this change has not been taken into account yet in the ADaM specification, e.g. to replace the Start Relative to Reference Period (CM.CMSTRF) by both the Start Relative to Reference Time Point (CM.CMSTRTPT) and the Start Reference Time Point (CM.CMSTTPT) to better identify the selected reference period. However, this finding is major and should be corrected. It shouldn't appear at all once the final ADaM is ready to be delivered.

Check #103: *A variable present in ADaM is indicated as a derived variable but the same SDTM variable name exist in SDTM variable metadata*

Following the ADaM IG 1.1, section 3.1, the variable name given to an ADaM variable should be chosen with care, to prevent unintended conflicts among other such names and standard numeric versions of possible SDTM variable names. Therefore all variables defined in ADaM are cross checked to ensure that this principle is applied.

Project: PROJECT-X
Study: STUDY-0001
Exception Metadata
Generated: 10SEP2018

Check ID	Study ID	Dataset Comparison	Variable Name	Variable Label	Variable Type	Variable Length	Codelist	Error Message	Row ID
103	STUDY-0001	ADCM vs CM	CMCAT	Category for Medication	char	100		A variable present in ADaM is indicated as a derived variable but the same SDTM variable name exist in SDTM variable metadata	38

During the ADaM parallel conversion process, some SDTM specific variables might be added into the ADaM datasets even if they don't have any analysis purpose but to improve the traceability and help the reviewer to clearly identify the origin of the data. However the traceability of these variables should ease the review and not mislead the reviewer.

Check #104: *SDTM.EX is present and neither ADSL.TRTSDT or ADSL.TRTSDTM are present*

Following the ADaM IG 1.1 section 3.2, the date of first exposure to treatment for a subject in a study (ADSL.TRTSDT and/or ADSL.TRTSDTM) are required if there is an investigational product. Note that ADSL.TRTSDT is not required to have the same value as the DM.RFXSTDTC. While both of these dates reflect the concept of first exposure, the ADaM date may be derived to support the analysis which may not necessarily be the very first date in the SDTM EX domain. This finding is major and should be corrected.

Check #105: *The variable referenced in the computational algorithm of the ADaM variable does not exist in SDTM or ADaM variable metadata*

Following the ADaM IG, the traceability is established by describing (via metadata) the algorithm used or steps taken to derive or populate an analysis variable from its immediate predecessor. Therefore, the consumer of an ADaM dataset must be able to identify clearly the data inputs and the algorithms used to create the derived information. The standard practice is to always identify a variable with a 2 level name, containing the dataset name and variable name, separated by a dot (e.g. "ADSL.USUBJID"). The macro scans all the user-defined algorithms and selects all the occurrences of text separated by a dot without any leading and/or trailing blank, and verifies that this selected variable name exists, either in SDTM or in ADaM.

Project: PROJECT-X
Study: STUDY-0001
Exception Metadata
Generated: 10SEP2018

Check ID	Study ID	Dataset Comparison	Variable Name	Variable Label	Variable Type	Variable Length	Codelist	Error Message	Row ID
105	STUDY-0001	ADSL.PPROCOM Contains ADSL.PPROFL	ADSL.PPROCOM	Per-Protocol Population Comment	char	200		The variable referenced in the computational algorithm of the ADaM variable does not exist in SDTM or ADaM variable metadata	360

In this example, the report highlighted that the algorithm of the variable "ADSL.PPROCOM" contains the variable 'ADSL.PPROFL" and that this variable doesn't exist. When we check the algorithm given in the define.xml file, we observe that there is indeed a typo in the name of this variable, and the value of ADSL.PPROFL should have been ADSL.PPRO**T**FL.

ADaM-IG 1.1

ADaM Reviewer's Guide
Statistical Analysis Plan

PPROCOM	Per-Protocol Population Comment	text	21		Derived: If patient is not included in the ADSL.PPROFL Equal to the first valid reason in the following list: "Open label treatment", "Not treated", "No baseline value", "Major Protocol Deviation".

Another example is given in this second extract of the report and is related to the algorithm of ADSL.ALCOHOLN.

Project: PROJECT-X
Study: STUDY-0001
Exception Metadata
Generated: 10SEP2018

Check ID	Study ID	Dataset Comparison	Variable Name	Variable Label	Variable Type	Variable Length	Codelist	Error Message	Row ID
105	STUDY-0001	ADSL.ALCOHOLN Contains SU.DOSTXT	ADSL.ALCOHOLN	Alcohol Status (N)	num	8	ALCOHOLN	The variable referenced in the computational algorithm of the ADaM variable does not exist in SDTM or ADaM variable metadata	358
105	STUDY-0001	ADSL.ALCOHOLN Contains SU.TRT	ADSL.ALCOHOLN	Alcohol Status (N)	num	8	ALCOHOLN	The variable referenced in the computational algorithm of the ADaM variable does not exist in SDTM or ADaM variable metadata	359

The check mentions that the algorithm contains both inexistent variables: SU.DOSTXT and SU.TRT. Indeed, there is a typo in both variable names, and the value of SU.DOSTXT and SU.TRT should have been SU.**SU**DOSTXT and SU.**SU**TRT respectively.

Despite the great sensitivity of this check to find the smallest spelling error, sometimes even invisible to the naked eye, this check has a great defect, which is its lack of specificity. In fact, any concatenation of two strings separated by a dot will be considered as a variable name. Therefore a lot of false positive findings will be generated as the dot is also used to make a hard stop at the end of a sentence, not always followed by a space. The macro will then look for some typical programming statement such as: “MISSING.ELSE” or “VALUE.IF”.

In case there is no occurrence at all for a specific check, a line is still populated with all values empty, expect the check ID and the error description message. In the Study ID column will be displayed “No exceptions” and the row ID value will be set to 0.

Project: PROJECT-X
Study: STUDY-0001
Exception Metadata
Generated: 10SEP2018

Check ID	Study ID	Dataset Comparison	Variable Name	Variable Label	Variable Type	Variable Length	Codelist	Error Message	Row ID
102	No exceptions							A variable is present in ADaM is indicated as a copy of a SDTM variable but the SDTM variable does not exist in SDTM variable metadata	0

- **Exception Data**

  Check #201: *A variable present in ADaM is indicated as a copy of a variable present in SDTM but the values are not identical*

Following the ADaM IG 1.1 section 3.1.1, any variable in an ADaM dataset whose name is the same as an SDTM variable must be a copy of the SDTM variable, and its label, meaning, and values must not be modified. ADaM adheres to a principle of harmonization known as “same name, same meaning, same values.”

Project: PROJECT-X
Study: STUDY-0001
Exception Data
Generated: 10SEP2018

Check ID	Study ID	Subject ID	Dataset and Variable Comparison	Variable Value	Error Message	Row ID
201	STUDY-0001	STUDY-0001-001234	ADAE.AESHOSP vs AE.AESHOSP	N	A variable present in ADaM is indicated as a copy of a variable present in SDTM but the values are not identical	11

In this first example, the check identified that for one of the records of a specific subject (i.e. ‘STUDY-0001-001234’), the value of ADAE.AESHOSP isn’t equal to the value of AE.AESHOSP. When we check the data by comparing this specific record in both SDTM.AE and ADAM.AE domain, we observe that there is indeed an issue as in ADAE the value is “N”, while in AE the values can only be ‘Y’ or missing. The derivation applied to this variable should be corrected.

Project: PROJECT-X
Study: STUDY-0001
Exception Data
Generated: 10SEP2018

Check ID	Study ID	Subject ID	Dataset and Variable Comparison	Variable Value	Error Message	Row ID
201	STUDY-0001	STUDY-0001-001234	ADAE.AEENDTC vs AE.AEENDTC	2017-08-31	A variable present in ADaM is indicated as a copy of a variable present in SDTM but the values are not identical	1

In this second example, the check identified that for one of the records of a specific subject (i.e. ‘STUDY-0001-001234’), the value of ADAE.AEENDTC isn’t equal to the value of AE.AEENDTC. When we check the data by comparing this specific record in both SDTM.AE and ADAM.AE domain, we observe that there is indeed an issue as in ADAE the reported End date of AE is missing, while End Date/Time of AE is equal to “2017-08-31” and the End Date of AE Imputation flag is missing. The derivation applied to this variable should be corrected.

VIEWTABLE: Sdtm.Ae (Adverse Events)

	DOMAIN	AESEQ	AESPID	AETERM	AEDECOD	AEBODSYS	AESTDTC	AEENDTC	AESTDY	AEENDY
847	AE	1	1.02	DEPRESSION	Depression	Psychiatric disorders	2017-02-07		182	

VIEWTABLE: Adam.Adae (Adverse Events Analysis Dataset)

	Reported Term for the Adverse Event	Dictionary-Derived Term	Body System or Organ Class	Reported Start Date of Adverse Event	Reported End date of Adverse Event	Start Date/Time of Adverse Event	Start Date of Adverse Event	Analysis Start Date Imputation Flag	End Date/Time of Adverse Event	End Date of Adverse Event	End Date of Adverse Event Imp. Flag	AE end date censored Flag
888	DEPRESSION	Depression	Psychiatric disorders	20170207		2017-02-07	07FEB2017		2017-08-31	31AUG2017		Y

Check #202: *The values of USUBJID are not present in SDTM.DM*

This issue is specific for the ADaM parallel conversion method because it is possible that the screen failure subjects were ignored in the SDTM datasets but were kept in the raw data and sponsors-define analysis datasets (ADS). This issue could not be met in case of a linear conversion because it would use the SDTM as only source. Within a given study, USUBJID is the key variable that links ADSL to other datasets (both SDTM and ADaM). Following the ADAM IG 1.1, ADSL.USUBJID is a required variable and should be exactly equal and equivalent to DM.USUBJID. This finding is major and should be corrected.

Check #203: *SRCDOM has a value that is not a SDTM dataset, an ADaM dataset name or null*

Following the ADaM IG 1.1, to refer to a record in a predecessor ADaM dataset, the value of SRCDOM should be set to the name of the predecessor dataset, SRCVAR set to the variable name and SRCSEQ set to the value of --SEQ in the predecessor dataset. This finding is major and should be corrected.

- **Log Overview**

This last tab is generated in order to verify that the macro ran completely and that the executions steps of the macro weren't stopped during the process because of an error, and that no warning or important note is printed in the log. If none are met, a record is generated with the value of 'No Errors, Warning in the Log' (see example 1). Otherwise, all NOTE, WARNING and ERROR are reported (see example 2) in order to be fixed.

Example 1

Project: PROJECT-X
Study: STUDY-0001
Log Overview
Generated: 10SEP2018

Log Message	Row ID
No Errors, Warnings in the Log	1

Example 2

Project: PROJECT-X
Study: STUDY-0001
Log Overview
Generated: 10SEP2018

Log Message	Row ID
NOTE: Invalid third argument to function SUBSTR at line 6 column 176.	1
WARNING: Variable VAR1 has different lengths on BASE and DATA files (BASE 200 DATA 255).	2

HOW TO USE THE MACRO

The standard way to call the macro is to define the name of the selected libraries (or its path) for both data and for the metadata as parameters. As a result, the macro will generate the standard output defined, containing the 6 tabs with the overview and detailed description of each finding:

```
%adam_sdtm_compa (
    studyid= STUDY-0001,
    sdtm_path= W:\PROJECT-X\STUDY-0001\SDTM\Metadata,
    sdtm_data= W:\PROJECT-X\STUDY-0001\SDTM,
    adam_path= W:\PROJECT-X\STUDY-0001\ADAM\Metadata,
    adam_data= W:\PROJECT-X\STUDY-0001\ADAM
);
```

We purposely made the parameter's name uncomplicated and easy to specify. It also allows us to develop this macro further and add more options or parameters if needed.

ADVANTAGES

- ✓ Easy to produce: One macro to be run; creates one output, containing 6 useful tabs.
- ✓ Easy to integrate: Runs on SAS Base, SAS Studio or DI Studio.
- ✓ Easy to share: 1 XML file - Small output size (up to 2,000Kb).
- ✓ Easy to open: Opens with Excel – Requires basic Office software installation.
- ✓ Easy to read: Gives overview and detailed information – Tailored to the reader.
- ✓ Easy to archive: Can be printed and stored with the other project validation forms.

CHALLENGES AND IMPROVEMENT

A lot of false positive findings are generated in this report mainly due to 2 of the checks:

- Check #101 - due to the discrepancies in SDTM length compared to ADaM (allowed by the ADaM IG). A suggestion for improvement is to apply the same resize principle to the ADaM by trimming the datasets before doing the comparison. However, it would be more conservative to apply this manipulation on a temporary copy of the ADaM library in order to avoid any modification of the real data. By comparing this "corrected ADaM" length to the SDTM length it could be possible to flag the false positive finding and by filtering the report on this flag, it would simplify the review.
- Check #105 – due to the wrong identification of variable names (e.g. can't find the "Missing.Else" variable). A new working instruction has been defined in BDLS internal SOP related to the ADaM mapping process to avoid the use of the "." at the end of each phrase to make a hard stop but rather to use the ";" at the end of the statement, such as in the basic SAS programming language.

Another challenge is that this macro in its actual state is only "executable" if both SDTM and ADAM conversion have been done by BDLS using the latest version of BDLS mapping template as the metadata used to do the comparison is imported from the mapping and not extracted from the datasets itself or the define.xml files. Further development should be made in order to allow the macro to extract back the metadata in the Define.xml.

CONCLUSION

Following CDISC's principles, one of the main goals of the ADaM data and metadata is to provide traceability between the analysis results and the data represented in SDTM. There are multiple methods to create ADaM and one of these is the parallel method, in which the conversion of both sets of legacy datasets is done independently. The FDA does not recommend a particular approach to legacy clinical study data conversion. However, based on reviewer experience, establishing traceability is one of the most problematic issues associated with any data conversion.

BDLS has developed two macros over the last couple of years to use in addition to the standard operational procedures to ensure that some major principles are correctly applied during the CDISC conversion of the raw datasets and analysis datasets. The execution of additional automated checks does not replace any of the validation steps but has an additional value to improve the quality of the metadata and the traceability by generating some reports that are easy to read and to understand.

As metadata is at least as important as the data itself, the improvement of the quality and correctness of the traceability is an undoubtedly a return on investment, it helps not only the teams but also the reviewer to understand and validate the data manipulation made to obtain the study results.

REFERENCES

[1] Clinical Data Interchange Standards Consortium, "Analysis Data Model Implementation Guide", Version 1.0, Issued December 17, 2009. Available at https://www.cdisc.org/standards/foundational/adam, verified September 28, 2018.

[2] Clinical Data Interchange Standards Consortium, "Analysis Data Model Implementation Guide", Version 1.1, Issued February 12, 2016. Available at https://www.cdisc.org/standards/foundational/adam, verified September 28, 2018.

[3] Clinical Data Interchange Standards Consortium, "Study Data Tabulation Model Implementation Guide: Human Clinical Trials", Version 3.2, Issued November 26, 2013. Available at https://www.cdisc.org/standards/foundational/sdtm, verified September 28, 2018.

[4] US Department of Health and Human Services, U.S. Food and Drug Administration. "Technical Specifications Document: Study Data Technical Conformance Guide", Version 4.1, Issued March 31, 2018. Available at http://www.fda.gov/downloads/ForIndustry/DataStandards/StudyDataStandards/UCM384744.pdf, verified September 28, 2018.

ACKNOWLEDGMENTS

I wish to thank all my former and current BDLS colleagues who generously provided their support, insight and perspective that greatly helped to develop the presented macro and improve the quality of this paper, but in particular: Matt Davies, Jean-Michel Bodart, Geert Peel and Joëlle Périer.

CONTACT INFORMATION

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

Roxane DEBRUS
Business & Decision Life Sciences
Sint-Lambertusstraat 141
1200 Brussels
+32 475 95 35 00
roxane.debrus@businessdecision.be
https://www.businessdecision-lifesciences.com