PharmaSUG2016 - Paper SS05

A Practical Approach to Re-sizing Character Variable Lengths for FDA Submission Datasets (both SDTM and ADaM)

Xiangchen (Bob) Cui, Alkermes, Inc, Waltham, MA

Min Chen, Alkermes, Inc, Waltham, MA

ABSTRACT

FDA issued Study Data Technical Conformance Guide [1] in October 2015, which stipulates "The allotted length for each column containing character (text) data should be set to the maximum length of the variable used across all datasets in the study". FDA/PhUSE 'Data Sizing Best Practices Recommendation' suggests optimizing the size of dataset through managing character variable length to save wasted space. OpenCDISC has built the checks for the compliance.

Re-sizing character variable length from the pre-determined in SDTM and ADaM to the maximum length of the variable on the actual data values is the common solution to be compliant with FDA rule.

Some sponsors resize character variable length at post database lock for FDA submission to solve the challenge of identifying or predicting up front the longest potential value for each character variable, to avoid risk of truncation. However keeping the resized variable length in metadata (define,xml), and keeping the same length of each variable inherited from SDTM domains in ADaM datasets both in datasets and their definex.ml are the most difficult in operational perspective.

This paper presents a SAS-based macro approach which automates the resizing of character variables in both SDTM and ADaM datasets, simultaneously updating the resized variable lengths in define.xml, and keeping their length in SDTM and ADaM datasets same as ones in define.xml. The strategy for handling each individual study and ISS/ISE is also proposed to share the vision to achieve technical accuracy and operational efficiency.

INTRODUCTION

The *Study Data Technical Conformance Guide* (the *Guide*) [1] that the FDA published in October 2015 describes the submission requirements that become effective with Prescription Drug User Fee Act (PDUFA) VI (PDUFA V, which is currently in effect, expires in September 2017). It stipulates "**The allotted length for each column containing character (text) data should be set to the maximum length of the variable used across all datasets in the study**" (Guide, p.7). Hence for the compliance to this rule, "column size reduction" should be applied to the submitted ADaM datasets, as well as the submitted SDTM datasets.

OpenCDISC has built the checks for the SDTM compliance per the request from the agency. However no checks have been built into ADaM Dataset Validator yet so far. Should we comply with the rule (column size reduction) for ADaM FDA submission? If not, the length of each variable inherited from SDTM domains in ADaM datasets should be kept the same as one in SDTM domain, due to the reduction of them in SDTM domains. It seems very "unclear" to the industry at the moment. It is probably due to the fact that the time has not come yet until September 2017.

Some sponsors have developed SAS macros to re-size character variable length at post database lock for FDA submission [3][4]. FDA/PhUSE 'Data Sizing Best Practices Recommendation' discusses optimizing the size of character variable length to avoid wasted space in data sets [2]. However they focus only on SDTM datasets, not on ADaM datasets, even though these could be applied to ADaM datasets. However no papers have discussed the reasonable solution to keeping the resized variable length in metadata (define,xml) for SDTM datasets, not speak of ADaM datasets. FDA/PhUSE recommends that "**Sponsors must reflect in the define documentation the actual variable and value level lengths in the submitted files, or in other words, the reduced lengths**". Hence keeping the length of each resized variable in SDTM domain the same as one in its define.xml is the most critical for FDA

submission. Furthermore, this principle should be applied to ADaM FDA submission, i.e., the length of each resized variable in SDTM domains inherited into ADaM datasets should be kept the same in ADaM metadata (define,xml). If we apply “column size reduction” to ADaM, it is a warranty to keep the lengths in datasets and in define.xml the same. Hence the preparation of define files for both SDTM and ADaM for FDA submission is one of the biggest challenges facing the industry in operational perspective now.

This paper presents a SAS-based macro approach which automates the resizing of each character variable length in both SDTM and ADaM datasets, simultaneously updating each resized variable length in define.xml, and keeping the length in ADaM datasets and their define.xml the same regardless of compliance with the FDA new rule, i.e., “column size reduction”. The strategy for handling both each individual study and ISS/ISE is also discussed.

INTRODUCTION OF OPENCDISC VALIDATOR REPORT FOR NON-COMPLIANCE

Uploading SDTM datasets and their define.xml into OpenCDISC Validator is one of critical steps for finalization of SDTM programming either for database lock or for FDA submission. OpenCDISC can identify the non-compliance to this rule, “column size reduction” for FDA submission. Display 1 provides an example of the **Error Message from OpenCDISC Validator Report** for variables from AE Domain due to the variables lengths exceeding the maximum length of actual data. The OpenCDISC rule and FDA’s rule are reported, along with the variable names and their extra lengths. Note: In **Values Column** the variable is to be reduced to the proposed number as the variable length.

OpenCDISC Validator Report

Issue Summary

Source	OpenCDISC ID	Publisher ID	Message	Severity	Found
AE					
	SD1082	FDAC036	Variable length is too long for actual data	Error	18

OpenCDISC ID	Publisher ID	Message	Description	Category	Severity
SD1082	FDAC036	Variable length is too long for actual data	Variable length should be assigned based on actual stored data to minimize file size. Datasets should be re-sized to the maximum length of actual data used prior to splitting.	Metadata	Error

Domain	Record	Count	Variables	Values	OpenCDISC ID	Publisher ID	Message	Category	Severity
AE			Variable, Excess	AEOUT, 8	SD1082	FDAC036	Variable length is too long for actual data	Metadata	Error
AE			Variable, Excess	AEACNOTH, 153	SD1082	FDAC036	Variable length is too long for actual data	Metadata	Error
AE			Variable, Excess	AEENDTC, 10	SD1082	FDAC036	Variable length is too long for actual data	Metadata	Error
AE			Variable, Excess	AEENRF, 13	SD1082	FDAC036	Variable length is too long for actual data	Metadata	Error
AE			Variable, Excess	AESTDTC, 10	SD1082	FDAC036	Variable length is too long for actual data	Metadata	Error
AE			Variable, Excess	AETERM, 135	SD1082	FDAC036	Variable length is too long for actual data	Metadata	Error
AE			Variable, Excess	AEHLT, 36	SD1082	FDAC036	Variable length is too long for actual data	Metadata	Error
AE			Variable, Excess	AESOC, 33	SD1082	FDAC036	Variable length is too long for actual data	Metadata	Error
AE			Variable, Excess	STUDYID, 9	SD1082	FDAC036	Variable length is too long for actual data	Metadata	Error
AE			Variable, Excess	AEREL, 18	SD1082	FDAC036	Variable length is too long for actual data	Metadata	Error

Display 1. An Example of Detailed Error Message from OpenCDISC Validator Report for AE Domain due to Variables Lengths Exceeding the Maximum Length of Actual Data

Display 2 shows an example of the define.xml for SDTM AE domain after resizing SDTM datasets. The variable lengths of AE were reduced.

Adverse Events (AE) [Location: AE.xpt]

Variable	Label	Key	Type	Length	Controlled Terms or Format	Origin	Derivation/Comment
STUDYID	Study Identifier	1	text	12		CRF Page 1	
DOMAIN	Domain Abbreviation		text	2	["AE" = "Adverse Events"] <Domain Abbreviation (AE)>	Assigned	AE
USUBJID	Unique Subject Identifier	2	text	23		Derived	TRNO+ '-' + PatientNo
AESOC	Primary System Organ Class		text	52	Adverse Event Dictionary	CRF Page 9	MedDRA dictionary assigned. Will be the same as AEBODSYS if the primary SOC was used for analysis.
AESOCCD	Primary System Organ Class Code		integer	8	Adverse Event Dictionary	CRF Page 9	MedDRA dictionary assigned. Will be the same as AEBDSYCD if the primary SOC was used for analysis.
AESEV	Severity/Intensity		text	8	["MILD" = "1; Grade 1", "MODERATE" = "2; Grade 2", "SEVERE" = "3; Grade 3"] <Severity/Intensity Scale for Adverse Events>	CRF Page 8	
AESER	Serious Event		text	1	["N" = "No", "Y" = "Yes"] <No Yes Response>	CRF Page 8	
AEACN	Action Taken with Study Treatment		text	16	["DOSE NOT CHANGED", "DRUG INTERRUPTED", "DRUG WITHDRAWN", "NOT APPLICABLE"] <Action Taken with Study Treatment>	CRF Page 8	

Display 2. A Snapshot of Define.xml Version 2.0 for SDTM AE Domain after the Resizing

INTRODUCTION OF RE-SIZING SDTM DATASETS AND ITS DEFINE.XML

Paper [3], [4], and [5] introduce SAS macros to reset the character variable length to the maximum length of the variable on the actual data values within a dataset. The SAS code inside the macro %trim1 [5] were modified to develop our macro to automate the resizing of SDTM datasets and the updating of the resized variable length in SDTM metadata for automatically generating define.xml. The updated metadata of SDTM dataset will be used to automate the updating of the lengths of the variables inherited from SDTM domains in ADaM datasets for its define.xml, which will be introduced in the following section.

Introduction of SDTM Define.xml Auto Generation

Paper [6] introduces our in-house built SAS-based macro tool to automate generation of define.xml V2.0 from SDTM specification. The specification in SAS data format converted from AE domain specification in MS spreadsheet, named as **SDTMSPEC.ALL_VARS**, is shown in Display 3, where **SDTMSPEC** is the standard ***libname*** for a subfolder of a study folder to store SDTM metadata of SDTM domains. The macro **%get_sdtm_define_xml** reads **SDTMSPEC.ALL_VARS and** automates generation of define.xml V2.0.

DOMAIN	VARIABLE	LABEL	TYPE	LENGTH	ORIGIN	CODELIST	CORE	COMMENT
AE	STUDYID	Study Identifier	Char	20	CRF Page 1		Req	
AE	DOMAIN	Domain Abbreviation	Char	2	Assigned	DOMAIN	Req	AE
AE	USUBJID	Unique Subject Identifier	Char	40	Derived		Req	TRNO+ '-' + PatientNo
AE	AETERM	Reported Term for the Adverse Event	Char	200	CRF Page 4		Req	
AE	AESEV	Severity/Intensity	Char	8	CRF Page 4	AESEV	Perm	
AE	AESER	Serious Event	Char	1	CRF Page 4	NY	Exp	
AE	AEACN	Action Taken with Study Treatment	Char	40	CRF Page 4	ACN	Exp	
AE	AEACNOTH	Other Action Taken	Char	200	CRF Page 4		Perm	
AE	AEREL	Causality	Char	40	CRF Page 4	REL	Exp	
AE	AEOUT	Outcome of Adverse Event	Char	40	CRF Page 4	OUT	Perm	
AE	AESTDTC	Start Date/Time of Adverse Event	Char	20	CRF Page 4	ISO 8601	Exp	
AE	AEENDTC	End Date/Time of Adverse Event	Char	20	CRF Page 4	ISO 8601	Exp	
SUPPAE	STUDYID	Study Identifier	Char	20	CRF Page 1		Req	
SUPPAE	RDOMAIN	Related Domain Abbreviation	Char	2	Assigned	DOMAIN	Req	
SUPPAE	USUBJID	Unique Subject Identifier	Char	40	Derived		Req	
SUPPAE	IDVAR	Identifying Variable	Char	8	Assigned		Exp	
SUPPAE	IDVARVAL	Identifying Variable Value	Char	8	Assigned		Exp	
SUPPAE	QNAM	Qualifier Variable Name	Char	8	CRF Page 4		Req	
SUPPAE	QLABEL	Qualifier Variable Label	Char	40	Assigned		Req	
SUPPAE	QVAL	Data Value	Char	200	CRF Page 4		Req	
SUPPAE	QORIG	Origin	Char	8	Assigned		Req	QORIG = "CRF" if collected on the CRF. QORIG = "ASSIGNED" for medical dictionary coding terms. QORIG = "EDT" if collected on external labs.
SUPPAE	QEVAL	Evaluator	Char	40	Assigned		Exp	QEVAL = " " if QORIG = "CRF".

Display 3. A Snapshot of AE and SUPPAE Specification

Modifying %TRIM1 by Saving A SAS Dataset Storing Resized Variable Names, and Their Original and New Lengths

In order to keep the same lengths in the define file as ones in the dataset, **SDTMSPEC.ALL_VARS.LENGTH** should be simultaneously updated by the resized variable lengths in the data. The macro %trim1 [5] was modified to output a SAS dataset from each macro call. The new SAS data has four variables: domain name, the character variable names, their original variable lengths, and their variable lengths in the resized new dataset. For non-zero observation datasets, all character variable lengths are from the resized datasets. Display 4 provides an example of the new SAS dataset saved from the macro call %TRIM1 for SDTM AE Domain. Note: The new lengths of Variable: DOMAIN, AESEV, and AESER are same as their original ones in this example.

Domain Name	Variable Name	Original Length	New Length
AE	STUDYID	20	11
AE	DOMAIN	2	2
AE	USUBJID	40	22
AE	AESPID	8	2
AE	AETERM	200	65
AE	AELLT	100	65
AE	AEDECOD	100	39
AE	AEHLT	100	64
AE	AEHLGT	100	64
AE	AEBODSYS	100	67
AE	AESOC	100	67
AE	AESEV	8	8
AE	AESER	1	1
AE	AEACN	40	16
AE	AEACNOTH	200	57
AE	AEREL	40	22
AE	AEOUT	40	32
AE	AESTDTC	20	10
AE	AEENDTC	20	10
AE	EPOCH	40	7
AE	AEENRF	20	7

Display 4. An Example of a New SAS Dataset saved from the macro call %TRIM1 for SDTM AE Domain

The SAS codes for saving the new SAS data are as follows.

```
data _varlen;if 0;run;

data _varlen;
     length domain $16. variable $8.;
     domain=%upcase("&indsn");
       %do i=1 %to &nvars;
           %help(&i);
            %if &type=C %then %do;
               variable="&&varname&i";
                   new_length=&&varlen&i;
                 output;
            %end;
       %end;
run;
```

For zero observation datasets, all variable lengths are set to 1. For example, IE domain would have zero observation. Display 5 provides an example of new SAS dataset saved from the macro call %TRIM1.

Variable Name	Original Length	New Length
STUDYID	20	1
DOMAIN	2	1
USUBJID	40	1
IETESTCD	8	1
IETEST	200	1
IECAT	40	1
IEORRES	1	1
IESTRESC	1	1
VISIT	40	1
IEDTC	20	1

Display 5. An Example of a New SAS Dataset saved from the macro call %TRIM1 for SDTM IE Domain

Its SAS codes for saving the new SAS data are as follows.

```
data _varlen;if 0;run;
data _varlen;
```

```
      length domain $16. variable $8.;
        domain=%upcase("&indsn");
        domain=substr(domain,6);
        %do i=1 %to &nvars;
            %help(&i);
             %if &type=C %then %do;
                variable="&&varname&i";
                    new_length=1;
                  output;
             %end;
        %end;
run;
```

After the study database is locked, all SDTM programming (both production and validation) is rerun. The final SDTM datasets are saved in the standard study folder: ..\Study123\final\data\sdtm\pd. The left side of Display 6 shows an example of final SDTM datasets before resizing. These un-resized SDTM datasets are used for ADaM Programming. They are resized by calling a SAS-based macro right after the un-resized SDTM datasets are available.

Introduction of the Macro for Resizing SDTM Datasets

Figure 1 shows the process flow of the macro **%resize_sdtm**

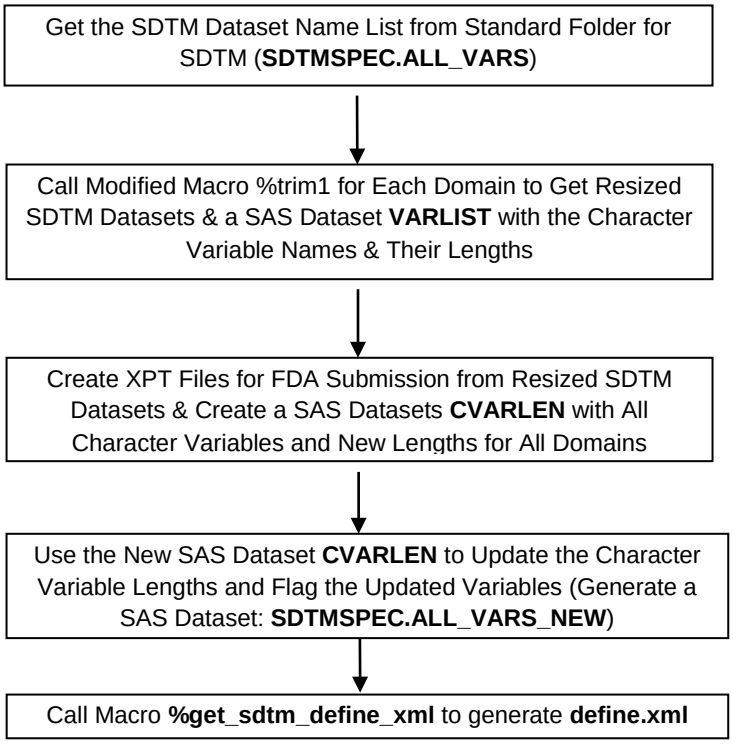

Figure 1. Overview of Process Flow of Macro %resize_sdtm

The calling of **%resize_sdtm** generated resized SDTM datasets in XPT format for FDA submission readiness in the standard SDTM esub subfolder of the study. The right side of Display 6 shows an example of resized SDTM datasets and their define.xml in esub folder, along with SDTM Study Data Reviewer's Guide (SDRG). The reduction of each data file size can be easily seen in Display 6.

These resized SDTM datasets are only for FDA submission, not for any internal ADaM programming. The same principle is also applied to resizing ADaM datasets in the following sections.

The calling of **%resize_sdtm** also generated a new SAS dataset, called **SDTMSPEC.ALL_VARS_NEW** which will be used to automate the generation of SDTM define.xml inside **%resize_sdtm**. The variable lengths in **SDTMSPEC.ALL_VARS_NEW** are updated by the lengths of the resized variables. The original length and flag for the Updated Variables are added into **SDTMSPEC.ALL_VARS_NEW.** Display 7 shows the snapshot of **SDTMSPEC.ALL_VARS_NEW** for the resized AE and SUPPAE with the updated length (LENGTH), original length (OLD_LEN) and flag (UPDATEDFL) for the updated variables.

SDTMSPEC.ALL_VARS_NEW will be used to update the length of each resized variable inherited from SDTM domains in ADaM datasets for keeping the same length in ADaM define.xml by a macro call **%resize_adam**.

Name	Type	Size
ae	SAS Data Set	1,836 KB
cm	SAS Data Set	2,808 KB
da	SAS Data Set	2,752 KB
dm	SAS Data Set	408 KB
ds	SAS Data Set	704 KB
dv	SAS Data Set	128 KB
eg	SAS Data Set	18,648 KB
ex	SAS Data Set	320 KB
ie	SAS Data Set	128 KB
lb	SAS Data Set	153,720 KB
mh	SAS Data Set	1,088 KB
pc	SAS Data Set	6,052 KB
qs	SAS Data Set	284,184 KB
se	SAS Data Set	576 KB
su	SAS Data Set	192 KB
suppae	SAS Data Set	1,920 KB
suppcm	SAS Data Set	3,200 KB
suppda	SAS Data Set	4,608 KB
suppdm	SAS Data Set	1,472 KB
suppds	SAS Data Set	448 KB
suppeg	SAS Data Set	320 KB
supplb	SAS Data Set	10,816 KB
suppqs	SAS Data Set	7,552 KB
suppsu	SAS Data Set	192 KB
suppsv	SAS Data Set	2,368 KB
sv	SAS Data Set	2,432 KB
ta	SAS Data Set	128 KB
te	SAS Data Set	128 KB
ti	SAS Data Set	128 KB
ts	SAS Data Set	128 KB
tv	SAS Data Set	128 KB
vs	SAS Data Set	15,168 KB

Name	Size	Type
Project123-SDRG	354 KB	Microsoft Word 97 - 2003 Document
AE	859 KB	SAS Xport Transport File
CM	1,229 KB	SAS Xport Transport File
DA	1,047 KB	SAS Xport Transport File
DM	134 KB	SAS Xport Transport File
DS	384 KB	SAS Xport Transport File
DV	8 KB	SAS Xport Transport File
EG	4,622 KB	SAS Xport Transport File
EX	69 KB	SAS Xport Transport File
IE	7 KB	SAS Xport Transport File
LB	60,611 KB	SAS Xport Transport File
MH	473 KB	SAS Xport Transport File
PC	2,349 KB	SAS Xport Transport File
QS	205,221 KB	SAS Xport Transport File
SE	204 KB	SAS Xport Transport File
SU	40 KB	SAS Xport Transport File
SUPPAE	1,431 KB	SAS Xport Transport File
SUPPCM	1,129 KB	SAS Xport Transport File
SUPPDA	2,245 KB	SAS Xport Transport File
SUPPDM	370 KB	SAS Xport Transport File
SUPPDS	105 KB	SAS Xport Transport File
SUPPEG	53 KB	SAS Xport Transport File
SUPPLB	2,233 KB	SAS Xport Transport File
SUPPQS	1,999 KB	SAS Xport Transport File
SUPPSU	20 KB	SAS Xport Transport File
SUPPSV	475 KB	SAS Xport Transport File
SV	846 KB	SAS Xport Transport File
TA	10 KB	SAS Xport Transport File
TE	4 KB	SAS Xport Transport File
TI	12 KB	SAS Xport Transport File
TS	14 KB	SAS Xport Transport File
TV	6 KB	SAS Xport Transport File
VS	7,449 KB	SAS Xport Transport File
define	442 KB	XML Document
define2-0-0	91 KB	XSL Stylesheet

Display 6. A Snapshot of SDTM Datasets and their Resized Datasets in XPT format and Their Define.xml for FDA submission readiness

DOMAIN	VARIABLE	LABEL	TYPE	OLD_LEN	LENGTH	UPDATEDFL
AE	AEACN	Action Taken with Study Treatment	Char	40	16	1
AE	AEACNOTH	Other Action Taken	Char	200	47	1
AE	AEBODSYS	Body System or Organ Class	Char	100	67	1
AE	AEDECOD	Dictionary-Derived Term	Char	100	39	1
AE	AEENDTC	End Date/Time of Adverse Event	Char	20	10	1
AE	AEOUT	Outcome of Adverse Event	Char	40	32	1
AE	AEREL	Causality	Char	40	22	1
AE	AESER	Serious Event	Char	1	1	.
AE	AESEV	Severity/Intensity	Char	8	8	.
AE	AESPID	Sponsor-Defined Identifier	Char	8	2	1
AE	AESTDTC	Start Date/Time of Adverse Event	Char	20	10	1
AE	AETERM	Reported Term for the Adverse Event	Char	200	65	1
AE	DOMAIN	Domain Abbreviation	Char	2	2	.
AE	STUDYID	Study Identifier	Char	20	11	1
AE	USUBJID	Unique Subject Identifier	Char	40	22	1
SUPPAE	IDVAR	Identifying Variable	Char	8	5	1
SUPPAE	IDVARVAL	Identifying Variable Value	Char	8	2	1
SUPPAE	QEVAL	Evaluator	Char	40	1	1
SUPPAE	QLABEL	Qualifier Variable Label	Char	40	39	1
SUPPAE	QNAM	Qualifier Variable Name	Char	8	8	.
SUPPAE	QORIG	Origin	Char	8	3	1
SUPPAE	QVAL	Data Value	Char	200	200	.
SUPPAE	RDOMAIN	Related Domain Abbreviation	Char	2	2	.
SUPPAE	STUDYID	Study Identifier	Char	20	11	1
SUPPAE	USUBJID	Unique Subject Identifier	Char	40	22	1

Display 7. A Snapshot of Resized AE and SUPPAE Specification with New Variables: Original Length and Flag for the Updated Variables

After calling of **%resize_sdtm,** the SDTM programming is "officially" done. There are two sets of SDTM datasets. One is un-resized (original) one, which will be used for ADaM programming. Another is resized one for FDA submission, along with their define.xml.

INTRODUCTION OF RE-SIZING ADAM DATASETS AND THEIR DEFINE.XML

Introduction of ADaM Define.xml Auto Generation

Paper [7] introduces our in-house built SAS-based macro tool to automate the generation of define.xml V2.0 from ADaM specification in MS word format. Once the ADaM metadata is ready, a macro call can automatically generate define.xml 2.0. Please refer to [7] for the details. Hence auto generation of ADaM define.xml for the resized ADaM datasets depends on the auto updates of ADaM metadata for the resized variable lengths.

Introduction of Resizing ADaM Datasets

As mentioned in **INTRODUCTION** section, it is "unclear" if we should resize ADaM datasets at this moment. **General Variable Conventions** from Analysis Data Model Implementation Guide Version 1.1 (p.14) states "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." However, to optimize file size, it is permissible that the length of the variables differ (e.g., trailing blanks may be removed)." Hence the resized character variables inherited into ADaM datasets from SDTM domains should be kept the same length in ADaM datasets as in the SDTM domains for the optimization of file size, and the submitted SDTM to FDA with reduced column sizes, while the derived character variables are not resized. We call it "partially resizing". We can also follow the same practice as STDM to resize all character variables in ADaM. It is called "entirely resizing". We will discuss the "partially resizing" and "entirely resizing" separately in the following sections.

Partially Resizing ADaM Datasets

A macro **%resize_adam** for partially resizing ADaM was developed differently from the one for SDTM resizing. For each variable which is resized in SDTM and is inherited into ADaM, its new length in SDTM can be retrieved from **SDTMSPEC.ALL_VARS_NEW**, which is the output from SDTM resizing programming. Hence we do not have to calculate all character variable actual lengths in each ADaM anymore. The new length is used to reset the variable

length in ADaM for resizing and replace the length in ADaM metadata for the update of its length in deine.xml. Figure 2 shows the process flow of **Partially Resizing ADaM** and Auto Generation of Define.xml.

A subset of **SDTMSPEC.ALL_VARS_NEW** from resizing SDTM is created by the condition UPDATEFL=1, please refer to Display 7; It is named as **SDTM_UPDATED_LEN_VARLIST**. Hence it is the metadata for resized variables in SDTM domains only. A new variable, named as **VARNLEN**, is added by combining the variable name and its new length. **VARNLEN** will be used in the macro **%resize**_. Display 8 shows the snapshot of **SDTM_UPDATED_LEN_VARLIST** in SAS data format from AE and SUPPAE.

DOM_VAR	SDTM_DOM	SDTM_VARIABLE	SDTM_OLD_LEN	SDTM_NEW_LENGTH	VARNLEN
AE.AEACN	AE	AEACN	40	16	AEACN $16
AE.AEACNOTH	AE	AEACNOTH	200	48	AEACNOTH $48
AE.AEBODSYS	AE	AEBODSYS	100	52	AEBODSYS $52
AE.AEDECOD	AE	AEDECOD	100	36	AEDECOD $36
AE.AEENDTC	AE	AEENDTC	20	10	AEENDTC $10
AE.AEENRF	AE	AEENRF	20	7	AEENRF $7
AE.AEHLGT	AE	AEHLGT	100	60	AEHLGT $60
AE.AEHLT	AE	AEHLT	100	57	AEHLT $57
AE.AELLT	AE	AELLT	100	23	AELLT $23
AE.AEOUT	AE	AEOUT	40	26	AEOUT $26
AE.AEREL	AE	AEREL	40	22	AEREL $22
AE.AESOC	AE	AESOC	100	52	AESOC $52
AE.AESPID	AE	AESPID	8	1	AESPID $1
AE.AESTDTC	AE	AESTDTC	20	10	AESTDTC $10
AE.AETERM	AE	AETERM	200	26	AETERM $26
AE.EPOCH	AE	EPOCH	40	9	EPOCH $9
AE.STUDYID	AE	STUDYID	20	12	STUDYID $12
AE.USUBJID	AE	USUBJID	40	23	USUBJID $23
SUPPAE.IDVAR	SUPPAE	IDVAR	8	5	IDVAR $5
SUPPAE.IDVARVAL	SUPPAE	IDVARVAL	8	1	IDVARVAL $1
SUPPAE.QEVAL	SUPPAE	QEVAL	40	1	QEVAL $1
SUPPAE.QLABEL	SUPPAE	QLABEL	40	37	QLABEL $37
SUPPAE.QORIG	SUPPAE	QORIG	8	3	QORIG $3
SUPPAE.QVAL	SUPPAE	QVAL	200	149	QVAL $149
SUPPAE.STUDYID	SUPPAE	STUDYID	20	12	STUDYID $12
SUPPAE.USUBJID	SUPPAE	USUBJID	40	23	USUBJID $23

Display 8. A Snapshot of SDTM_UPDATED_LEN_VARLIST in SAS data format from AE and SUPPAE

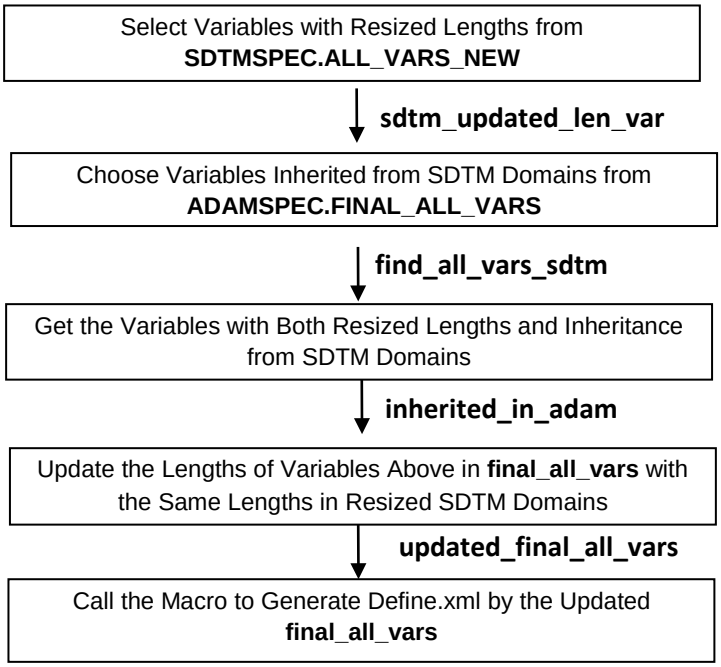

Figure 2. Overview of Process Flow of Partially Resizing ADaM and Auto Generation of Define.xml

A SAS data named **ADAMPSEC.FINAL_ALL_VARS** in our ADaM specification programming subfolder is the key input for auto generation of define.xml V2.0. It is the combination of all ADaM dataset specifications, refer to [7]. Display 9 shows the snapshot of it from ADAE.

VARIABLE	LABEL	TYPE	LENGTH	ORIGIN	CONTROLLED_TERMINOLOGY	COMMENT	CORE
STUDYID	Study Identifier	Char	20	Predecessor		AE.STUDYID	Req
USUBJID	Unique Subject Identifier	Char	40	Predecessor		AE.USUBJID	Req
SUBJID	Subject Identifier for the Study	Char	20	Predecessor		DM.SUBJID	Req
AETERM	Reported Term for the Adverse Event	Char	200	Predecessor		AE.AETERM	Req
AEBODSYS	Body System or Organ Class	Char	100	Predecessor	MedDRA	AE.AEBODSYS	Req
AESER	Serious Event	Char	1	Predecessor		AE.AESER	Req
AESEV	Severity/Intensity	Char	8	Predecessor		AE.AESEV	Perm
ASEV	Analysis Severity/Intensity	Char	8	Derived		If AESEV='MILD' then ASEV='Mild'; Else if AESEV='MODERATE' then ASEV='Moderate'; Else if AESEV='SEVERE' then ASEV='Severe'; Else if AESEV='' and ASTDT<TRTSDT then ASEV='Mild'; Else if AESEV='' and AST	Perm
ASEVN	Analysis Severity/Intensity (N)	Num	8	Assigned	ASEVN (ASEV): (1) 1=Mild (2) 2=Moderate (3) 3=Severe	1 if ASEV='Mild'; 2 if ASEV='Moderate'; 3 if ASEV='Severe'	Perm
AEACN	Action Taken with Study Treatment	Char	40	Predecessor	ACN: (1) DOSE NOT CHANGED (2) DRUG INTERRUPTED (3) DRUG WITHDRAWN (4) NOT APPLICABLE	AE.AEACN	Perm
AEACNOTH	Other Action Taken	Char	200	Predecessor		AE.AEACNOTH	Perm
AEREL	Causality	Char	40	Predecessor		AE.AEREL	Perm
AREL	**Analysis Causality**	**Char**	**40**	**Derived**		**If AEREL is not missing then AREL=propcase(AEREL); Else AREL='Missing'.**	**Perm**

ARELN	Analysis Causality (N)	Num	8	Assigned	ARELN (AREL): (1) 1=Definitely Not Related (2) 2=Probably Not Related (3) 3=Possibly Related (4) 4=Probably Related (5) 5=Definitely Related (6) 9=Missing	1 if AREL='Definitely Not Related'; 2 if AREL='Probably Not Related'; 3 if AREL='Possibly Related'; 4 if AREL='Probably Related'; 5 if AREL='Definitely Related'; 9 if AREL='Missing'	Perm
AEOUT	Outcome of Adverse Event	Char	40	Predecessor		AE.AEOUT	Perm
AEOUTN	Outcome of Adverse Event (N)	Num	8	Assigned	OUTN (AEOUT): (1) 1=NOT RECOVERED/NOT RESOLVED (2) 2=RECOVERED/RESOLVED (3) 3=RECOVERED/RESOLVED WITH SEQUELAE (4) 4=FATAL	1 if AEOUT='NOT RECOVERED/NOT RESOLVED'; 2 if AEOUT='RECOVERED/RESOLVED'; 3 if AEOUT='RECOVERED/RESOLVED WITH SEQUELAE'; 4 if AEOUT='FATAL'	Perm
ADISCRFL	AE Leading to Study Discontinuation (R)	Char	1	Derived	YNULL.NY: (1) Y	ADISCRFL='Y' if AEACN='DRUG WITHDRAWN'	Perm
AESTDTC	Start Date/Time of Adverse Event	Char	20	Predecessor		AE.AESTDTC	Perm
ASTDT	Analysis Start Date	Num	8	Derived	YYMMDD10.	ASTDT=input(substr(AESTDTC, 1, 10), yymmdd10.) if AESTDTC is a complete date; Note: check if there are any partial AE start dates	Cond
AEENDTC	End Date/Time of Adverse Event	Char	20	Predecessor		AE.AEENDTC	Perm
AENDT	Analysis End Date	Num	8	Derived	YYMMDD10.	AENDT=input(AEENDTC,yymmdd10.) if AEENDTC is a complete date; Note: check if there are any partial AE end date	Cond
ADURN	AE Duration (N)	Num	8	Derived		ADURN=AENDT-ASTDT+1	Perm
ADURU	Analysis Duration Units	Char	8	Assigned		'DAYS'	Cond
TRTEMFL	Treatment Emergent Analysis Flag	Char	1	Derived	YNULL.NY: (1) Y	TRTEMFL='Y' if .Z<ADSL.TRTSDT<=ASTDT<=EOTVDT; For any AE with ANL01FL='Y', if its ASEVN is not greater than the maximum of ASEVN from AEs with same AEDECOD present	Cond

Display 9. A Snapshot of ADaM All specification from ADAE in SAS data format

By the selection condition: ORIGIN='Predecessor', a subset of **FINAL_ALL_VARS** is created and named as **FINAL_ALL_VARS_SDTM**, which only contains the variables inherited from SDTM. Display 10 shows the snapshot of all ADaM specification from ADAE in SAS data format only with the variables inherited from AE Domain.

DOM_VAR	VARIABLE	LABEL	TYPE	ADAM _LEN	ORIGIN	CONTROLLED_ TERMINOLOGY	CORE
AE.STUDYID	STUDYID	Study Identifier	Char	20	Predecessor		Req
AE.USUBJID	USUBJID	Unique Subject Identifier	Char	40	Predecessor		Req
DM.SUBJID	SUBJID	Subject Identifier for the Study	Char	20	Predecessor		Req
AE.AETERM	AETERM	Reported Term for the Adverse Event	Char	200	Predecessor		Req
AE.AEBODSYS	AEBODSYS	Body System or Organ Class	Char	100	Predecessor	MedDRA	Req
AE.AESER	AESER	Serious Event	Char	1	Predecessor		Req
AE.AESEV	AESEV	Severity/Intensity	Char	8	Predecessor		Perm
AE.AEACN	AEACN	Action Taken with Study Treatment	Char	40	Predecessor	ACN: (1) DOSE NOT CHANGED (2) DRUG INTERRUPTED (3) DRUG WITHDRAWN (4) NOT APPLICABLE	Perm
AE.AEACNOTH	AEACNOTH	Other Action Taken	Char	200	Predecessor		Perm
AE.AEREL	AEREL	Causality	Char	40	Predecessor		Perm
AE.AEOUT	AEOUT	Outcome of Adverse Event	Char	40	Predecessor		Perm
AE.AESTDTC	AESTDTC	Start Date/Time of Adverse Event	Char	20	Predecessor		Perm
AE.AEENDTC	AEENDTC	End Date/Time of Adverse Event	Char	20	Predecessor		Perm

Display 10. A Snapshot of All ADaM Specification from ADAE in SAS data format only with Variables Inherited from AE Domain

By merging **SDTM_UPDATED_LEN_VARLIST** and **FINAL_ALL_VARS_SDTM** by **DOM_VAR,** the resized variables in SDTM domains, inherited into ADaM datasets, are identified. The new SAS dataset, named as **INHERITED_IN_ADAM**, has the variable names and their original and new lengths. Display 11 shows the snapshot of **INHERITED_IN_ADAM** from ADAE with the resized variables, inherited from AE domain.

VARIABLE	LABEL	TYPE	DOM_VAR	ADAM LEN	SDTM_OLD_LEN	SDTM_NEW_LENGTH	ADAMVR	SDTMVAR
STUDYID	Study Identifier	Char	AE.STUDYID	20	20	12	1	1
USUBJID	Unique Subject Identifier	Char	AE.USUBJID	40	40	23	1	1
SUBJID	Subject Identifier for the Study	Char	DM.SUBJID	20	20	10	1	1
AESPID	Sponsor-Defined Identifier	Char	AE.AESPID	8	8	1	1	1
AETERM	Reported Term for the Adverse Event	Char	AE.AETERM	200	200	26	1	1
AELLT	Lowest Level Term	Char	AE.AELLT	100	100	23	1	1
AEDECOD	Dictionary-Derived Term	Char	AE.AEDECOD	100	100	36	1	1
AEHLT	High Level Term	Char	AE.AEHLT	100	100	57	1	1
AEHLGT	High Level Group Term	Char	AE.AEHLGT	100	100	60	1	1
AEBODSYS	Body System or Organ Class	Char	AE.AEBODSYS	100	100	52	1	1
AESOC	Primary System Organ Class	Char	AE.AESOC	100	100	52	1	1
AEACN	Action Taken with Study Treatment	Char	AE.AEACN	40	40	16	1	1
AEACNOTH	Other Action Taken	Char	AE.AEACNOTH	200	200	48	1	1
AEREL	Causality	Char	AE.AEREL	40	40	22	1	1
AEOUT	Outcome of Adverse Event	Char	AE.AEOUT	40	40	26	1	1
AESTDTC	Start Date/Time of Adverse Event	Char	AE.AESTDTC	20	20	10	1	1
AEENDTC	End Date/Time of Adverse Event	Char	AE.AEENDTC	20	20	10	1	1

Display 11. A Snapshot of All INHERITED_IN_ADAM from ADAE in SAS data format with the Resized Variables, Inherited from AE Domain

By merging **FINAL_ALL_VARS** and **INHERITED_IN_ADAM** by **DOMAIN and VARIABLE,** the length will be updated by **INHERITED_IN_ADAM.SDTM_NEW_LENGTH** if the variables are in **INHERITED_IN_ADAM.** The updated SAS dataset is named **UPDATED_FINAL_ALL_VARS,** which will be used to call the macro **%get_adam_definexml_call** for auto generation of define.xml for ADaM. Hence the define.xml will keep the same lengths as ones in SDTM for the resized variables in SDTM domains, which were inherited into ADaM datasets. Display 12 shows the snapshot of **UPDATED_FINAL_ALL_VARS** from ADAE. The variables with non-missing SDTM_NEW_LENGTH have the updated LENGTH from ones in SDTM.AE.

VARIABLE	LABEL	TYPE	LENGTH	ORIGIN	CONTROLLED_TERMINOLOGY	COMMENT	CORE	SDTM_NEW_LENGTH	ADAM_OLD_LEN
STUDYID	Study Identifier	Char	12	Predecessor		AE.STUDYID	Req	12	20
USUBJID	Unique Subject Identifier	Char	23	Predecessor		AE.USUBJID	Req	23	40
SUBJID	Subject Identifier for the Study	Char	10	Predecessor		DM.SUBJID	Req	10	20
AETERM	Reported Term for the Adverse Event	Char	26	Predecessor		AE.AETERM	Req	26	200
AEBODSYS	Body System or Organ Class	Char	52	Predecessor	MedDRA	AE.AEBODSYS	Req	52	100
AESER	Serious Event	Char	1	Predecessor		AE.AESER	Req	.	1
AESEV	Severity/Intensity	Char	8	Predecessor		AE.AESEV	Perm	.	8
ASEV	Analysis Severity/Intensity	Char	8	Derived		If AESEV='MILD' then ASEV='Mild'; Else if AESEV='MODERATE' then ASEV='Moderate'; Else if AESEV='SEVERE' then ASEV='Severe'; Else if AESEV='' and ASTDT<TRTSDT then ASEV='Mild'; Else if AESEV='' and AST	Perm	.	8
ASEVN	Analysis Severity/Intensity (N)	Num	8	Assigned	ASEVN (ASEV): (1) 1=Mild (2) 2=Moderate (3) 3=Severe	1 if ASEV='Mild'; 2 if ASEV='Moderate'; 3 if ASEV='Severe'	Perm	.	8
AEACN	Action Taken with Study	Char	16	Predecessor	ACN: (1) DOSE NOT CHANGED	AE.AEACN	Perm	16	40

	Treatment				(2) DRUG INTERRUPTED (3) DRUG WITHDRAWN (4) NOT APPLICABLE				
AEACNOTH	Other Action Taken	Char	48	Predecessor		AE.AEACNOTH	Perm	48	200
AEREL	Causality	Char	22	Predecessor		AE.AEREL	Perm	22	40
AREL	**Analysis Causality**	**Char**	**40**	**Derived**		**If AEREL is not missing then AREL=propcase(AEREL); Else AREL='Missing'.**	**Perm**	**.**	**40**
ARELN	Analysis Causality (N)	Num	8	Assigned	ARELN (AREL): (1) 1=Definitely Not Related (2) 2=Probably Not Related (3) 3=Possibly Related (4) 4=Probably Related (5) 5=Definitely Related (6) 9=Missing	1 if AREL='Definitely Not Related'; 2 if AREL='Probably Not Related'; 3 if AREL='Possibly Related'; 4 if AREL='Probably Related'; 5 if AREL='Definitely Related'; 9 if AREL='Missing'	Perm	.	8
AEOUT	Outcome of Adverse Event	Char	26	Predecessor		AE.AEOUT	Perm	26	40
AEOUTN	Outcome of Adverse Event (N)	Num	8	Assigned	OUTN (AEOUT): (1) 1=NOT RECOVERED/NOT RESOLVED (2) 2=RECOVERED/RESOLVED (3) 3=RECOVERED/RESOLVED WITH SEQUELAE (4) 4=FATAL	1 if AEOUT='NOT RECOVERED/NOT RESOLVED'; 2 if AEOUT='RECOVERED/RESOLVED'; 3 if AEOUT='RECOVERED/RESOLVED WITH SEQUELAE'; 4 if AEOUT='FATAL'	Perm	.	8
ADISCRFL	AE Leading to Study Discontinuation (R)	Char	1	Derived	YNULL.NY: (1) Y	ADISCRFL='Y' if AEACN='DRUG WITHDRAWN'	Perm	.	1
AESTDTC	Start Date/Time of Adverse Event	Char	10	Predecessor		AE.AESTDTC	Perm	10	20
ASTDT	Analysis Start Date	Num	8	Derived	YYMMDD10.	ASTDT=input(substr(AESTDTC, 1, 10), yymmdd10.) if AESTDTC is a complete date; Note: check if there are any partial AE start dates	Cond	.	8
AEENDTC	End Date/Time of Adverse Event	Char	10	Predecessor		AE.AEENDTC	Perm	10	20
AENDT	Analysis End Date	Num	8	Derived	YYMMDD10.	AENDT=input(AEENDTC,yymmdd10.) if AEENDTC is a complete date; Note: check if there are any partial AE end date	Cond	.	8
ADURN	AE Duration (N)	Num	8	Derived		ADURN=AENDT-ASTDT+1	Perm	.	8
ADURU	Analysis Duration Units	Char	8	Assigned		'DAYS'	Cond	.	8
TRTEMFL	Treatment Emergent Analysis Flag	Char	1	Derived	YNULL.NY: (1) Y	For BASETYPE='Entire Safety Period', TRTEMFL='Y' if .Z<ADSL.TRTSDT<=ASTDT<=EOTVDT; For any AE with ANL01FL='Y', if its ASEVN is not greater than the maximum of ASEVN from AEs with same AEDECOD present	Cond	.	1

Display 12. A Snapshot of UPDATED_FINAL_ALL_VARS from ADAE in SAS data format with the Same Lengths As Ones in Resized AE for the Resized Variables Inherited from AE Domain

Display 13 shows the snapshot of define.xml from ADAE. It shows that the lengths of AEACN, AEACNOTH, and AEREL are the same as ones in SDTM.AE, shown in Display 2. However the length of derived variable: ADAE.AREL is 40 characters long, and it is not resized due to the partially resizing.

Adverse Event Analysis Dataset (ADAE) [Location: ADAE.xpt]

Variable	Label	Type	Length / Display Format	Controlled Terms or Format	Source/Derivation/Comment
STUDYID	Study Identifier	text	12		Predecessor: AE.STUDYID
USUBJID	Unique Subject Identifier	text	23		Predecessor: AE.USUBJID

AEACN	Action Taken with Study Treatment	text	16	["DOSE NOT CHANGED", "DRUG INTERRUPTED", "DRUG WITHDRAWN", "NOT APPLICABLE"] <ADAE.ACN>	Predecessor: AE.AEACN
AEACNOTH	Other Action Taken	text	48		Predecessor: AE.AEACNOTH
AEREL	Causality	text	22		Predecessor: AE.AEREL
AREL	Analysis Causality	text	40	ADAE.AREL	Derived: If AEREL is not missing then AREL=propcase(AEREL); Else AREL='Missing'.
ARELN	Analysis Causality (N)	integer	8	ADAE.ARELN	Assigned: 1 if AREL='Definitely Not Related'; 2 if AREL='Probably Not Related'; 3 if AREL='Possibly Related'; 4 if AREL='Probably Related'; 5 if AREL='Definitely Related'; 9 if AREL='Missing'

Display 13. An Example of Define.xml Version 2.0 for ADAE Dataset with the Alteration of Length of Inherited SDTM Variables

The SAS codes for generating ADaM define.xml are as follows.

```
*** From SDTM specs,choose variables with resized lengths;
data sdtm_updated_len_varlist;
     length varnlen $16.;
     set sdtmspec.all_vars_new;
     varnlen=strip(variable)||' $'||strip(put(length,best.));
     where updatedfl=1;
run;
proc sort data=sdtm_updated_len_varlist
     out=sdtm_updated_len_varlist2
         (keep=domain dom_var variable varnlen old_len new_length
          rename=(domain=sdtm_dom variable=sdtm_variable old_len=sdtm_old_len
                  new_length=sdtm_new_length));
     by dom_var;
run;
*** From ADaM specs, choose variables inherited from SDTM domains;
proc sort data=adamspec.final_all_vars
           out=final_all_vars_sdtm(rename=(length=adam_len comment=dom_var));
     by comment domain variable;
     where origin='Predecessor';
run;
*** Get the variables with both resized lengths and inheritance from SDTM domain;
data inherited_in_adam;
     merge final_all_vars_sdtm(in=a) sdtm_updated_len_varlist2(in=b);
     by dom_var;
     if a then adamvr=1;
     if b then sdtmvar=1;
     if a and b;
run;
proc sort data=inherited_in_adam;by domain variable;run;

*** Update the lengths of variables from final_all_vars with both resized
    lengths and inheritance from SDTM domain with ones from the resizing in SDTM;
proc sort data=adamspec.final_all_vars out=final_all_vars;
```

```
     by domain variable;
run;
data updated_final_all_vars;
     merge final_all_vars inherited_in_adam(in=b keep=domain variable
sdtm_new_length);
     by domain variable;
     adam_old_len=length;
     if b then LENGTH=sdtm_new_length;
run;

*** Call the macro for define.xml;
%get_adam_definexml_call(lib=adamspec,vardata=final_all_vars,domaindata=all_domains,ou
tdir=&adam_pgm_spec.\,xmldir=&adam_xml.\);
```

The partial SAS codes for resizing the lengths of variables inherited from SDTM domains are as follows.

```
*** Get the ADaM Dataset Names;
%macro get_datanm;
    proc sort data=updated_final_all_vars out=datalist nodupkeys;by domain;run;
    data _null_;
         set datalist end=last;
        if last then call symput('totn',strip(put(_n_,best.)));
        call symput('datanm'||strip(put(_n_,best.)),domain);
    run;
%mend;
*** Resizing the lengths of variables inherited from Each SDTM domain;
%macro trim2(indsn=,outdsn=);
    proc sort data=updated_final_all_vars
               out=_adam_spec(keep=domain variable varseq sdtm_dom comment varnlen);
         by varseq;
         where domain="&indsn";
    run;
    proc sql noprint;
         select trim(left(varnlen)) into: newlen separated by ' ' from _adam_spec;
    run;
    quit;
    options varlenchk=nowarn;
    data &outdsn;
         length &newlen;;
        set adam.&indsn;
    run;
%mend;
```

SAS codes inside macro %resize_adam for calling %trim2 and create XPT files for FDA submission;

```
     %get_datanm;
     %do j=1 %to &totn;
         %let _name=&&datanm&j;
         %trim2(indsn=%str(&_name),outdsn=%str(&_name.2));
       *** create XPT files and store them in final\esub\analysis for FDA submission with resized variables
         inherited from SDTM domains;
         libname xp xport "&adam_xml.\&_name..xpt";
         proc copy in=work out=xp memtype=data;
              select &_name.2;
         run;
     %end;
```

Entirely Resizing ADaM Datasets

After SDTM datasets are finalized, all ADaM programming (both production and validation) is rerun. The final ADaM datasets are saved in the standard study folder: ..\Study123\final\data\ADaM\pd. These ADaM datasets are used for resized for FDA submission.

Similar to Resizing SDTM domains, all character variables in ADaM dataset can be resized by calling modified macro %trim1. The actual lengths of all character variables in each ADaM dataset need to be calculated for resizing and the new lengths is used for updates of define.xml. Please refer to the section for SDTM resizing. After the calling macro **%resize_adam**, the character variables in each ADaM data are resized. The metadata is updated for the resized variable lengths. Display 14 shows the snapshot of **FINAL_ALL_VARS** from ADAE. It shows that the length of resized ADAE.AREL is 22 characters long, instead of 40, originally defined length, compared to the partially resizing ADaM Datasets, shown in Display 13.

VARIABLE	LABEL	TYPE	LENGTH	ORIGIN	CONTROLLED_TERMINOLOGY	COMMENT	CORE	ADAM_OL_LEEN
STUDYID	Study Identifier	Char	12	Predecessor		AE.STUDYID	Req	12
USUBJID	Unique Subject Identifier	Char	23	Predecessor		AE.USUBJID	Req	23
SUBJID	Subject Identifier for the Study	Char	10	Predecessor		DM.SUBJID	Req	10
AETERM	Reported Term for the Adverse Event	Char	26	Predecessor		AE.AETERM	Req	26
AEBODSYS	Body System or Organ Class	Char	52	Predecessor	MedDRA	AE.AEBODSYS	Req	52
AESER	Serious Event	Char	1	Predecessor		AE.AESER	Req	1
AESEV	Severity/Intensity	Char	8	Predecessor		AE.AESEV	Perm	8
ASEV	Analysis Severity/Intensity	Char	8	Derived		If AESEV='MILD' then ASEV='Mild'; Else if AESEV='MODERATE' then ASEV='Moderate'; Else if AESEV='SEVERE' then ASEV='Severe'; Else if AESEV='' and ASTDT<TRTSDT then ASEV='Mild'; Else if AESEV='' and AST	Perm	8
ASEVN	Analysis Severity/Intensity (N)	Num	8	Assigned	ASEVN (ASEV): 1) 1=Mild (2) 2=Moderate (3) 3=Severe	1 if ASEV='Mild'; 2 if ASEV='Moderate'; 3 if ASEV='Severe'	Perm	8
AEACN	Action Taken with Study Treatment	Char	16	Predecessor	ACN: (1) DOSE NOT CHANGED (2) DRUG INTERRUPTED (3) DRUG WITHDRAWN (4) NOT APPLICABLE	AE.AEACN	Perm	16
AEACNOTH	Other Action Taken	Char	48	Predecessor		AE.AEACNOTH	Perm	48
AEREL	Causality	Char	22	Predecessor		AE.AEREL	Perm	22
AREL	**Analysis Causality**	**Char**	**22**	**Derived**		**If AEREL is not missing then AREL=propcase(AEREL); Else AREL='Missing'.**	**Perm**	**40**
ARELN	Analysis Causality (N)	Num	8	Assigned	ARELN (AREL): (1) 1=Definitely Not Related (2) 2=Probably Not Related (3) 3=Possibly Related (4) 4=Probably Related (5) 5=Definitely Related (6) 9=Missing	1 if AREL='Definitely Not Related'; 2 if AREL='Probably Not Related'; 3 if AREL='Possibly Related'; 4 if AREL='Probably Related'; 5 if AREL='Definitely Related'; 9 if AREL='Missing'	Perm	8

Display 14. shows that the length of resized ADAE.AREL is 22 characters long, instead of 40, originally defined length.

Display 15 shows the snapshot of define.xml from ADAE. It shows that all character variable lengths are resized, compared to “partially resized” in Display 13, where the derived variable: ADAE.AREL is not resized.

Adverse Event Analysis Dataset (ADAE) [Location: ADAE.xpt]

Variable	Label	Type	Length / Display Format	Controlled Terms or Format	Source/Derivation/Comment
STUDYID	Study Identifier	text	12		Predecessor: AE.STUDYID
USUBJID	Unique Subject Identifier	text	23		Predecessor: AE.USUBJID
AEACN	Action Taken with Study Treatment	text	16	["DOSE NOT CHANGED", "DRUG INTERRUPTED", "DRUG WITHDRAWN", "NOT APPLICABLE"] <ADAE.ACN>	Predecessor: AE.AEACN
AEACNOTH	Other Action Taken	text	48		Predecessor: AE.AEACNOTH
AEREL	Causality	text	22		Predecessor: AE.AEREL
AREL	Analysis Causality	text	22	ADAE.AREL	Derived: If AEREL is not missing then AREL=propcase (AEREL); Else AREL='Missing'.
ARELN	Analysis Causality (N)	integer	8	ADAE.ARELN	Assigned: 1 if AREL='Definitely Not Related'; 2 if AREL='Probably Not Related'; 3 if AREL='Possibly Related'; 4 if AREL='Probably Related'; 5 if AREL='Definitely Related'; 9 if AREL='Missing'

Display 15. An Example of Define.xml Version 2.0 for ADAE Dataset with All Resized Character Variables The left side of Display 16 shows an example of the un-resized ADaM dataset, which are used to generate tables, figures, and listings (TFLs) internally. They are resized by calling a macro **%resize_ADaM**. After calling of **%resize_ ADaM for entirely resizing,** the ADaM programming is finalized. The right side of Display 16 shows an example of the resize ADaM datasets in XPT format and their Define.xml for FDA submission readiness. The reduction of each data file size can be easily seen in Display 16.

Name	Type	Size
adae	SAS Data Set	10,584 KB
adaep	SAS Data Set	3,808 KB
adcgis	SAS Data Set	30,912 KB
adcm	SAS Data Set	20,480 KB
adcows	SAS Data Set	25,016 KB
adcssrs	SAS Data Set	988,528 KB
adda	SAS Data Set	103,992 KB
addv	SAS Data Set	344 KB
adeg	SAS Data Set	129,092 KB
adhama	SAS Data Set	286,000 KB
adhamdp	SAS Data Set	64,856 KB
adlb	SAS Data Set	1,320,448 KB
admadrs	SAS Data Set	513,416 KB
adsl	SAS Data Set	1,792 KB
adslp	SAS Data Set	576 KB
advs	SAS Data Set	199,716 KB

Name	Size	Type
ADAE	5,499 KB	SAS Xport Transport File
ADAEP	1,964 KB	SAS Xport Transport File
ADCGIS	14,104 KB	SAS Xport Transport File
ADCM	9,815 KB	SAS Xport Transport File
ADCOWS	12,120 KB	SAS Xport Transport File
ADCSSRS	631,296 KB	SAS Xport Transport File
ADDA	55,373 KB	SAS Xport Transport File
ADDV	93 KB	SAS Xport Transport File
ADEG	58,948 KB	SAS Xport Transport File
ADHAMA	131,465 KB	SAS Xport Transport File
ADHAMDP	38,061 KB	SAS Xport Transport File
ADLB	599,299 KB	SAS Xport Transport File
ADMADRS	309,422 KB	SAS Xport Transport File
ADSL	789 KB	SAS Xport Transport File
ADSLP	311 KB	SAS Xport Transport File
ADVS	108,109 KB	SAS Xport Transport File

Display 16. A Snapshot of ADaM Datasets, and Their Resized Datasets in XPT format with Their Define.xml for FDA submission readiness

In summary, two sets of ADaM datasets are generated from ADaM programming after data base lock in our ADAM programming process. One is un-resized (original) one, which will be used for TFLs programming internally. Another is the resized one for FDA submission, along with their define.xml. Side by side comparison of file sizes of each ADaM dataset highlights the reduction of file sizes from each character variable column size reduction. Between partially resizing and entirely resizing introduced above, entirely resizing ADaM is our approach.

How to Handle Resizing SDTM and ADaM for ISS/ISE?

Integrated Summary of Safety (ISS) and an Integrated Summary of Efficacy (ISE) sometimes are part of FDA submission. Some sponsors create integrated SDTM datasets to support the creation of integrated ADaM datasets. It is justified at least by recoding events and medications into a common version of a dictionary. The discussion of integrated SDTM dataset structures and content is outside the scope of this presentation. Table 1 shows three approaches for SDTM and ADaM programming for ISS/ISE FDA submission.

	SDTM Integration Method	Resizing SDTM Datasets for FDA Submission	ADaM for ISS/ISE Method	Resizing ADaM Datasets for FDA Submission
1	integrated SDTM datasets	Same as an individual study	Use integrated SDTM datasets	Same as an individual study
2	No Integration	No Need	Use multiple Sets of SDTM	Same as an individual study
3	No Integration	No Need	Use multiple Sets of ADaM	Same as an individual study

Table 1. Different Integration Methods for SDTM and ADaM for ISS/ISE FDA Submission

Method 3 is our recommendation if there are no recoding events or medications into a common version of a dictionary.

What is the Impact of Resizing SDTM/ADaM for FDA Submission to Our SDTM/ADaM Programming?

With the help of our SAS-based macro introduced above, resizing SDTM and ADaM has almost zero impact to our clinical programming. Our standard SDTM specification and ADaM standard templates can be used to develop SDTM and ADaM without identifying or predicting up front the longest potential value for each character variable, to avoid risk of truncation. It facilitates the standardization of programming process. The auto generation of define.xml for SDTM and ADaM also makes FDA submission preparation to be more technically accurate and operationally efficient. This method is especially useful to ISS/ISE programming since the standardized lengths of character variables across all the studies avoids the potential truncation issue and/or programming warnings when integrating SDTM/ADaM data from different studies.

CONCLUSION

This paper introduces a SAS-based macro approach which automates the resizing of character variables in both SDTM and ADaM datasets, and simultaneously updating the resized variable lengths in define.xml. We also discuss the strategy for handling each individual study and ISS/ISE in terms of resizing SDTM and ADaM to share the vision to achieve technical accuracy and operational efficiency.

REFERENCES

[1] U.S. Department of Health and Human Services, Food and Drug Administration, Study Data Technical Conformance Guide: Technical Specifications Document. December 2014. Available at http://www.fda.gov/downloads/ForIndustry/DataStandards/StudyDataStandards/UCM384744.pdf.

[2] FDA/PhUSE CSS Data Quality Working Group. “Data Sizing Best Practices Recommendation”. PhUSE Wiki. Available at http://www.phusewiki.org/wiki/index.php?title=Data_Sizing_Best_Practices_Recommendation

[3] Sandra VanPelt Nguyen. “Reducing Variable Lengths for Submission Dataset Size Reduction”. PharmaSUG 2014, Paper CC37. Available at http://www.pharmasug.org/proceedings/2014/CC/PharmaSUG-2014-CC37.pdf

[4] Jianjun Tan and Eric Liao. “A macro to re-size character variable length”, PharmaSUG China 2015-26.

[5] Wayne Zhong. “Efficiently Trim Character Variable Lengths to Fit Data, Reduce Dataset Size”. PharmaSUG2012, Paper CC17

[6] Min Chen, Xiangchen Cui, “A SAS® Macro Tool to Automate Generation of Define.xml V2.0 from SDTM Specification for FDA Submission”. PharmaSUG2016 - Paper SS08

[7] Min Chen, Xiangchen Cui, “A SAS® Macro Tool to Automate Generation of Define.xml V2.0 from ADaM Specification for FDA Submission”. PharmaSUG2015 - Paper AD06

CONTACT INFORMATION

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

Name: Xiangchen (Bob) Cui**,** Ph.D.
Enterprise: Alkermes, Inc.
Address: 852 Winter Street
City, State ZIP: Waltham, MA 02451

Work Phone: 781-609-6038
Fax: 781-609-5855
E-mail:xiangchen.cui@alkermes.com

Name: Min Chen**,** Ph.D.
Enterprise: Alkermes, Inc.
Address: 852 Winter Street
City, State ZIP: Waltham, MA 02451

Work Phone: 781-609-6047
Fax: 781-609-5855
E-mail: min.chen@alkermes.com