SESUG 2023 Paper 226

For Clinical Trials: A Faster and Smoother Approach to Create your SDTM and ADAM Define Specifications for Define.xml with SAS®

Star Nze, M.S.

ABSTRACT

After final TLFs have been sent to the Sponsor, it's time for SAS programmers to create the SDTM and ADAM Define specs for the Define.xml file. Define spec creation can be a tedious task at times, especially with studies that have many different domains that were used for data analysis. Instead of referencing a previous study's define specs and manually entering data repetitively, we will explore a more programmatic way of completing the specs to simplify the process and help speed up task time for end-of-study FDA submission requirements. The use of Excel and SAS in tandem to pull annotations from the Case Report Form PDF document will be pivotal in aiding to streamline the Define specification process. The resulting excel file and SAS program will be transferable between studies with only minor updates being needed in the file and program for study-to-study usage.

INTRODUCTION

When clinical trials are conducted, clinical data is collected and analyzed using CDISC data standards. The purpose of CDISC (Clinical Data Interchange Standards Consortium) is to allow for effective global standardization and harmonization of clinical trial data interpretability across healthcare systems.

The following is an overview of the clinical trial data collection and analysis process:

- **CDASH** (Clinical Data Acquisition Standards Harmonization) standardizes the collection of clinical data
- **SDTM** (Study Data Tabulation Model) standardizes the organization and formatting of clinical data
- **ADAM** (Analysis Data Model) standardizes the performance of statistical analyses and traceability of results from SDTM datasets
- **SDTM IG/ADAM IG** (Implementation Guides-IG) are used to explain specific instances of clinical data assignment and use-cases for their respective models
- **SDTM CT/ADAM CT** (Controlled Terminology-CT) are pre-specified values that are used in CDISC-defined and CDISC-compliant datasets for reassigning and standardizing purposes and is required for datasets that are to be submitted to the FDA for their respective models
- **aCRF** (Annotated CRF) a form that is filled out by the study site that collects patient data during a clinical trial study. When annotated, the information in the CRF is mapped to codelists located in the SDTM CT Document

What is Define.xml?

Define.xml (Case Report Tabulation Data Definition Specification document) is a metadata document that explains and itemizes contents in the datasets that were collected during the process of a clinical trial. It helps the viewer understand the source of origination for the variables used in the SDTM datasets and how variables could relate to one another. This document is usually expected by the FDA to accompany file packages that are sent to them for study submissions. The Define-XML Specification document 2.1 version that we will be referencing in this paper is located on the CDISC website. The link to this document is listed in the References section of this paper.

What are Define Specifications?

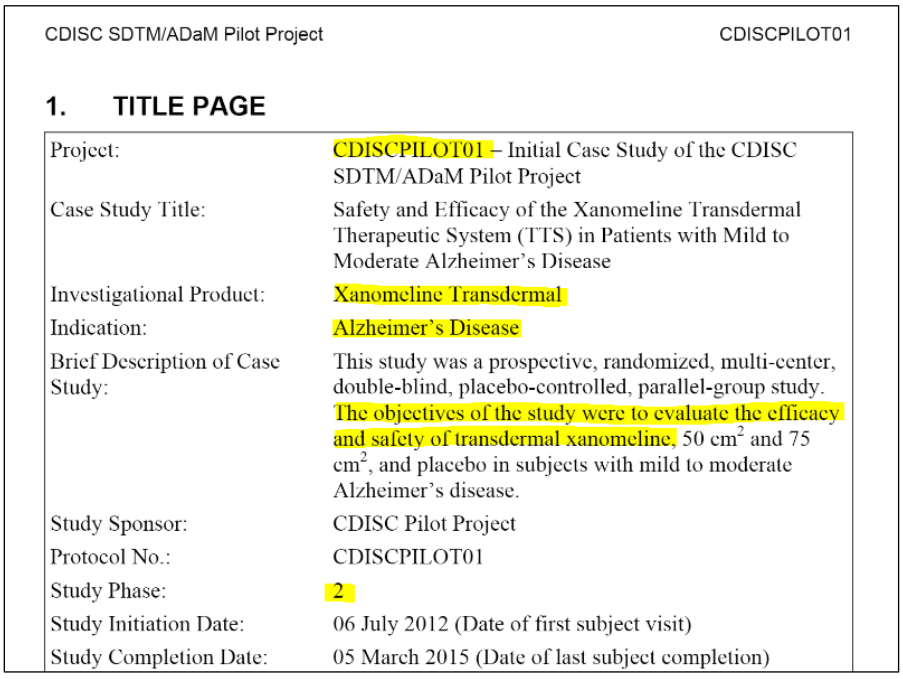
Define Specifications is an excel file that is to be completed prior to Define.xml creation. This file is created after the export files have been verified for CDISC IG and CT implementation accuracy with a data standard validator system and it is directly used for the final Define.xml document.

Pinnacle 21® Community will be used as the data standard compliance validator in this paper for references to outputs and file structures. It is not imperative that any one validator be used, but depending on the validator, excel tab names may vary. Nonetheless, equivalent tab names containing similar information should be present in the resulting file irrespective of the validation system used.

The purpose of this paper is to provide a foundation in which to streamline the Define Specifications creation process. Due to the repetitive nature of the task, it could be possible to create a program that will allow for some of the more time-consuming tasks of this file to be completed in minimal time. While the completion of the Specifications Document will not be autonomous without any intervention, if it is possible by any measurable amount to increase productivity and decrease task time, it is only then that one will have more time to build upon current foundations for increased future efficiency. Because SDTMs are the foundation to ADAMs and our goal is to take an introductory approach in the explanation of this process, we will be solely focusing on SDTM foundations, however, the process for ADAM Define Specifications will be identical to what is discussed throughout this paper. In some cases, the results used in the SDTM Specifications document can be used directly in the ADAM Specifications document, allowing for even more efficiency and faster completion of the document(s).

A DEEP DIVE INTO THE DEFNE SPECIFICATIONS DOCUMENT

In this paper, we will be referencing The CDISCPILOT01 Study with the datasets and study information available on the CDISC Pilot Project GitHub page (https://github.com/cdisc-org/sdtm-adam-pilot-project). A subset of the available datasets for this study will be used for illustrative purposes. A synopsis of the study is shown below.



CDISC SDTM/ADaM Pilot Project CDISCPILOT01

1. TITLE PAGE

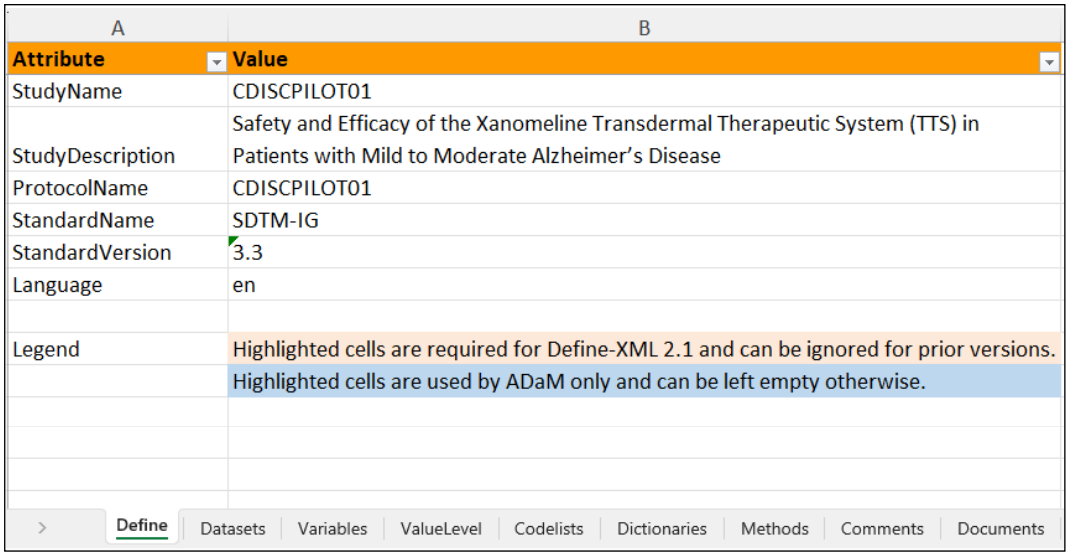
Project:	CDISCPILOT01 – Initial Case Study of the CDISC SDTM/ADaM Pilot Project
Case Study Title:	Safety and Efficacy of the Xanomeline Transdermal Therapeutic System (TTS) in Patients with Mild to Moderate Alzheimer's Disease
Investigational Product:	Xanomeline Transdermal
Indication:	Alzheimer's Disease
Brief Description of Case Study:	This study was a prospective, randomized, multi-center, double-blind, placebo-controlled, parallel-group study. The objectives of the study were to evaluate the efficacy and safety of transdermal xanomeline, 50 cm^2 and 75 cm^2, and placebo in subjects with mild to moderate Alzheimer's disease.
Study Sponsor:	CDISC Pilot Project
Protocol No.:	CDISCPILOT01
Study Phase:	2
Study Initiation Date:	06 July 2012 (Date of first subject visit)
Study Completion Date:	05 March 2015 (Date of last subject completion)

Figure 1: CDISCPILOT01 Study Synopsis

We will begin by assuming that there are no errors in the provided SDTM datasets which should be checked and resolved using a validator before proceeding. Next, using a validator, generate the SDTM Define Specifications excel file with the validated SDTM XPORT (.XPT) files. In the resulting excel file, you will find nine excel sheets (*Define, Datasets, Variables, ValueLevel, Codelists, Dictionaries, Methods, Comments, Documents*).

An overview of the contents of an SDTM Define Specifications Excel File:

- **Define** – a general overview of the clinical study and specifies which IG was used to analyze the study
- **Datasets** – a general overview of each of the SDTM datasets that were created for the study
- **Variables** – a detailed list of each variable contained within each SDTM dataset and its attributes
- **ValueLevel** – detailed list about each generated finding or result variable in each SDTM dataset
- **Codelists** – detailed list of all variables generated in the SDTM datasets that have an associated codelist in the Controlled Terminology document
- **Dictionaries** – itemizes any dictionaries used in the study for data collection and analysis
- **Methods** – itemizes any methods of numerical sorting or computations used throughout the study for data analysis purposes
- **Comments** – for any further comments to explain SDTM variables
- **Documents** – any .PDF documents that are to be included as part of the submission package



	A	B
1	**Attribute**	**Value**
2	StudyName	CDISCPILOT01
3	StudyDescription	Safety and Efficacy of the Xanomeline Transdermal Therapeutic System (TTS) in Patients with Mild to Moderate Alzheimer's Disease
4	ProtocolName	CDISCPILOT01
5	StandardName	SDTM-IG
6	StandardVersion	3.3
7	Language	en
8		
9	Legend	Highlighted cells are required for Define-XML 2.1 and can be ignored for prior versions.
10		Highlighted cells are used by ADaM only and can be left empty otherwise.

Define | Datasets | Variables | ValueLevel | Codelists | Dictionaries | Methods | Comments | Documents

Figure 2: CDISCPILOT01 SDTM Define Specifications document - Define Tab

Of the above listed tabs, the *Variables*, *ValueLevel*, and *Codelists* excel sheets are measurably the most time-consuming to complete. Since all three excel sheets can be completed in a similar manner once the first instance of the program is created, we will focus on the *Variables* excel sheet. The *Variables* excel sheet can be considered foundational for which all other sheets in the document will be based upon. We will be able to integrate the *Variables* excel sheet into a SAS program without too much difficulty and focusing on just this one tab will allow for easier comprehension of the program creation process. After the creation of the *Variables* portion of the SAS program and excel sheet output, one will have the freedom to then be creative in how to further implement code for the completion of the rest of the specification file.

SDTM DEFINE SPECIFICATIONS SAS PROGRAM

Step 1 – IMPORT FILES

To create this SAS program, ensure that you are using an SAS environment that is either up-to-date or will be compatible. In this paper, SAS Enterprise Guide 7.1 will be used. The first step in being able to create the SDTM Define Specifications SAS program is importing the excel sheets that are associated with each tab that we would like to work with as well as importing the SDTM CT and the SDTM IG files. For the SDTM CT we will be using the 2022-06-24 version and for the SDTM IG we will use the 3.3 version. The links to these documents are listed in the References section of this paper. Both have corresponding .XLSX files and these files can be downloaded from the CDISC library (https://library.cdisc.org/browser). To do this import, we will use the SAS XLSX engine. An example of how to import an excel sheet from an excel workbook using the XLSX engine is shown. Repeat this step for all the excel documents/files you would like to be imported into SAS.

```
/*Import an Excel Sheet into SAS*/
proc import datafile="\\source_path\source_folder\filename.xlsx" /*location of file to import*/
        dbms=xlsx /*xlsx engine selected*/
        out=work.sas_dataset_name /*assigns name of the output folder and new SAS dataset name*/
        replace;
        sheet=excel_sheet_name; /*the specific sheet name from the excel file*/
        getnames=yes; /*assigns first record as the column names*/
run;
```

We will import the *Variables* excel sheet from the SDTM Define Specifications Document, the SDTM CT, and the SDTM IG files into SAS.

For the **Variables** dataset, there are variables that were already populated by the dataset validator system from the imported datasets. We can use this pre-filled data to help us complete the rest of the **Variables** dataset. When completed, we will then export the completed dataset as an excel file and copy-paste the data into the original SDTM Define Specifications excel file. This will be the same step for each final dataset that is completed for each corresponding tab or excel sheet in the excel file. To make the explanation of the process easier to follow, we will be using only a subset of all the available sample datasets for this study in this paper. The remaining available datasets populated in the file and those omitted for ease of viewing will all follow the same principles. The following datasets shown are the subset of datasets that will be used. A portion of the current entries in the *Variables* excel sheet is also shown below.

Dataset	Label	Class	SubClass	Structure	Key Variables
AE	Adverse Events	EVENTS		One record per adverse event per subject	STUDYID, USUBJID, AEDECOD, AESTDTC
CM	Concomitant Medications	INTERVENTIONS		One record per recorded intervention occurrence or constan	STUDYID, USUBJID, CMTRT, CMSTDTC
DM	Demographics	SPECIAL PURPOSE		One record per subject	STUDYID, USUBJID
DS	Disposition	EVENTS		One record per disposition status or protocol milestone per	STUDYID, USUBJID, DSDECOD, DSSTDTC
LB	Laboratory Test Results	FINDINGS		One record per lab test per time point per visit per subject	STUDYID, USUBJID, LBTESTCD, VISITNUM
MH	Medical History	EVENTS		One record per medical history event per subject	STUDYID, USUBJID, MHDECOD
QS	Questionnaires	FINDINGS		One record per questionnaire per question per time point pe	STUDYID, USUBJID, QSCAT, QSTESTCD, VISITNUM
SUPPAE	Supplemental Qualifiers for AE	RELATIONSHIP		One record per IDVAR, IDVARVAL, and QNAM value per subj	STUDYID, RDOMAIN, USUBJID, IDVAR, IDVARVAL, QNAM
SUPPDM	Supplemental Qualifiers for DM	RELATIONSHIP		One record per IDVAR, IDVARVAL, and QNAM value per subj	STUDYID, RDOMAIN, USUBJID, IDVAR, IDVARVAL, QNAM
SUPPDS	Supplemental Qualifiers for DS	RELATIONSHIP		One record per IDVAR, IDVARVAL, and QNAM value per subj	STUDYID, RDOMAIN, USUBJID, IDVAR, IDVARVAL, QNAM
SUPPLB	Supplemental Qualifiers for LB	RELATIONSHIP		One record per IDVAR, IDVARVAL, and QNAM value per subj	STUDYID, RDOMAIN, USUBJID, IDVAR, IDVARVAL, QNAM
VS	Vital Signs	FINDINGS		One record per vital sign measurement per time point per vi	STUDYID, USUBJID, VSTESTCD, VISITNUM, VSTPTREF, VSTPTNUM

Define | Datasets | Variables | ValueLevel | Codelists | Dictionaries | Methods | Comments | Documents

Figure 3: CDISCPILOT01 SDTM Define Specifications document – Datasets Tab

A	B	C	D	E	F	G	H	I	J
Version	Variable Order	Class	Dataset Name	Variable Name	Variable Label	Type	CDISC CT Codelist Code(s)	Codelist Submission Values	Described Value Domain(s)
SDTMIG v3.3	1	Special-Purpose	CO	STUDYID	Study Identifier	Char			
SDTMIG v3.3	2	Special-Purpose	CO	DOMAIN	Domain Abbreviation	Char			
SDTMIG v3.3	3	Special-Purpose	CO	RDOMAIN	Related Domain Abbreviation	Char			
SDTMIG v3.3	4	Special-Purpose	CO	USUBJID	Unique Subject Identifier	Char			
SDTMIG v3.3	5	Special-Purpose	CO	COSEQ	Sequence Number	Num			
SDTMIG v3.3	6	Special-Purpose	CO	IDVAR	Identifying Variable	Char			
SDTMIG v3.3	7	Special-Purpose	CO	IDVARVAL	Identifying Variable Value	Char			
SDTMIG v3.3	8	Special-Purpose	CO	COREF	Comment Reference	Char			
SDTMIG v3.3	9	Special-Purpose	CO	COVAL	Comment	Char			
SDTMIG v3.3	10	Special-Purpose	CO	COEVAL	Evaluator	Char			
SDTMIG v3.3	11	Special-Purpose	CO	COEVALID	Evaluator Identifier	Char	C96777		
SDTMIG v3.3	12	Special-Purpose	CO	CODTC	Date/Time of Comment	Char			ISO 8601
SDTMIG v3.3	13	Special-Purpose	CO	CODY	Study Day of Comment	Num			
SDTMIG v3.3	1	Special-Purpose	DM	STUDYID	Study Identifier	Char			
SDTMIG v3.3	2	Special-Purpose	DM	DOMAIN	Domain Abbreviation	Char			
SDTMIG v3.3	3	Special-Purpose	DM	USUBJID	Unique Subject Identifier	Char			
SDTMIG v3.3	4	Special-Purpose	DM	SUBJID	Subject Identifier for the Study	Char			
SDTMIG v3.3	5	Special-Purpose	DM	RFSTDTC	Subject Reference Start Date/Time	Char			ISO 8601
SDTMIG v3.3	6	Special-Purpose	DM	RFENDTC	Subject Reference End Date/Time	Char			ISO 8601
SDTMIG v3.3	7	Special-Purpose	DM	RFXSTDTC	Date/Time of First Study Treatment	Char			ISO 8601
SDTMIG v3.3	8	Special-Purpose	DM	RFXENDTC	Date/Time of Last Study Treatment	Char			ISO 8601
SDTMIG v3.3	9	Special-Purpose	DM	RFICDTC	Date/Time of Informed Consent	Char			ISO 8601

Figure 3.1: CDISC SDTM Implementation Guide v3.3 .XLSX file

A	B	C	D	E	F	G
Code	Codelist Code	Codelist Extensible (Yes/No)	Codelist Name	CDISC Submission Value	CDISC Synonym(s)	CDISC Definition
C141657		No	10-Meter Walk/Run Functional Test Test Code	TENMW1TC	10-Meter Walk/Run Functional Test Test Code	10-Meter Walk/Run test code.
C174106	C141657		10-Meter Walk/Run Functional Test Test Code	TENMW101	TENMW1-Was Walk/Run Performed	10-Meter Walk/Run - Was the 10-meter walk/run performed?
C141700	C141657		10-Meter Walk/Run Functional Test Test Code	TENMW102	TENMW1-Time to Walk/Run 10 Meters	10-Meter Walk/Run - If yes, time to walk or run 10 meters.
C147592	C141657		10-Meter Walk/Run Functional Test Test Code	TENMW103	TENMW1-Wear Orthoses	10-Meter Walk/Run - If yes, did subject wear orthoses?
C141701	C141657		10-Meter Walk/Run Functional Test Test Code	TENMW104	TENMW1-Test Grade	10-Meter Walk/Run - Test grade.
C141656		No	10-Meter Walk/Run Functional Test Test Name	TENMW1TN	10-Meter Walk/Run Functional Test Test Name	10-Meter Walk/Run test name.
C141701	C141656		10-Meter Walk/Run Functional Test Test Name	TENMW1-Test Grade	TENMW1-Test Grade	10-Meter Walk/Run - Test grade.
C141700	C141656		10-Meter Walk/Run Functional Test Test Name	TENMW1-Time to Walk/Run 10 Meters	TENMW1-Time to Walk/Run 10 Meters	10-Meter Walk/Run - If yes, time to walk or run 10 meters.
C174106	C141656		10-Meter Walk/Run Functional Test Test Name	TENMW1-Was Walk/Run Performed	TENMW1-Was Walk/Run Performed	10-Meter Walk/Run - Was the 10-meter walk/run performed?
C147592	C141656		10-Meter Walk/Run Functional Test Test Name	TENMW1-Wear Orthoses	TENMW1-Wear Orthoses	10-Meter Walk/Run - If yes, did subject wear orthoses?
C141663		No	4-Stair Ascend Functional Test Test Code	A4STR1TC	4-Stair Ascend Functional Test Test Code	4-Stair Ascend test code.
C174103	C141663		4-Stair Ascend Functional Test Test Code	A4STR101	A4STR1-Was 4-Stair Ascend Performed	4-Stair Ascend - Was the 4-stair ascend performed?
C141706	C141663		4-Stair Ascend Functional Test Test Code	A4STR102	A4STR1-Time to Do 4-Stair Ascend	4-Stair Ascend - If yes, time taken to do 4-stair ascend.
C147590	C141663		4-Stair Ascend Functional Test Test Code	A4STR103	A4STR1-Wear Orthoses	4-Stair Ascend - If yes, did subject wear orthoses?
C141707	C141663		4-Stair Ascend Functional Test Test Code	A4STR104	A4STR1-Test Grade	4-Stair Ascend - Test grade.
C141662		No	4-Stair Ascend Functional Test Test Name	A4STR1TN	4-Stair Ascend Functional Test Test Name	4-Stair Ascend test name.

Figure 3.2: CDISC SDTM Controlled Terminology, 2022-06-24 .XLSX file

Order	Dataset	Variable	Label	Data Type	Length	Significant Digit	Format	Mandatory	Assigned Value	Codelist	Common	Origin
1	AE	STUDYID	Study Identifier	text	12			Yes				
2	AE	DOMAIN	Domain Abbreviation	text	2			Yes				
3	AE	USUBJID	Unique Subject Identifier	text	11			No				
4	AE	AESEQ	Sequence Number	integer	8			Yes				
5	AE	AESPID	Sponsor-Defined Identifier	text	3			No				
6	AE	AETERM	Reported Term for the Adve	text	200			Yes				
7	AE	AELLT	Lowest Level Term	text	100			No				
8	AE	AELLTCD	Lowest Level Term Code	integer	8			No				
9	AE	AEDECOD	Dictionary-Derived Term	text	200			Yes				
10	AE	AEPTCD	Preferred Term Code	integer	8			No				
11	AE	AEHLT	High Level Term	text	100			No				
12	AE	AEHLTCD	High Level Term Code	integer	8			No				
13	AE	AEHLGT	High Level Group Term	text	100			No				
14	AE	AEHLGTCD	High Level Group Term Cod	integer	8			No				
15	AE	AEBODSYS	Body System or Organ Class	text	67			No				
16	AE	AEBDSYCD	Body System or Organ Class	integer	8			No				
17	AE	AESOC	Primary System Organ Class	text	100			No				
18	AE	AESOCCD	Primary System Organ Class	integer	8			No				
19	AE	AESEV	Severity/Intensity	text	8			No				
20	AE	AESER	Serious Event	text	1			No				
21	AE	AEACN	Action Taken with Study Tr	text	30			No				
22	AE	AEREL	Causality	text	8			No				
23	AE	AEOUT	Outcome of Adverse Event	text	200			No				
24	AE	AESCAN	Involves Cancer	text	1			No				

Define | Datasets | Variables | ValueLevel | Codelists | Dictionaries | Methods | Comments | Documents

Figure 4: CDISCPILOT01 SDTM Define Specifications document – Variables Tab (Note: This will be the main dataset)

After importing the necessary files into SAS as newly created datasets, we will retrieve and import the aCRF information for this study. A page of the clinical report form (CRF) with annotations for our example study is shown.

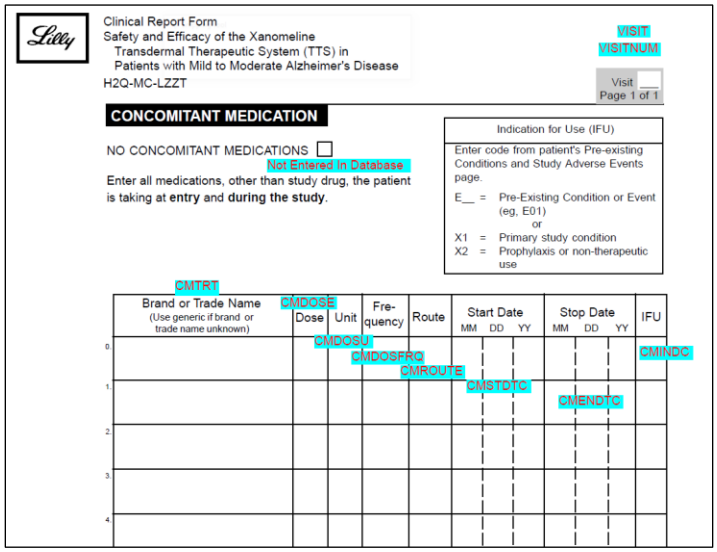


Figure 5: CDISCPILOT01 Annotated Clinical Report Form (aCRF)

An efficient way to import the aCRF data is to first download a metadata document from a pdf viewing program that includes annotations and page numbers of these annotations in the file. If using Adobe Acrobat then the file will be saved as an .FDF file. Open notepad and drag the .FDF file into notepad. Next, open the SDTM CT excel file and create a new tab called *Pages*. Copy what is in the notepad window and paste it into the SDTM CT *Pages* tab. Finally, save the SDTM CT file. We can now import the *Pages* tab into SAS to create a **Pages** dataset.

Figure 6: Imported aCRF into SDTM CT Excel file – Pages Tab

```
/*Importing Variables Tab*/

proc import datafile="\\source_path\source_folder\cdiscpilot01 study\SDTM Define
Specifications.xlsx"
       dbms=xlsx
       out=work.variables
       replace;
       sheet=Variables;
       getnames=yes;
run;

/*Importing SDTMCT*/
proc import datafile="\\source_path\source_folder\cdiscpilot01 study\sdtmct_20220624.xlsx"
       dbms=xlsx
       out=work.sdtmct
       replace;
       sheet=Terminology;
       getnames=yes;
run;

/*Importing SDTMIG*/
proc import datafile="\\source_path\source_folder\cdiscpilot01 study\SDTMIG_v3.3.xlsx"
       dbms=xlsx
       out=work.sdtmig
       replace;
       sheet=sdtmigv3_3;
       getnames=yes;
run;

/*Importing aCRF Page Numbers*/
proc import datafile="\\source_path\source_folder\cdiscpilot01 study\sdtmct_20220624.xlsx"
       dbms=xlsx
       out=work.pages
       replace;
       sheet=Pages;
       getnames=no;
run;
```

Step 2 – PREEMPTIVE DATA PRESERVATION

To preserve the order of the column names found in the *Variables* excel sheet from the SDTM Define Specifications document for our final programmed SAS dataset(s), we will create an ATTRIB macro called "variable_order", this macro that will be called later in the program. To also preserve the original order of the imported data from the SDTM Define Specifications document, in a new dataset called **Variables1**, we will create an ascending number variable unique to each row of the dataset called "new_ord".

```
/*Specifying the order of the column names in the Variables Tab found in the SDTM
Specifications Excel File*/
%macro variable_order;
attrib
Order label='var1'
Dataset label='var2'
Variable label='var3'
Label label='var4'
'Data Type'n label='var5'
Length label='var6'
'Significant Digits'n label='var7'
Format label='var8'
Mandatory label='var9'
'Assigned Value'n label='var10'
Codelist label='var11'
Common label='var12'
Origin label='var13'
Source label='var14'
Pages label='var15'
Method label='var16'
Predecessor label='var17'
Role label='var18'
'Has No Data'n label='var19'
Comment label='var20'
'Developer Notes'n label='var21'
;
%mend variable_order;


/*Preserving the inital order of the rows from the Excel File by creating an ascending
order variable for up to N total rows*/
data variables1;
       set variables;

       new_ord=_n_;
run;
```

Step 3 – CREATION OF COLUMN-VARIABLES

There will be six variables that will be created for the excel file in this program: "Pages", "Codelist", "Format", "Method", "Origin", and "Source". Please note that not all variables shown in the SDTM Define Specifications document will need to be completed. Completion of variables depends on the version of resources used and the SDTM variable definition procedures.

PAGES

To create the "Pages" variable, we will have to parse the dataset to extract annotations and the pages those annotations are on. Any annotation and page number replications should be filtered out. Data formatting and cleaning to ensure uniformity for all the entries may be necessary. Perform data manipulation, either by transposing or by use of macros, to create a single column of all SDTM variable names and a new column for each unique page number for each variable name. Next, create a macro or use data step programming, to iterate through all the available columns for each page number entry. This macro will result in one concatenated space-delimited list variable containing all the page numbers. We will then clean up the dataset and fix naming conventions for the finalized "Pages" variable. Merge the resulting **Pages1** dataset onto the **Variables1** dataset, we will call this new dataset **UPDATED_VARS** dataset throughout the paper. Some of the intermediary datasets, the final resulting dataset, and the code for this portion are shown.

PAGES1

Filter and Sort | Query Builder | Where | Data | Describe

	var1	var2	vars	pgs
1	<</BS 3694 0...	<</BS 3694 0...	--DTC [AEDTC...	6
2	<</BS 3625 0...	<</BS 3625 0...	--DTC [AEDTC...	21
3	<</BS 3612 0...	<</BS 3612 0...	--DTC [AEDTC...	24
4	<</BS 3592 0...	<</BS 3592 0...	--DTC [AEDTC...	31
5	<</BS 3575 0...	<</BS 3575 0...	--DTC [AEDTC...	35
6	<</BS 3547 0...	<</BS 3547 0...	--DTC [AEDTC...	41
7	<</BS 3511 0...	<</BS 3511 0...	--DTC [AEDTC...	48
8	<</BS 3497 0...	<</BS 3497 0...	--DTC [AEDTC...	51
9	<</BS 3467 0...	<</BS 3467 0...	--DTC [AEDTC...	57
10	<</BS 3440 0...	<</BS 3440 0...	--DTC [AEDTC...	66
11	<</BS 3411 0...	<</BS 3411 0...	--DTC [AEDTC...	72
12	<</BS 3384 0...	<</BS 3384 0...	--DTC [AEDTC...	81
13	<</BS 3356 0...	<</BS 3356 0...	--DTC [AEDTC...	87
14	<</BS 3352 0...	<</BS 3352 0...	--DTC [QSDTC...	89
15	<</BS<</W 0>...	<</BS<</W 0>...	--DTC [QSDTC...	107
16	<</BS<</W 0>...	<</BS<</W 0>...	--DTC [QSDTC...	98
17	<</BS<</W 0>...	<</BS<</W 0>...	--DTC [QSDTC...	115
18	<</BS<</W 0>...	<</BS<</W 0>...	--DTC [QSDTC...	127
19	<</BS<</W 0>...	<</BS<</W 0>...	--ENDTC [AEE...	120
20	<</BS<</W 0>...	<</BS<</W 0>...	--ENDTC [AEE...	121
21	<</BS<</W 0>...	<</BS<</W 0>...	--ENDTC [AEE...	122
22	<</BS<</W 0>...	<</BS<</W 0>...	--SEV [AESEV,...	120
23	<</BS<</W 0>...	<</BS<</W 0>...	--SPID [AESPI...	120
24	<</BS<</W 0>...	<</BS<</W 0>...	--SPID [AESPI...	121
25	<</BS<</W 0>...	<</BS<</W 0>...	--SPID [AESPI...	122
26	<</BS<</W 0>...	<</BS<</W 0>...	--STDTC [AES...	120
27	<</BS<</W 0>...	<</BS<</W 0>...	--STDTC [AES...	121
28	<</BS<</W 0>...	<</BS<</W 0>...	--STDTC [AES...	122
29	<</BS<</W 0>...	<</BS<</W 0>...	--STDTC [SVS...	115
30	<</BS<</W 0>...	<</BS<</W 0>...	--STDTC [SVS...	127
31	<</BS<</W 0>...	<</BS<</W 0>...	--STDTC [SVS...	6
32	<</BS<</W 0>...	<</BS<</W 0>...	--STDTC [SVS...	21
33	<</BS<</W 0>...	<</BS<</W 0>...	--STDTC [SVS...	24
34	<</BS<</W 0>...	<</BS<</W 0>...	--STDTC [SVS...	31
35	<</BS<</W 0>...	<</BS<</W 0>...	--STDTC [SVS...	35
36	<</BS<</W 0>...	<</BS<</W 0>...	--STDTC [SVS...	41
37	<</BS<</W 0>...	<</BS<</W 0>...	--STDTC [SVS...	48
38	<</BS<</W 0>...	<</BS<</W 0>...	--STDTC [SVS...	51
39	<</BS<</W 0>...	<</BS<</W 0>...	--STDTC [SVS...	57
40	<</BS<</W 0>...	<</BS<</W 0>...	--STDTC [SVS...	66
41	<</BS<</W 0>...	<</BS<</W 0>...	--STDTC [SVS...	72
42	<</BS<</W 0>...	<</BS<</W 0>...	--STDTC [SVS...	81
43	<</BS<</W 0>...	<</BS<</W 0>...	--STDTC [SVS...	87
44	<</BS<</W 0>...	<</BS<</W 0>...	--STDTC [SVS...	89

Figure 7: Parsing dataset for data extraction – Pages column

PAGES2_1

Filter and Sort | Query Builde

	vars	pgs
1	AEREL	120
2	AEREL	121
3	AEREL	122
4	AESCAN	120
5	AESCAN	121
6	AESCAN	122
7	AESCONG	120
8	AESCONG	121
9	AESCONG	122
10	AESDISAB	120
11	AESDISAB	121
12	AESDISAB	122
13	AESDTH	120
14	AESDTH	121
15	AESDTH	122
16	AESER	120
17	AESER	121
18	AESER	122
19	AESEV	121
20	AESEV	122
21	AESHOSP	120
22	AESHOSP	121
23	AESHOSP	122
24	AESLIFE	120
25	AESLIFE	121
26	AESLIFE	122
27	AESMIE	120
28	AESMIE	121
29	AESMIE	122
30	AESOD	120
31	AESOD	121
32	AESOD	122
33	AESPID	105
34	AESPID	138
35	CMDOSE	123
36	CMDOSE	124
37	CMDOSE	125

Figure 8: Extracted annotation variables and page numbers – Pages column

PAGES3_1

Filter and Sort | Query Builder | Where | Data | Describe | Graph | Analyze | Export | Send To

	vars	_NAME_	COL1	COL2	COL3	COL4	COL5	COL6	COL7	COL8	COL9	COL10	COL11	COL12
1	AEDTC	pgs	6	21	24	31	35	41	48	51	57	66	72	81
2	AEENDTC	pgs	120	121	122	.	.	.	.	.	.	.	.	.
3	AEREL	pgs	120	121	122	.	.	.	.	.	.	.	.	.
4	AESCAN	pgs	120	121	122	.	.	.	.	.	.	.	.	.
5	AESCONG	pgs	120	121	122	.	.	.	.	.	.	.	.	.
6	AESDISAB	pgs	120	121	122	.	.	.	.	.	.	.	.	.
7	AESDTH	pgs	120	121	122	.	.	.	.	.	.	.	.	.
8	AESER	pgs	120	121	122	.	.	.	.	.	.	.	.	.
9	AESEV	pgs	120	121	122	.	.	.	.	.	.	.	.	.
10	AESHOSP	pgs	120	121	122	.	.	.	.	.	.	.	.	.
11	AESLIFE	pgs	120	121	122	.	.	.	.	.	.	.	.	.
12	AESMIE	pgs	120	121	122	.	.	.	.	.	.	.	.	.
13	AESOD	pgs	120	121	122	.	.	.	.	.	.	.	.	.
14	AESPID	pgs	105	120	121	122	138	.	.	.	.	.	.	.
15	AESTDTC	pgs	115	120	121	122	127	.	.	.	.	.	.	.
16	AETERM	pgs	120	121	122	.	.	.	.	.	.	.	.	.
17	CMDOSE	pgs	123	124	125	.	.	.	.	.	.	.	.	.
18	CMDOSFRQ	pgs	123	124	125	.	.	.	.	.	.	.	.	.
19	CMDOSU	pgs	123	124	125	.	.	.	.	.	.	.	.	.
20	CMDTC	pgs	6	21	24	31	35	41	48	51	57	66	72	81
21	CMENDTC	pgs	123	124	125	.	.	.	.	.	.	.	.	.
22	CMINDC	pgs	123	124	125	.	.	.	.	.	.	.	.	.
23	CMROUTE	pgs	123	124	125	.	.	.	.	.	.	.	.	.
24	CMSTDTC	pgs	123	124	125	.	.	.	.	.	.	.	.	.
25	CMTRT	pgs	123	124	125	.	.	.	.	.	.	.	.	.
26	DMDTC	pgs	6	.	.	.	.	.	.	.	.	.	.	.
27	DSDECOD	pgs	105	138	.	.	.	.	.	.	.	.	.	.
28	DSDTC	pgs	6	21	24	31	35	41	48	51	57	66	72	81
29	DSSTDTC	pgs	6	21	24	31	35	41	48	51	57	66	72	81
30	DSTERM	pgs	105	138	.	.	.	.	.	.	.	.	.	.
31	EXENDTC	pgs	104	137	.	.	.	.	.	.	.	.	.	.
32	EXSTDTC	pgs	24	31	35	41	48	51	57	66	72	81	87	89
33	MHDTC	pgs	6	.	.	.	.	.	.	.	.	.	.	.
34	MHENDTC	pgs	120	121	122	.	.	.	.	.	.	.	.	.
35	MHSEV	pgs	120	.	.	.	.	.	.	.	.	.	.	.
36	MHSPID	pgs	13	120	121	122	.	.	.	.	.	.	.	.
37	MHSTDTC	pgs	11	13	14	120	121	122	.	.	.	.	.	.
38	MHTERM	pgs	11	13	14	120	121	122	.	.	.	.	.	.
39	QSCAT	pgs	9	10	25	26	27	37	43	53	58	59	60	61
40	QSDTC	pgs	6	21	24	31	35	41	48	51	57	66	72	81
41	QSORRES	pgs	9	10	25	26	27	28	37	43	53	58	59	60
42	QSSCAT	pgs	9	26	27	28	37	43	53	60	61	62	68	75
43	QSTESTCD	pgs	9	10	25	26	27	28	37	43	53	58	59	60
44	RACE	pgs	6	.	.	.	.	.	.	.	.	.	.	.
45	SCDTC	pgs	6	.	.	.	.	.	.	.	.	.	.	.

Figure 9: Dataset prior to use of the iteration macro – Pages column

PAGES3_3

Filter and Sort | Query Builder | Where | Data | Describe | Gra

	pages1	variable
1	6 21 24 31 35 41 48 51 57 66 72 81 87 89 98 115 127	AEDTC
2	120 121 122	AEENDTC
3	120 121 122	AEREL
4	120 121 122	AESCAN
5	120 121 122	AESCONG
6	120 121 122	AESDISAB
7	120 121 122	AESDTH
8	120 121 122	AESER
9	120 121 122	AESEV
10	120 121 122	AESHOSP
11	120 121 122	AESLIFE
12	120 121 122	AESMIE
13	120 121 122	AESOD
14	105 120 121 122 138	AESPID
15	115 120 121 122 127	AESTDTC
16	120 121 122	AETERM
17	123 124 125	CMDOSE
18	123 124 125	CMDOSFRQ
19	123 124 125	CMDOSU
20	6 21 24 31 35 41 48 51 57 66 72 81 87 89 98 115 127	CMDTC

Figure 10: Concatenated page numbers and annotation variables – Pages column

```
/*Formatting pages dataset to extract the variable names and the associated
page numbers*/
data pages1;
      set pages(rename=(A=var1));

       if findw(var1,'Contents') ne 0;
       if findw(var1,'Not Entered In Database') = 0;
       var2=var1;

       vars=strip(scan(scan(substr(var2, index(var2,
'Contents(')),2,'(''þÿ'),1,')''']''when''='));

       vars=tranwrd(vars,'\r','');

       pgs=input(strip(scan(scan(substr(var2, index(var2, 'Page')),1,'/'),2,'
')),10.);

run;


proc sort nodupkey data=pages1;
      by vars pgs ;
run;


/*macro to iterate through each list item and create a new column and stack
as new dataset*/
%macro scanvar;

%do i=1 %to 8;

data pages1x_&i;
      set pages1_1;
      vars=scan(varsx,&i,',');output;

      drop varsx;

      %end;

run;

data pages1_2;
      set pages1x_:;
      where vars ne '';
run;

proc sort data=pages1_2;
by pgs vars;
run;

%mend scanvar;

%scanvar;
```

```
/*transposing dataset to stack page numbers horizontally*/

proc transpose data=pages3 out=pages3_1;
      by vars;
      var pgs;
run;

/*macro to combine all page numbers per variable delimited by a space*/
%macro stackpgs;
data pages3_2;
      set pages3_1;

      pgs=strip(col1)||' '||
      %do i=2 %to 40-1;
      strip(col&i)||' '||
      %end;
      strip(col40);

run;
%mend stackpgs;

%stackpgs;

/*final pages result*/
data pages3_3;
      length pages1 variable $150.;
      set pages3_2;

      variable=upcase(vars);
      pages1=tranwrd(pgs,'.','');

      keep pages1 variable;

run;



/*adding the pages to the imported Variables dataset*/
data pages_final;
      length pages1 variable $150.;
      merge variables1(in=a) pages3_3;
      by variable;
      if a;

      drop pages;
      rename pages1=pages;
run;
```

CODELIST AND FORMAT

To create the "Codelist" and "Format" variables, we will create a new dataset from the **sdtmig** dataset called **sdtmig1** and create new temp variables "codelistx" and "formatx" keeping only where entries are non-missing. Next, remove duplicates and merge this new dataset onto the **UPDATED_VARS** dataset. We will then create a new dataset from the **sdtmct** dataset called **sdtmct1**. We will assign new variables "codelistx" and "codelist_name". Next, remove duplicates and merge this new dataset onto the **UPDATED_VARS** dataset by "codelistx". We will then clean up the dataset and fix naming conventions for the finalized "Codelist" and "Format" variables. Some of the intermediary datasets, the final resulting dataset, and the code for this portion are shown.

SDTMIG1

Filter and Sort | Query Builder | Where | D

	variable	codelistx	formatx
1	AEACN	C66767	
2	AEBDSYCD		MedDRA
3	AECONTRT	C66742	
4	AEDECOD		MedDRA
5	AEDUR		ISO 8601
6	AEENDTC		ISO 8601
7	AEENRF	C66728	
8	AEENRTPT	C66728	
9	AEHLGT		MedDRA
10	AEHLGTCD		MedDRA
11	AEHLT		MedDRA
12	AEHLTCD		MedDRA
13	AELLT		MedDRA
14	AELLTCD		MedDRA
15	AELOC	C74456	
16	AEOUT	C66768	
17	AEPRESP	C66742	
18	AEPTCD		MedDRA
19	AESCAN	C66742	
20	AESCONG	C66742	
21	AESDISAB	C66742	
22	AESDTH	C66742	
23	AESER	C66742	
24	AESEV	C66769	
25	AESHOSP	C66742	
26	AESLIFE	C66742	
27	AESMIE	C66742	
28	AESOC		MedDRA
29	AESOCCD		MedDRA
30	AESOD	C66742	
31	AESTDTC		ISO 8601
32	AGDOSFRM	C66726	
33	AGDOSFRQ	C71113	
34	AGDOSU	C71620	
35	AGDUR		ISO 8601
36	AGENDTC		ISO 8601
37	AGENRF	C66728	
38	AGENRTPT	C66728	
39	AGEU	C66781	
40	AGOCCUR	C66742	
41	AGPRESP	C66742	
42	AGROUTE	C66729	
43	AGSTAT	C66789	
44	AGSTDTC		ISO 8601
45	AGSTRF	C66728	

Figure 11: SDTM IG dataset with new assigned variables – Codelist and Format columns

MERGED_PG_IG

Filter and Sort | Query Builder | Where | Data | Describe | Graph | Analyze | Export | Send To

	pages	codelistx	formatx	variable	Order	Dataset	Label	Data Type
1				ACTARM	22	DM	Description of...	text
2				ACTARMCD	21	DM	Actual Arm Co...	text
3			MedDRA	AEBDSYCD	16	AE	Body System o...	integer
4				AEBODSYS	15	AE	Body System o...	text
5			MedDRA	AEDECOD	9	AE	Dictionary-Deri...	text
6	6 21 24 31 35 4...			AEDTC	31	AE	Date/Time of C...	datetime
7	120 121 122		ISO 8601	AEENDTC	33	AE	End Date/Time...	datetime
8				AEENDY	35	AE	Study Day of E...	integer
9			MedDRA	AEHLGT	13	AE	High Level Gro...	text
10			MedDRA	AEHLGTCD	14	AE	High Level Gro...	integer
11			MedDRA	AEHLT	11	AE	High Level Term	text
12			MedDRA	AEHLTCD	12	AE	High Level Ter...	integer
13			MedDRA	AELLT	7	AE	Lowest Level T...	text
14			MedDRA	AELLTCD	8	AE	Lowest Level T...	integer
15			MedDRA	AEPTCD	10	AE	Preferred Term...	integer
16	120 121 122			AEREL	22	AE	Causality	text
17				AESEQ	4	AE	Sequence Num...	integer
18			MedDRA	AESOC	17	AE	Primary Syste...	text
19			MedDRA	AESOCCD	18	AE	Primary Syste...	integer
20	105 120 121 1...			AESPID	5	AE	Sponsor-Defin...	text
21	115 120 121 1...		ISO 8601	AESTDTC	32	AE	Start Date/Tim...	datetime
22				AESTDY	34	AE	Study Day of S...	integer
23	120 121 122			AETERM	6	AE	Reported Term...	text
24				AGE	14	DM	Age	integer
25				ARM	20	DM	Description of...	text
26				ARMCD	19	DM	Planned Arm C...	text
27				CMCLAS	9	CM	Medication Cla...	text
28				CMDECOD	7	CM	Standardized...	text
29	123 124 125			CMDOSE	10	CM	Dose per Admi...	float
30	6 21 24 31 35 4...			CMDTC	17	CM	Date/Time of C...	datetime
31	123 124 125		ISO 8601	CMENDTC	19	CM	End Date/Time...	datetime
32				CMENDY	21	CM	Study Day of E...	integer
33	123 124 125			CMINDC	8	CM	Indication	text
34				CMSEQ	4	CM	Sequence Num...	integer
35				CMSPID	5	CM	Sponsor-Defin...	text
36	123 124 125		ISO 8601	CMSTDTC	18	CM	Start Date/Tim...	text
37				CMSTDY	20	CM	Study Day of S...	integer
38	123 124 125			CMTRT	6	CM	Reported Nam...	text
39			ISO 3166-1 Alp...	COUNTRY	23	DM	Country	text
40	6		ISO 8601	DMDTC	24	DM	Date/Time of C...	datetime
41				DMDY	25	DM	Study Day of C...	integer
42				DOMAIN	2	AE	Domain Abbrev...	text
43				DOMAIN	2	CM	Domain Abbrev...	text
44				DOMAIN	2	DM	Domain Abbrev...	text
45				DOMAIN	2	DS	Domain Abbrev...	text

Figure 12: SDTM IG dataset merged with main dataset – Codelist and Format columns

SDTMCT1

Filter and Sort | Query Builder | Wh

	codelistx	codelist_name
1	C100000	PERCUTANEOUS C...
2	C100001	PERCUTANEOUS C...
3	C100002	PERCUTANEOUS C...
4	C100003	PERCUTANEOUS MI...
5	C100004	PERICARDIAL STRIP...
6	C100005	POST-CARDIAC TRA...
7	C100006	PRE-OPERATIVE EV...
8	C100007	PREVIOUSLY IMPLA...
9	C100008	RESCUE PERCUTAN...
10	C100011	RIGHT VENTRICULA...
11	C100014	SPONTANEOUS SU...
12	C100015	STAGED PERCUTAN...
13	C100016	SUBJECT DELAY IN...
14	C100017	SURGICAL MAZE
15	C100018	SYNCOPE WITH HIG...
16	C100019	SYNCOPE WITH IND...
17	C100020	THREE VESSEL DIS...
18	C100021	TIMIFLOW
19	C100022	TRANSCATHETER A...
20	C100023	TWO VESSEL DISEA...
21	C100024	TYPICAL CORONAR...
22	C100025	COULD NOT OBTAIN...
23	C100026	UNABLE TO OBTAIN...
24	C100027	UNABLE TO OBTAIN...
25	C100028	UNABLE TO POSITIO...
26	C100029	UNABLE TO POSITIO...
27	C100030	UPGRADE TO A DEV...
28	C100031	URGENT
29	C100032	LSNCPCLS
30	C100033	CARDIAC ARREST/A...
31	C100034	LEAD DISLODGEME...
32	C100035	FAULTY CONNECTO...
33	C100036	IDIOPATHIC PRIMAR...
34	C100037	ACC/AHA LESION C...
35	C100038	ACC/AHA LESION C...
36	C100039	ACC/AHA LESION C...

Figure 13: SDTM CT dataset with new assigned variables – Codelist and Format columns

MERGED_PG_IG_CT1

Filter and Sort | Query Builder | Where | Data | Describe | Graph | Analyze | Expo

	codelist	format	variable	pages	Order	Data
1			ACTARM		22	DM
2			ACTARMCD		21	DM
3	MedDRA		AEBDSYCD		16	AE
4			AEBODSYS		15	AE
5	MedDRA		AEDECOD		9	AE
6			AEDTC	6 21 24 31 35 4...	31	AE
7		ISO 8601	AEENDTC	120 121 122	33	AE
8			AEENDY		35	AE
9	MedDRA		AEHLGT		13	AE
10	MedDRA		AEHLGTCD		14	AE
11	MedDRA		AEHLT		11	AE
12	MedDRA		AEHLTCD		12	AE
13	MedDRA		AELLT		7	AE
14	MedDRA		AELLTCD		8	AE
15	MedDRA		AEPTCD		10	AE
16			AEREL	120 121 122	22	AE
17			AESEQ		4	AE
18	MedDRA		AESOC		17	AE
19	MedDRA		AESOCCD		18	AE
20			AESPID	105 120 121 1...	5	AE
21		ISO 8601	AESTDTC	115 120 121 1...	32	AE
22			AESTDY		34	AE
23			AETERM	120 121 122	6	AE
24			AGE		14	DM
25			ARM		20	DM
26			ARMCD		19	DM
27			CMCLAS		9	CM
28			CMDECOD		7	CM
29			CMDOSE	123 124 125	10	CM
30			CMDTC	6 21 24 31 35 4...	17	CM
31		ISO 8601	CMENDTC	123 124 125	19	CM
32			CMENDY		21	CM
33			CMINDC	123 124 125	8	CM
34			CMSEQ		4	CM
35			CMSPID		5	CM
36		ISO 8601	CMSTDTC	123 124 125	18	CM
37			CMSTDY		20	CM
38			CMTRT	123 124 125	6	CM
39		ISO 3166-1 Alp...	COUNTRY		23	DM
40		ISO 8601	DMDTC	6	24	DM
41			DMDY		25	DM
42			DOMAIN		2	AE
43			DOMAIN		2	CM
44			DOMAIN		2	DM
45			DOMAIN		2	DS

Figure 14: SDTM CT dataset merged with main dataset – Codelist and Format columns

```
/*renaming a variable to merge*/
data sdtmig1;
      set sdtmig;

      variable=strip('Variable Name'n);
      codelistx=scan(strip('CDISC CT Codelist Code(s)'n),1,';');
      formatx=strip('Described Value Domain(s)'n);

      where 'CDISC CT Codelist Code(s)'n ne '' or 'Described Value
Domain(s)'n ne '';

      keep variable codelistx formatx;
run;

proc sort data=sdtmig1 nodupkey;
      by  variable codelistx formatx;
run;

/*merging sdtmig data onto resulting new Variable dataset with included pages
column*/
data merged_pg_ig;
      merge pages_final(in=a) sdtmig1;
      by variable;
      if a;
run;

proc sort data=merged_pg_ig;
      by codelistx;
run;

/*assigning variables for data manipulation*/
data sdtmct1;
      set sdtmct;

      codelistx=strip(code);
      codelist_name=strip('CDISC Submission Value'n);

      keep codelistx codelist_name;
run;

proc sort data=sdtmct1 nodupkey;
      by codelistx;
run;

/*merging sdtmct data onto resulting new current Variable dataset with added
IG data*/
data merged_pg_ig_ct;
      length variable $150.;
      merge merged_pg_ig(in=a) sdtmct1;
      by codelistx;
      if a;
run;

/*completion of adding Format and Codelist Variables*/
data merged_pg_ig_ct1;
      set merged_pg_ig_ct;
```

```
      codelist_name1=codelist_name;
      formatx1=formatx;

      if formatx='MedDRA' then codelist_name1=formatx;
      if formatx='MedDRA' then formatx1='';

      drop codelist format codelist_name formatx codelistx;
      rename codelist_name1=codelist formatx1=format;
run;
```

METHOD, ORIGIN, AND SOURCE

To create the "Method", "Origin", and "Source" variables, we will create a new dataset from the resulting **UPDATED_VARS** dataset from the previous section. We will reference the Define-XML Specification document that was introduced earlier in this paper for the assignment rules of the variables depending on the entries in the related columns. An excerpt for the assignment rules from the Define-XML Specification document is shown.

Type	Definition
Collected	A value that is actually observed and recorded by a person or obtained by an instrument. Note that a collected entry translated to a synonymous controlled term still has a type Collected.
Derived	A value that is calculated by an algorithm or reproducible rule, and which is dependent upon other data values, including data values available within the dataset or externally provided data values. MethodDef must be used to document the algorithm or rule used for a derived value.
Assigned	Data that is either: Determined by individual judgment as provided by an evaluator, or Coded terms supplied as part of a coding process, or Values set independently of any subject-related data value in order to complete a dataset.
Protocol	Data that is defined as part of the study protocol, investigator instructions, standard operating procedures or trial design preparation
Predecessor	An entry that is copied from a variable in another dataset. The Description child element identifies the dataset and variable that is copied.
Not Available	Used when the origin is not available and cannot be determined. Sponsors should specify additional details that may be helpful to the reviewer in the Comments section of the data definition file.

Figure 15: Guide for Origin-column assignment – Method, Origin, and Source columns (CDISC Define-XML Specification v2.1 pg. 30)

Type	Source				Notes
	Subject	Investigator	Vendor	Sponsor	
Collected	ePro	CRF	Lab data, ECG	X	This term should be used for clinical data that were actually observed or recorded by a person or received from an instrument; it should not be used for data that have been interpreted, calculated, or derived from other information.
Derived	X	X	Lab data, ECG	SDTM	Derivation examples include calculations performed during data collection (e.g., --DY). Other derivation examples: calculations within ePRO (e.g., questionnaire section scores) and calculations within EDC (e.g., BMI, BSA).
Assigned	X	X	Adjudicator	SDTM	Examples of this include third-party attributions by an adjudicator, coded terms that are supplied as part of a coding process, and values that are set independently of any subject-related data values in order to complete SDTM fields such as DOMAIN and --TESTCD
Protocol	X	X	X	SDTM	An example would be VSPOS (Vital Signs Position), which could be specified in the protocol and be provided by other means (e.g. CRF, eDT).
Predecessor	X	X	X	X	Use when a value is an exact copy of another value in an SDTM dataset.

Figure 16: Guide for Source-column assignment – Method, Origin, and Source columns (CDISC Define-XML Specification v2.1 pg. 31)

For the “Method” column: we will equate the “Variable” column to the “Method” column for any variable that was computed algorithmically. SDTM variables such as age, sequences, and study days fall into this category since they were computed within the SDTM programs.

For the “Origin” column: we will assign either Collected, Derived, Protocol, or Assigned to the “Origin” column depending on the origination case of the variable. We can use variable-specific cases to increase efficiency by assigning entries based on which variables are missing/non-missing, and indexing columns to check for certain strings and words.

For the “Source” column: we will assign either Investigator, Sponsor, or Vendor to the “Source” column depending on the specified origination case of the variable and who the data was collected by. We can use variable-specific cases to increase efficiency by assigning entries based on the “Origin” column assignment, which variables are missing/non-missing, and indexing columns to check for certain strings and words.

The final resulting dataset and the code for this portion are shown.

Filter and Sort | Query Builder | Where | Data | Describe | Graph | Analyze | Export | Send To

	method	origin	source	codelist	format	variable	pages	Order	Dataset	Label	Data Type	Length
1		Assigned	Sponsor			ACTARM		22	DM	Description of...	text	20
2		Assigned	Sponsor			ACTARMCD		21	DM	Actual Arm Co...	text	8
3		Assigned	Vendor	MedDRA		AEBDSYCD		16	AE	Body System o...	integer	8
4		Assigned	Sponsor			AEBODSYS		15	AE	Body System o...	text	67
5		Assigned	Vendor	MedDRA		AEDECOD		9	AE	Dictionary-Den...	text	200
6		Collected	Investigator			AEDTC	6 21 24 31 35 4...	31	AE	Date/Time of C...	datetime	
7		Derived	Sponsor		ISO 8601	AEENDTC	120 121 122	33	AE	End Date/Time...	datetime	
8	AEENDY	Assigned	Sponsor			AEENDY		35	AE	Study Day of E...	integer	8
9		Assigned	Vendor	MedDRA		AEHLGT		13	AE	High Level Gro...	text	100
10		Assigned	Vendor	MedDRA		AEHLGTCD		14	AE	High Level Gro...	integer	8
11		Assigned	Vendor	MedDRA		AEHLT		11	AE	High Level Term	text	100
12		Assigned	Vendor	MedDRA		AEHLTCD		12	AE	High Level Ter...	integer	8
13		Assigned	Vendor	MedDRA		AELLT		7	AE	Lowest Level T...	text	100
14		Assigned	Vendor	MedDRA		AELLTCD		8	AE	Lowest Level T...	integer	8
15		Assigned	Vendor	MedDRA		AEPTCD		10	AE	Preferred Term...	integer	8
16		Collected	Investigator			AEREL	120 121 122	22	AE	Causality	text	8
17	AESEQ	Assigned	Sponsor			AESEQ		4	AE	Sequence Num...	integer	8
18		Assigned	Vendor	MedDRA		AESOC		17	AE	Primary Syste...	text	100
19		Assigned	Vendor	MedDRA		AESOCCD		18	AE	Primary Syste...	integer	8
20		Collected	Investigator			AESPID	105 120 121 1...	5	AE	Sponsor-Defin...	text	3
21		Derived	Sponsor		ISO 8601	AESTDTC	115 120 121 1...	32	AE	Start Date/Tim...	datetime	
22	AESTDY	Assigned	Sponsor			AESTDY		34	AE	Study Day of S...	integer	8
23		Collected	Investigator			AETERM	120 121 122	6	AE	Reported Term...	text	200
24	AGE	Assigned	Sponsor			AGE		14	DM	Age	integer	8
25		Assigned	Sponsor			ARM		20	DM	Description of...	text	20
26		Assigned	Sponsor			ARMCD		19	DM	Planned Arm C...	text	8
27		Assigned	Sponsor			CMCLAS		9	CM	Medication Cla...	text	42
28		Assigned	Sponsor			CMDECOD		7	CM	Standardized...	text	24
29		Collected	Investigator			CMDOSE	123 124 125	10	CM	Dose per Admi...	float	7
30		Collected	Investigator			CMDTC	6 21 24 31 35 4...	17	CM	Date/Time of C...	datetime	
31		Derived	Sponsor		ISO 8601	CMENDTC	123 124 125	19	CM	End Date/Time...	datetime	
32	CMENDY	Assigned	Sponsor			CMENDY		21	CM	Study Day of E...	integer	8
33		Collected	Investigator			CMINDC	123 124 125	8	CM	Indication	text	34
34	CMSEQ	Assigned	Sponsor			CMSEQ		4	CM	Sequence Num...	integer	8
35		Assigned	Sponsor			CMSPID		5	CM	Sponsor-Defin...	text	2
36		Derived	Sponsor		ISO 8601	CMSTDTC	123 124 125	18	CM	Start Date/Tim...	text	10
37	CMSTDY	Assigned	Sponsor			CMSTDY		20	CM	Study Day of S...	integer	8
38		Collected	Investigator			CMTRT	123 124 125	6	CM	Reported Nam...	text	44
39		Derived	Sponsor		ISO 3166-1 Alp...	COUNTRY		23	DM	Country	text	3
40		Derived	Sponsor		ISO 8601	DMDTC	6	24	DM	Date/Time of C...	datetime	
41	DMDY	Assigned	Sponsor			DMDY		25	DM	Study Day of C...	integer	8
42		Assigned	Sponsor			DOMAIN		2	AE	Domain Abbrev...	text	2
43		Assigned	Sponsor			DOMAIN		2	CM	Domain Abbrev...	text	2
44		Assigned	Sponsor			DOMAIN		2	DM	Domain Abbrev...	text	2
45		Assigned	Sponsor			DOMAIN		2	DS	Domain Abbrev...	text	2
46		Assigned	Sponsor			DOMAIN		2	LB	Domain Abbrev...	text	2

Figure 17: Completion of all column-variable updates and assignments needed for the final output

```
/*Now defining Method, Origin, and Source Variables*/
data merged_define;
      length method1 origin1 source1 $100.;
      set merged_pg_ig_ct1;

/*defining Method column -- any variable that was computed or derived by a
formula across all or within any SDTM(S) by an algorithm*/
      if (index(variable,'SEQ'))ne 0 then method1=variable;
      if (index(variable,'AGE'))ne 0 then method1=variable;
      if (index(variable,'USUBJID'))ne 0 then method1=variable;
      if (index(variable,'DY'))ne 0 then method1=variable;


/*defining Origin column -- describes how the variable originated*/
      if pages ne '' and format='' then origin1='Collected';
      if format ne '' or method ne '' then origin1='Derived';
      if (index(variable,'STUDYID'))ne 0 then origin1='Protocol';
      if (index(codelist,'MedDRA'))ne 0 then origin1='Assigned';
      if origin1='' then origin1='Assigned';

/*defining Source column -- indicates the deciding entity for the assignment
of the origin*/
      if pages ne '' and format='' and origin1='Collected' then
source1='Investigator';
      if format ne '' or method ne '' and origin1='Derived' then
source1='Sponsor';
      if (index(variable,'STUDYID'))ne 0 and origin1='Protocol' then
source1='Sponsor';
      if (index(codelist,'MedDRA'))ne 0 and origin1='Assigned' then
source1='Vendor';
      if origin1='Assigned' and (index(codelist,'MedDRA')) = 0 then
source1='Sponsor';

      drop method origin source;
      rename method1=method origin1=origin source1=source;

run;
```

Step 4 – THE FINALE

PRESERVING THE ORDER AFTER COMPLETION

To arrange the final dataset in the original order (both rows and columns) of the SDTM Define Specifications document, we refer to a couple of items created when we first started making this program. First, to get the dataset in the correct row-order, we will sort the final **UPDATED_VARS** dataset by the "new_ord" variable. Secondly, to get the **UPDATED_VARS** dataset in the correct column-order, we will call the ATTRIB macro. The final dataset name for this program section is called **VARIABLES_TAB**.

The final resulting dataset and the code for this portion are shown.

VARIABLES_TAB

Filter and Sort | Query Builder | Where | Data | Describe | Graph | Analyze | Export | Send To

	new_ord	Order	Dataset	Variable	Label	Data Type	Length	Significant...	Format	Mandatory	Assigned V...	Codelist	Common	Origin	Source	Pages	Method
1	1	1	AE	STUDYID	Study Identifier	text	12			Yes				Protocol	Sponsor	6	STUDYID
2	2	2	AE	DOMAIN	Domain Abbre...	text	2			Yes				Assigned	Sponsor		
3	3	3	AE	USUBJID	Unique Subjec...	text	11			No				Assigned	Sponsor		USUBJID
4	4	4	AE	AESEQ	Sequence Nu...	integer	8			Yes				Assigned	Sponsor		AESEQ
5	5	5	AE	AESPID	Sponsor-Defin...	text	3			No				Collected	Investigator	105 120 121 1...	
6	6	6	AE	AETERM	Reported Term...	text	200			Yes				Collected	Investigator	120 121 122	
7	7	7	AE	AELLT	Lowest Level T...	text	100			No		MedDRA		Assigned	Vendor		
8	8	8	AE	AELLTCD	Lowest Level T...	integer	8			No		MedDRA		Assigned	Vendor		
9	9	9	AE	AEDECOD	Dictionary-Deri...	text	200			Yes		MedDRA		Assigned	Vendor		
10	10	10	AE	AEPTCD	Preferred Term...	integer	8			No		MedDRA		Assigned	Vendor		
11	11	11	AE	AEHLT	High Level Term	text	100			No		MedDRA		Assigned	Vendor		
12	12	12	AE	AEHLTCD	High Level Ter...	integer	8			No		MedDRA		Assigned	Vendor		
13	13	13	AE	AEHLGT	High Level Gro...	text	100			No		MedDRA		Assigned	Vendor		
14	14	14	AE	AEHLGTCD	High Level Gro...	integer	8			No		MedDRA		Assigned	Vendor		
15	15	15	AE	AEBODSYS	Body System o...	text	67			No				Assigned	Sponsor		
16	16	16	AE	AEBDSYCD	Body System o...	integer	8			No		MedDRA		Assigned	Vendor		
17	17	17	AE	AESOC	Primary Syste...	text	100			No		MedDRA		Assigned	Vendor		
18	18	18	AE	AESOCCD	Primary Syste...	integer	8			No		MedDRA		Assigned	Vendor		
19	19	19	AE	AESEV	Severity/Intens...	text	8			No		AESEV		Collected	Investigator	120 121 122	
20	20	20	AE	AESER	Serious Event	text	1			No		NY		Collected	Investigator	120 121 122	
21	21	21	AE	AEACN	Action Taken...	text	30			No		ACN		Assigned	Sponsor		
22	22	22	AE	AEREL	Causality	text	8			No				Collected	Investigator	120 121 122	
23	23	23	AE	AEOUT	Outcome of Ad...	text	200			No		OUT		Assigned	Sponsor		
24	24	24	AE	AESCAN	Involves Cancer	text	1			No		NY		Collected	Investigator	120 121 122	
25	25	25	AE	AESCONG	Congenital An...	text	1			No		NY		Collected	Investigator	120 121 122	
26	26	26	AE	AESDISAB	Persist or Signi...	text	1			No		NY		Collected	Investigator	120 121 122	
27	27	27	AE	AESDTH	Results in Death	text	1			No		NY		Collected	Investigator	120 121 122	
28	28	28	AE	AESHOSP	Requires or Pr...	text	1			No		NY		Collected	Investigator	120 121 122	
29	29	29	AE	AESLIFE	Is Life Threate...	text	1			No		NY		Collected	Investigator	120 121 122	
30	30	30	AE	AESOD	Occurred with...	text	1			No		NY		Collected	Investigator	120 121 122	
31	31	31	AE	AEDTC	Date/Time of C...	datetime				No				Collected	Investigator	6 21 24 31 35...	
32	32	32	AE	AESTDTC	Start Date/Tim...	datetime			ISO 8601	No				Derived	Sponsor	115 120 121 1...	
33	33	33	AE	AEENDTC	End Date/Time...	datetime			ISO 8601	No				Derived	Sponsor	120 121 122	
34	34	34	AE	AESTDY	Study Day of S...	integer	8			No				Assigned	Sponsor		AESTDY
35	35	35	AE	AEENDY	Study Day of E...	integer	8			No				Assigned	Sponsor		AEENDY
36	36	1	CM	STUDYID	Study Identifier	text	12			Yes				Protocol	Sponsor	6	STUDYID
37	37	2	CM	DOMAIN	Domain Abbre...	text	2			Yes				Assigned	Sponsor		
38	38	3	CM	USUBJID	Unique Subjec...	text	11			No				Assigned	Sponsor		USUBJID
39	39	4	CM	CMSEQ	Sequence Nu...	integer	8			Yes				Assigned	Sponsor		CMSEQ
40	40	5	CM	CMSPID	Sponsor-Defin...	text	2			No				Assigned	Sponsor		
41	41	6	CM	CMTRT	Reported Nam...	text	44			Yes				Collected	Investigator	123 124 125	
42	42	7	CM	CMDECOD	Standardized...	text	24			No				Assigned	Sponsor		
43	43	8	CM	CMINDC	Indication	text	34			No				Collected	Investigator	123 124 125	
44	44	9	CM	CMCLAS	Medication Cla...	text	42			No				Assigned	Sponsor		
45	45	10	CM	CMDOSE	Dose per Admi...	float	7	3		No				Collected	Investigator	123 124 125	
46	46	11	CM	CMDOSU	Dose Units	text	17			No		UNIT		Collected	Investigator	123 124 125	
47	47	12	CM	CMDOSFRQ	Dosing Freque...	text	15			No		FREQ		Collected	Investigator	123 124 125	
48	48	13	CM	CMROUTE	Route of Admi...	text	200			No		ROUTE		Collected	Investigator	123 124 125	
49	49	14	CM	VISITNUM	Visit Number	integer	8			No				Collected	Investigator	6 21 24 31 35...	
50	50	15	CM	VISIT	Visit Name	text	19			No				Collected	Investigator	6 21 24 31 35...	
51	51	16	CM	VISITDY	Planned Study...	integer	8			No				Assigned	Sponsor		VISITDY

Figure 18: Result after using the ordering variable and calling the ATTRIB macro

```
/*Order by original assigned order of variables from imported file*/
proc sort data=merged_define;
        by new_ord;
run;

/****************************************/
/** FINAL VARIABLES TAB FOR EXCEL FILE **/
/****************************************/

data Variables_Tab;
        %variable_order;  /*keeps the same order of variable column names found in the
define document*/
        set merged_define;
run;
```

EXPORTING THE DATASET

To export the **VARIABLES_TAB** dataset from SAS into Excel, ensure the dataset is open, then click on "Send To" and choose "Microsoft Excel". Afterwards, you should see an excel file pop-up called *Book 1* and an excel sheet called *Sheet 1*. This will be the output that should be copied into the SDTM Define Specifications *Variables* excel sheet. Since SAS Enterprise Guide 7.1 is used for this example, if using a different SAS environment, the export process may slightly differ. The resulting excel file is shown.

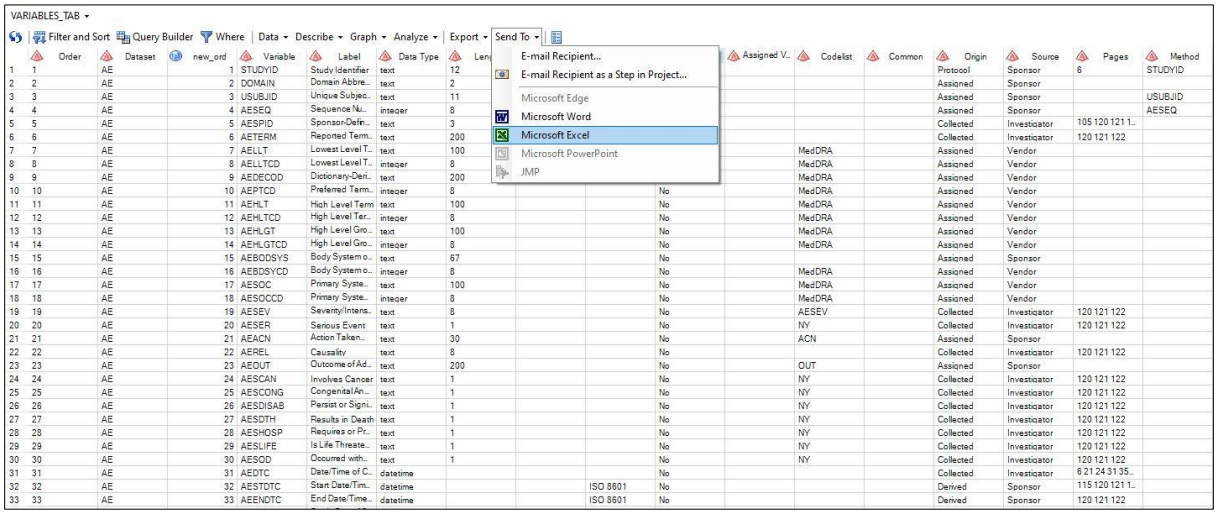
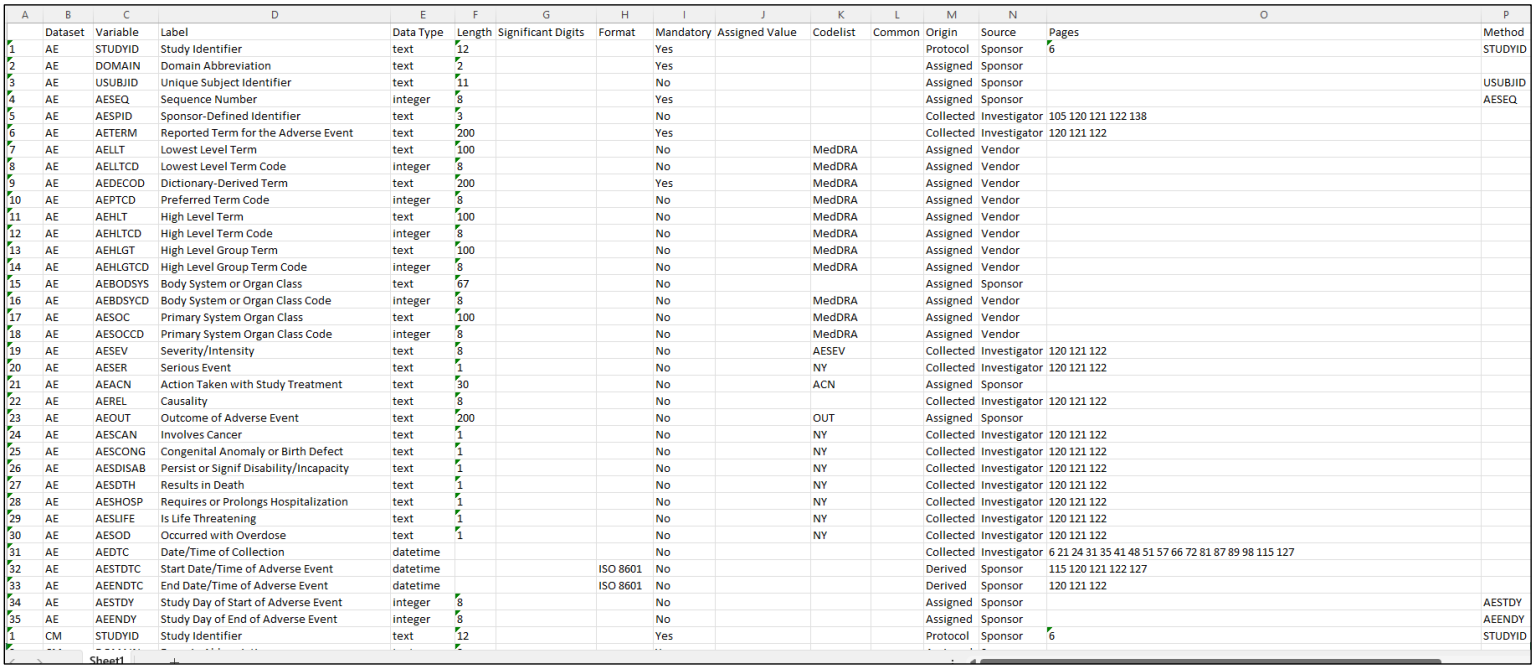


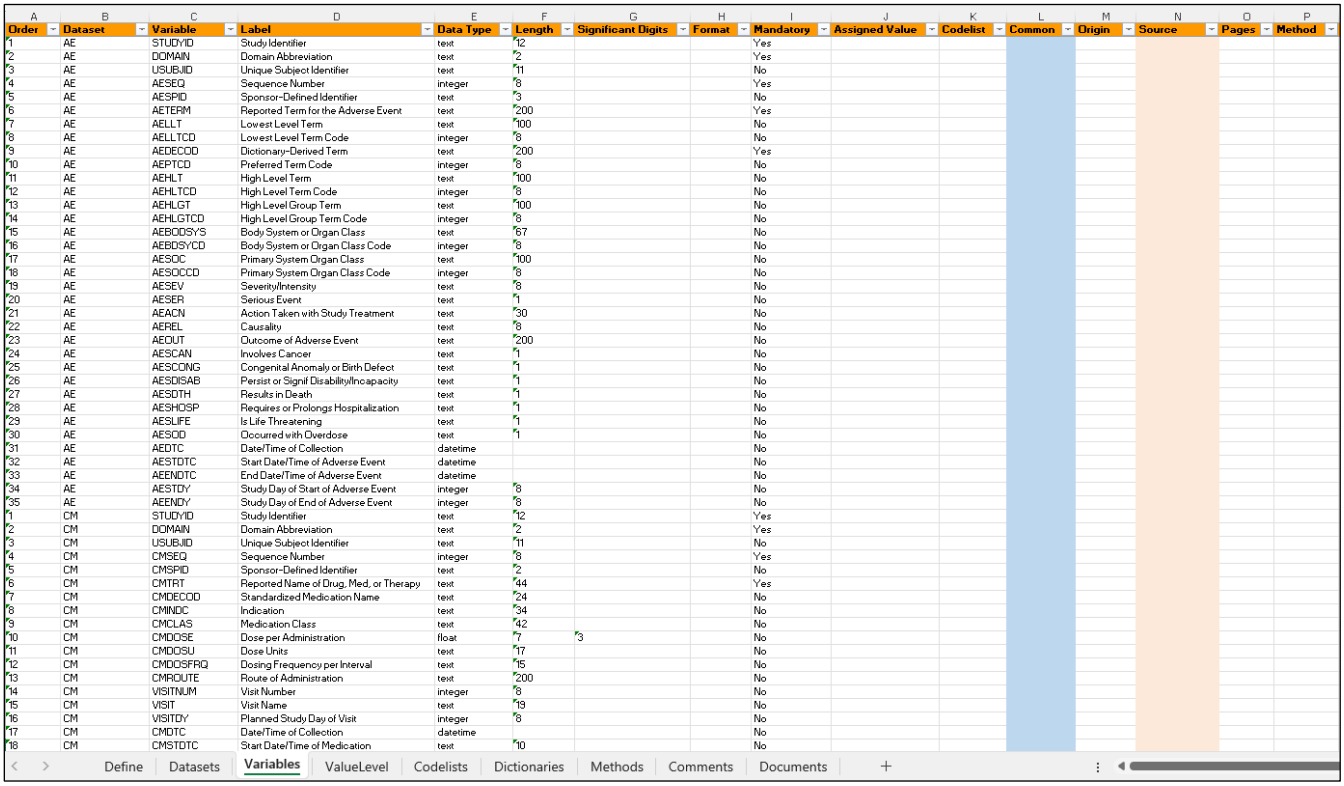
Figure 19: Exporting the dataset into an Excel file



	Dataset	Variable	Label	Data Type	Length	Significant Digits	Format	Mandatory	Assigned Value	Codelist	Common	Origin	Source	Pages	Method
1	AE	STUDYID	Study Identifier	text	12			Yes				Protocol	Sponsor	6	
2	AE	DOMAIN	Domain Abbreviation	text	2			Yes				Assigned	Sponsor		
3	AE	USUBJID	Unique Subject Identifier	text	11			No				Assigned	Sponsor		
4	AE	AESEQ	Sequence Number	integer	8			Yes				Assigned	Sponsor		
5	AE	AESPID	Sponsor-Defined Identifier	text	3			No				Collected	Investigator	105 120 121 122 138	
6	AE	AETERM	Reported Term for the Adverse Event	text	200			Yes				Collected	Investigator	120 121 122	
7	AE	AELLT	Lowest Level Term	text	100			No		MedDRA		Assigned	Vendor		
8	AE	AELLTCD	Lowest Level Term Code	integer	8			No		MedDRA		Assigned	Vendor		
9	AE	AEDECOD	Dictionary-Derived Term	text	200			Yes		MedDRA		Assigned	Vendor		
10	AE	AEPTCD	Preferred Term Code	integer	8			No		MedDRA		Assigned	Vendor		
11	AE	AEHLT	High Level Term	text	100			No		MedDRA		Assigned	Vendor		
12	AE	AEHLTCD	High Level Term Code	integer	8			No		MedDRA		Assigned	Vendor		
13	AE	AEHLGT	High Level Group Term	text	100			No		MedDRA		Assigned	Vendor		
14	AE	AEHLGTCD	High Level Group Term Code	integer	8			No		MedDRA		Assigned	Vendor		
15	AE	AEBODSYS	Body System or Organ Class	text	67			No				Assigned	Sponsor		
16	AE	AEBDSYCD	Body System or Organ Class Code	integer	8			No		MedDRA		Assigned	Vendor		
17	AE	AESOC	Primary System Organ Class	text	100			No		MedDRA		Assigned	Vendor		
18	AE	AESOCCD	Primary System Organ Class Code	integer	8			No		MedDRA		Assigned	Vendor		
19	AE	AESEV	Severity/Intensity	text	8			No		AESEV		Collected	Investigator	120 121 122	
20	AE	AESER	Serious Event	text	1			No		NY		Collected	Investigator	120 121 122	
21	AE	AEACN	Action Taken with Study Treatment	text	30			No		ACN		Assigned	Sponsor		
22	AE	AEREL	Causality	text	8			No				Collected	Investigator	120 121 122	
23	AE	AEOUT	Outcome of Adverse Event	text	200			No		OUT		Assigned	Sponsor		
24	AE	AESCAN	Involves Cancer	text	1			No		NY		Collected	Investigator	120 121 122	
25	AE	AESCONG	Congenital Anomaly or Birth Defect	text	1			No		NY		Collected	Investigator	120 121 122	
26	AE	AESDISAB	Persist or Signif Disability/Incapacity	text	1			No		NY		Collected	Investigator	120 121 122	
27	AE	AESDTH	Results in Death	text	1			No		NY		Collected	Investigator	120 121 122	
28	AE	AESHOSP	Requires or Prolongs Hospitalization	text	1			No		NY		Collected	Investigator	120 121 122	
29	AE	AESLIFE	Is Life Threatening	text	1			No		NY		Collected	Investigator	120 121 122	
30	AE	AESOD	Occurred with Overdose	text	1			No		NY		Collected	Investigator	120 121 122	
31	AE	AEDTC	Date/Time of Collection	datetime				No				Collected	Investigator	6 21 24 31 35 41 48 51 57 66 72 81 87 89 98 115 127	
32	AE	AESTDTC	Start Date/Time of Adverse Event	datetime			ISO 8601	No				Derived	Sponsor	115 120 121 122 127	
33	AE	AEENDTC	End Date/Time of Adverse Event	datetime			ISO 8601	No				Derived	Sponsor	120 121 122	
34	AE	AESTDY	Study Day of Start of Adverse Event	integer	8			No				Assigned	Sponsor		AESTDY
35	AE	AEENDY	Study Day of End of Adverse Event	integer	8			No				Assigned	Sponsor		AEENDY
1	CM	STUDYID	Study Identifier	text	12			Yes				Protocol	Sponsor	6	STUDYID

Figure 20: The SAS dataset exported as an Excel file

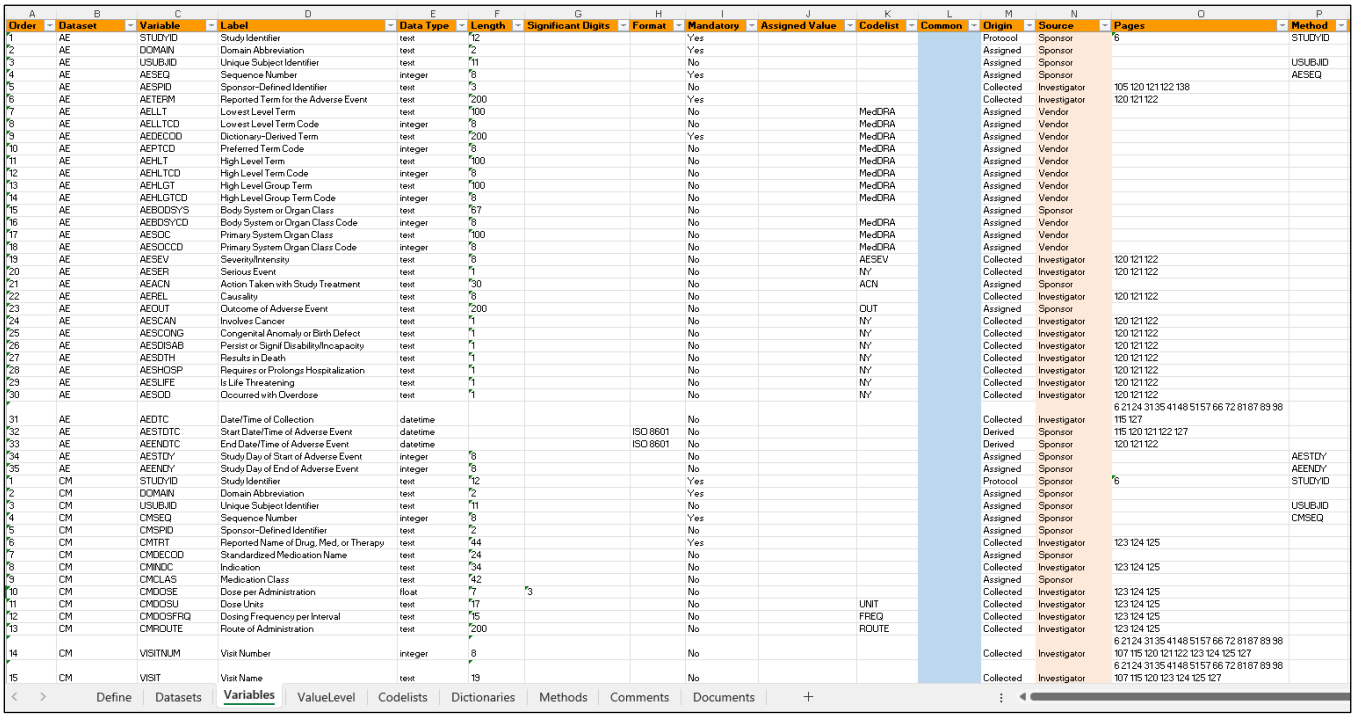
The before and after completion of the SDTM Define Specifications document *Variables* excel sheet is shown.



Order	Dataset	Variable	Label	Data Type	Length	Significant Digits	Format	Mandatory	Assigned Value	Codelist	Common	Origin	Source	Pages	Method
1	AE	STUDYID	Study Identifier	text	12			Yes							
2	AE	DOMAIN	Domain Abbreviation	text	2			Yes							
3	AE	USUBJID	Unique Subject Identifier	text	11			No							
4	AE	AESEQ	Sequence Number	integer	8			Yes							
5	AE	AESPID	Sponsor-Defined Identifier	text	3			No							
6	AE	AETERM	Reported Term for the Adverse Event	text	200			Yes							
7	AE	AELLT	Lowest Level Term	text	100			No							
8	AE	AELLTCD	Lowest Level Term Code	integer	8			No							
9	AE	AEDECOD	Dictionary-Derived Term	text	200			Yes							
10	AE	AEPTCD	Preferred Term Code	integer	8			No							
11	AE	AEHLT	High Level Term	text	100			No							
12	AE	AEHLTCD	High Level Term Code	integer	8			No							
13	AE	AEHLGT	High Level Group Term	text	100			No							
14	AE	AEHLGTCD	High Level Group Term Code	integer	8			No							
15	AE	AEBODSYS	Body System or Organ Class	text	67			No							
16	AE	AEBDSYCD	Body System or Organ Class Code	integer	8			No							
17	AE	AESOC	Primary System Organ Class	text	100			No							
18	AE	AESOCCD	Primary System Organ Class Code	integer	8			No							
19	AE	AESEV	Severity/Intensity	text	8			No							
20	AE	AESER	Serious Event	text	1			No							
21	AE	AEACN	Action Taken with Study Treatment	text	30			No							
22	AE	AEREL	Causality	text	8			No							
23	AE	AEOUT	Outcome of Adverse Event	text	200			No							
24	AE	AESCAN	Involves Cancer	text	1			No							
25	AE	AESCONG	Congenital Anomaly or Birth Defect	text	1			No							
26	AE	AESDISAB	Persist or Signif Disability/Incapacity	text	1			No							
27	AE	AESDTH	Results in Death	text	1			No							
28	AE	AESHOSP	Requires or Prolongs Hospitalization	text	1			No							
29	AE	AESLIFE	Is Life Threatening	text	1			No							
30	AE	AESOD	Occurred with Overdose	text	1			No							
31	AE	AEDTC	Date/Time of Collection	datetime				No							
32	AE	AESTDTC	Start Date/Time of Adverse Event	datetime				No							
33	AE	AEENDTC	End Date/Time of Adverse Event	datetime				No							
34	AE	AESTDY	Study Day of Start of Adverse Event	integer	8			No							
35	AE	AEENDY	Study Day of End of Adverse Event	integer	8			No							
1	CM	STUDYID	Study Identifier	text	12			Yes							
2	CM	DOMAIN	Domain Abbreviation	text	2			Yes							
3	CM	USUBJID	Unique Subject Identifier	text	11			No							
4	CM	CMSEQ	Sequence Number	integer	8			Yes							
5	CM	CMSPID	Sponsor-Defined Identifier	text	2			No							
6	CM	CMTRT	Reported Name of Drug, Med, or Therapy	text	44			Yes							
7	CM	CMDECOD	Standardized Medication Name	text	24			No							
8	CM	CMINDC	Indication	text	34			No							
9	CM	CMCLAS	Medication Class	text	42			No							
10	CM	CMDOSE	Dose per Administration	float	7	3		No							
11	CM	CMDOSU	Dose Units	text	17			No							
12	CM	CMDOSFRQ	Dosing Frequency per Interval	text	15			No							
13	CM	CMROUTE	Route of Administration	text	200			No							
14	CM	VISITNUM	Visit Number	integer	8			No							
15	CM	VISIT	Visit Name	text	19			No							
16	CM	VISITDY	Planned Study Day of Visit	integer	8			No							
17	CM	CMDTC	Date/Time of Collection	datetime				No							
18	CM	CMSTDTC	Start Date/Time of Medication	text	10			No							

Define | Datasets | Variables | ValueLevel | Codelists | Dictionaries | Methods | Comments | Documents

Figure 21: Before the completion of the SDTM Define Specifications document



Order	Dataset	Variable	Label	Data Type	Length	Significant Digits	Format	Mandatory	Assigned Value	Codelist	Common	Origin	Source	Pages	Method
1	AE	STUDYID	Study Identifier	text	12			Yes				Protocol	Sponsor	6	STUDYID
2	AE	DOMAIN	Domain Abbreviation	text	2			Yes				Assigned	Sponsor		
3	AE	USUBJID	Unique Subject Identifier	text	11			No				Assigned	Sponsor		USUBJID
4	AE	AESEQ	Sequence Number	integer	8			Yes				Assigned	Sponsor		AESEQ
5	AE	AESPID	Sponsor-Defined Identifier	text	3			No				Collected	Investigator	105 120 121 122 138	
6	AE	AETERM	Reported Term for the Adverse Event	text	200			Yes				Collected	Investigator	120 121 122	
7	AE	AELLT	Lowest Level Term	text	100			No		MedDRA		Assigned	Vendor		
8	AE	AELLTCD	Lowest Level Term Code	integer	8			No		MedDRA		Assigned	Vendor		
9	AE	AEDECOD	Dictionary-Derived Term	text	200			Yes		MedDRA		Assigned	Vendor		
10	AE	AEPTCD	Preferred Term Code	integer	8			No		MedDRA		Assigned	Vendor		
11	AE	AEHLT	High Level Term	text	100			No		MedDRA		Assigned	Vendor		
12	AE	AEHLTCD	High Level Term Code	integer	8			No		MedDRA		Assigned	Vendor		
13	AE	AEHLGT	High Level Group Term	text	100			No		MedDRA		Assigned	Vendor		
14	AE	AEHLGTCD	High Level Group Term Code	integer	8			No		MedDRA		Assigned	Vendor		
15	AE	AEBODSYS	Body System or Organ Class	text	67			No				Assigned	Sponsor		
16	AE	AEBDSYCD	Body System or Organ Class Code	integer	8			No		MedDRA		Assigned	Vendor		
17	AE	AESOC	Primary System Organ Class	text	100			No		MedDRA		Assigned	Vendor		
18	AE	AESOCCD	Primary System Organ Class Code	integer	8			No		MedDRA		Assigned	Vendor		
19	AE	AESEV	Severity/Intensity	text	8			No		AESEV		Collected	Investigator	120 121 122	
20	AE	AESER	Serious Event	text	1			No		NY		Collected	Investigator	120 121 122	
21	AE	AEACN	Action Taken with Study Treatment	text	30			No		ACN		Assigned	Sponsor		
22	AE	AEREL	Causality	text	8			No				Collected	Investigator	120 121 122	
23	AE	AEOUT	Outcome of Adverse Event	text	200			No		OUT		Assigned	Sponsor		
24	AE	AESCAN	Involves Cancer	text	1			No		NY		Collected	Investigator	120 121 122	
25	AE	AESCONG	Congenital Anomaly or Birth Defect	text	1			No		NY		Collected	Investigator	120 121 122	
26	AE	AESDISAB	Persist or Signif Disability/Incapacity	text	1			No		NY		Collected	Investigator	120 121 122	
27	AE	AESDTH	Results in Death	text	1			No		NY		Collected	Investigator	120 121 122	
28	AE	AESHOSP	Requires or Prolongs Hospitalization	text	1			No		NY		Collected	Investigator	120 121 122	
29	AE	AESLIFE	Is Life Threatening	text	1			No		NY		Collected	Investigator	120 121 122	
30	AE	AESOD	Occurred with Overdose	text	1			No		NY		Collected	Investigator	120 121 122	
31	AE	AEDTC	Date/Time of Collection	datetime				No				Collected	Investigator	6 21 24 31 35 41 48 51 57 66 72 81 87 89 98 115 127	
32	AE	AESTDTC	Start Date/Time of Adverse Event	datetime			ISO 8601	No				Derived	Sponsor	115 120 121 122 127	
33	AE	AEENDTC	End Date/Time of Adverse Event	datetime			ISO 8601	No				Derived	Sponsor	120 121 122	
34	AE	AESTDY	Study Day of Start of Adverse Event	integer	8			No				Assigned	Sponsor		AESTDY
35	AE	AEENDY	Study Day of End of Adverse Event	integer	8			No				Assigned	Sponsor		AEENDY
1	CM	STUDYID	Study Identifier	text	12			Yes				Protocol	Sponsor	6	STUDYID
2	CM	DOMAIN	Domain Abbreviation	text	2			Yes				Assigned	Sponsor		
3	CM	USUBJID	Unique Subject Identifier	text	11			No				Assigned	Sponsor		USUBJID
4	CM	CMSEQ	Sequence Number	integer	8			Yes				Assigned	Sponsor		CMSEQ
5	CM	CMSPID	Sponsor-Defined Identifier	text	2			No				Assigned	Sponsor		
6	CM	CMTRT	Reported Name of Drug, Med, or Therapy	text	44			Yes				Collected	Investigator	123 124 125	
7	CM	CMDECOD	Standardized Medication Name	text	24			No				Assigned	Sponsor		
8	CM	CMINDC	Indication	text	34			No				Collected	Investigator	123 124 125	
9	CM	CMCLAS	Medication Class	text	42			No				Assigned	Sponsor		
10	CM	CMDOSE	Dose per Administration	float	7	3		No				Collected	Investigator	123 124 125	
11	CM	CMDOSU	Dose Units	text	17			No		UNIT		Collected	Investigator	123 124 125	
12	CM	CMDOSFRQ	Dosing Frequency per Interval	text	15			No		FREQ		Collected	Investigator	123 124 125	
13	CM	CMROUTE	Route of Administration	text	200			No		ROUTE		Collected	Investigator	123 124 125	
14	CM	VISITNUM	Visit Number	integer	8			No				Collected	Investigator	6 21 24 31 35 41 48 51 57 66 72 81 87 89 98 107 115 120 121 122 123 124 125 127	
15	CM	VISIT	Visit Name	text	19			No				Collected	Investigator	6 21 24 31 35 41 48 51 57 66 72 81 87 89 98 107 115 120 123 124 125 127	

Define | Datasets | Variables | ValueLevel | Codelists | Dictionaries | Methods | Comments | Documents

Figure 22: After the completion of the SDTM Define Specifications document

CONCLUSION

Completing SDTM and ADAM Define Specifications is not as tedious as it may seem. SAS and Excel are very useful tools that can help in many ways to expedite and shorten task times. While this paper was only focused on the completion of the *Variables* excel sheet in the SDTM Define Specifications document, the discussion and presentation in this paper is translatable to all other instances of Define Specification creation. Such instances are inclusive of all other tabs located in the SDTM Define Specifications document as well as all tabs located in the ADAM Define Specifications document. Following the steps outlined, one can apply the same principles to create a complete program that outputs datasets for all tabs in the SDTM Define Specifications document. One can then copy the corresponding columns from the output excel sheets into the Specifications document.

While any program created for completing Define Specifications is a useful and time-saving tool, it is always important to look over your outputs to ensure that correct rules and procedures are followed. If for any reason there is an issue that is discovered after the creation of the SAS program or after completing the Define Specification document, changes can either be made directly in the Define Specifications excel file, or in the Define Specifications SAS program. After a SAS code change/update in the program, another final SAS dataset and excel output sheet should be created. A well-organized and well-documented general-purpose SDTM and ADAM program made for Define Specifications creation and completion can be used for future clinical trial study specification documents, however, one should generally expect to modify variables and conditions in the program to suite the study. Nonetheless, the overall functional aspect of the SAS program should still be applicable.

APPENDIX: FULL SAS CODE PROGRAM

```
/*************************************************************************/
/**** SDTM Define Specifications - SAS Code for "Variables" Tab Completion ****/
/****                Written By: Star Nze for SESUG 2023
****/
/*************************************************************************/

/*Importing Variables Tab*/
proc import datafile="\\source_path\source_folder\cdiscpilot01 study\SDTM Define
Specifications.xlsx"
       dbms=xlsx
       out=work.variables
       replace;
       sheet=Variables;
       getnames=yes;
run;


/*Importing SDTMCT*/
proc import datafile="\\source_path\source_folder\cdiscpilot01
study\sdtmct_20220624.xlsx"
       dbms=xlsx
       out=work.sdtmct
       replace;
       sheet=Terminology;
       getnames=yes;
run;


/*Importing SDTMIG*/
proc import datafile="\\source_path\source_folder\cdiscpilot01 study\SDTMIG_v3.3.xlsx"
       dbms=xlsx
       out=work.sdtmig
       replace;
       sheet=sdtmigv3_3;
       getnames=yes;
run;


/*Importing aCRF Page Numbers*/
proc import datafile="\\source_path\source_folder\cdiscpilot01
study\sdtmct_20220624.xlsx"
       dbms=xlsx
       out=work.pages
       replace;
       sheet=Pages;
       getnames=no;

run;

/*START OF VARIABLES PROGRAM*/

/*Specifying the order of the column names in the Variables Tab found in the SDTM
Specifications Excel File*/
%macro variable_order;
attrib
Order label='var1'
Dataset label='var2'
Variable label='var3'
Label label='var4'
'Data Type'n label='var5'
```

```
Length label='var6'
'Significant Digits'n label='var7'
Format label='var8'
Mandatory label='var9'
'Assigned Value'n label='var10'
Codelist label='var11'
Common label='var12'
Origin label='var13'
Source label='var14'
Pages label='var15'
Method label='var16'
Predecessor label='var17'
Role label='var18'
'Has No Data'n label='var19'
Comment label='var20'
'Developer Notes'n label='var21'
;
%mend variable_order;

/*Preserving the inital order of the rows from the Excel File by creating an ascending
order variable for up to N total rows*/
data variables1;
      set variables;

      new_ord=_n_;
run;


/*Formatting pages dataset to extract the variable names and the associated page
numbers*/
data pages1;
      set pages(rename=(A=var1));

       if findw(var1,'Contents') ne 0;
       if findw(var1,'Not Entered In Database') = 0;
       var2=var1;

       vars=strip(scan(scan(substr(var2, index(var2,
'Contents')),2,'(''þÿ'),1,')''']''when''='));

       vars=tranwrd(vars,'\r','');

       pgs=input(strip(scan(scan(substr(var2, index(var2, 'Page')),1,'/'),2,'
')),10.);

run;


proc sort nodupkey data=pages1;
      by vars pgs ;
run;


/*subsetting for variables having -- prefixes*/;
data pages1_1;
      set pages1;

      where substr(vars,1,2)="--";

      varsx=scan(vars,2,'[');

      keep pgs varsx;
run;
```

```
/*subsetting for all other variables without -- prefix*/;
data pages2_1;
	set pages1;

	where substr(vars,1,2) ne "--";

	keep pgs vars;
run;

/*macro to iterate through each list item and create a new column and stack as new
dataset*/
%macro scanvar;

%do i=1 %to 8;

data pages1x_&i;
	set pages1_1;
	vars=scan(varsx,&i,',');output;
	drop varsx;

	%end;

run;

data pages1_2;
	set pages1x_:;
	where vars ne '';
run;

proc sort data=pages1_2;
	by pgs vars;
run;

%mend scanvar;

%scanvar;


/*more page data manipulation*/
data pages3;
	set pages1_2 pages2_1;

	vars=strip(vars);

	if vars='M H S T D T C' then vars='MHSTDTC';

run;

proc sort data=pages3;
	by vars pgs;
run;

/*transposing dataset to stack page numbers horizontally*/

proc transpose data=pages3 out=pages3_1;
	by vars;
	var pgs;
run;

/*macro to combine all page numbers per variable delimited by a space*/
%macro stackpgs;
data pages3_2;
	set pages3_1;
```

```
        pgs=strip(col1)||' '||
        %do i=2 %to 40-1;
        strip(col&i)||' '||
        %end;
        strip(col40);

run;
%mend stackpgs;

%stackpgs;

/*final pages result*/
data pages3_3;
        length pages1 variable $150.;
        set pages3_2;

        variable=upcase(vars);
        pages1=tranwrd(pgs,'.','');

        keep pages1 variable;

run;

proc sort data=pages3_3;
        by variable;
run;

data variables1;
        length variable $150.;
        set variables1;

        if dataset='SUPPQS' and variable='QVAL' then
variable=strip(dataset)||'.'||strip(variable);
run;

proc sort data=variables1;
        by variable;
run;


/*adding the pages to the imported Variables dataset*/
data pages_final;
        length pages1 variable $150.;
        merge variables1(in=a) pages3_3;
        by variable;
        if a;

        drop pages;
        rename pages1=pages;
run;

proc sort data=pages_final;
        by variable;
run;

/*renaming a variable to merge*/
data sdtmig1;
        set sdtmig;

        variable=strip('Variable Name'n);
        codelistx=scan(strip('CDISC CT Codelist Code(s)'n),1,';');
        formatx=strip('Described Value Domain(s)'n);
```

```
       where 'CDISC CT Codelist Code(s)'n ne '' or 'Described Value Domain(s)'n ne '';

       keep variable codelistx formatx;
run;

proc sort data=sdtmig1 nodupkey;
       by  variable codelistx formatx;
run;

/*merging sdtmig data onto resulting new Variable dataset with included pages column*/
data merged_pg_ig;
       merge pages_final(in=a) sdtmig1;
       by variable;
       if a;
run;

proc sort data=merged_pg_ig;
       by codelistx;
run;

/*assigning variables for data manipulation*/
data sdtmct1;
       set sdtmct;

       codelistx=strip(code);
       codelist_name=strip('CDISC Submission Value'n);

       keep codelistx codelist_name;
run;

proc sort data=sdtmct1 nodupkey;
       by codelistx;
run;

/*merging sdtmct data onto resulting new current Variable dataset with added IG data*/
data merged_pg_ig_ct;
       length variable $150.;
       merge merged_pg_ig(in=a) sdtmct1;
       by codelistx;
       if a;
run;

/*completion of adding Format and Codelist Variables*/
data merged_pg_ig_ct1;
       set merged_pg_ig_ct;

       codelist_name1=codelist_name;
       formatx1=formatx;

       if formatx='MedDRA' then codelist_name1=formatx;
       if formatx='MedDRA' then formatx1='';

       drop codelist format codelist_name formatx codelistx;
       rename codelist_name1=codelist formatx1=format;
run;



/*Now defining Method, Origin, and Source Variables*/
data merged_define;
       length method1 origin1 source1 $100.;
       set merged_pg_ig_ct1;
```

```
/*defining Method column -- any variable that was computed or derived by a formula
across all or within any SDTM(S) by an algorithm*/
      if (index(variable,'SEQ'))ne 0 then method1=variable;
      if (index(variable,'AGE'))ne 0 then method1=variable;
      if (index(variable,'USUBJID'))ne 0 then method1=variable;
      if (index(variable,'DY'))ne 0 then method1=variable;


/*defining Origin column -- describes how the variable originated*/
      if pages ne '' and format='' then origin1='Collected';
      if format ne '' or method ne '' then origin1='Derived';
      if (index(variable,'STUDYID'))ne 0 then origin1='Protocol';
      if (index(codelist,'MedDRA'))ne 0 then origin1='Assigned';
      if origin1='' then origin1='Assigned';

/*defining Source column -- indicates the deciding entity for the assignment of the
origin*/
      if pages ne '' and format='' and origin1='Collected' then
source1='Investigator';
      if format ne '' or method ne '' and origin1='Derived' then source1='Sponsor';
      if (index(variable,'STUDYID'))ne 0 and origin1='Protocol' then
source1='Sponsor';
      if (index(codelist,'MedDRA'))ne 0 and origin1='Assigned' then source1='Vendor';
      if origin1='Assigned' and (index(codelist,'MedDRA')) = 0 then
source1='Sponsor';

      drop method origin source;
      rename method1=method origin1=origin source1=source;

run;


/*Order by original assigned order of variables from imported file*/
proc sort data=merged_define;
      by new_ord;
run;

/***************************************/
/** FINAL VARIABLES TAB FOR EXCEL FILE **/
/***************************************/

data Variables_Tab;
      %variable_order;  /*keeps the same order of variable column names found in the
define document*/
      set merged_define;
run;
```

REFERENCES

1. SDTM Implementation Guide (SDTM-IG v3.3)
   https://www.cdisc.org/system/files/members/standard/foundational/SDTMIG_v3.3_FINAL.pdf

2. CDISC SDTM/CDASH Controlled Terminology (SDTM-CT 2022-06-24)
   https://evs.nci.nih.gov/ftp1/CDISC/SDTM/Archive/CDASH%20Terminology%202022-06-24.pdf

3. Define-XML Specification (Define-XML v2.1)
   https://www.cdisc.org/standards/data-exchange/define-xml/define-xml-v2-1

4. CDISC Pilot Project GitHub Page
   https://github.com/cdisc-org/sdtm-adam-pilot-project

5. CDISC Pilot SDTM/ADAM Pilot Project Final CSR with Synopsis of Study
   https://github.com/cdisc-org/sdtm-adam-pilot-project/blob/master/updated-pilot-submission-package/900172/m5/53-clin-stud-rep/535-rep-effic-safety-stud/5351-stud-rep-contr/cdiscpilot01/cdiscpilot01.pdf

6. CDISC Website
   https://www.cdisc.org

7. CDISC Library Browser
   https://library.cdisc.org/browser

CONTACT INFORMATION

For any comments, questions, and/or suggestions, contact the author at:

Star Nze, M.S.
Statistical Programmer
Email: Sn4512@gmail.com
LinkedIn: www.linkedin.com/in/starn
Website: https://sn4512.wixsite.com/starn