Transforming Data for CDISC and ISS

Sy Truong, Meta-Xceed, Inc, Fremont, CA

ABSTRACT

Analyzing and reporting data using SAS is only as effective as the data that you are working with. Working with large sets of data requires that the data be uniform and standardized. When working with clinical trials data, this becomes apparent when you are working on an Integrated Safety Summary (ISS) or when you are transforming data into a standard such as CDISC. This paper presents methodologies and tools to automate and optimize the transformation of large sets of data in these scenarios. Some of the topics this paper will address include:

1. Organizing the transformation of data in a specification
2. Expressing the transformation specification in a model
3. Automating the process of applying the transformation model.
4. Methods of validating the transformation

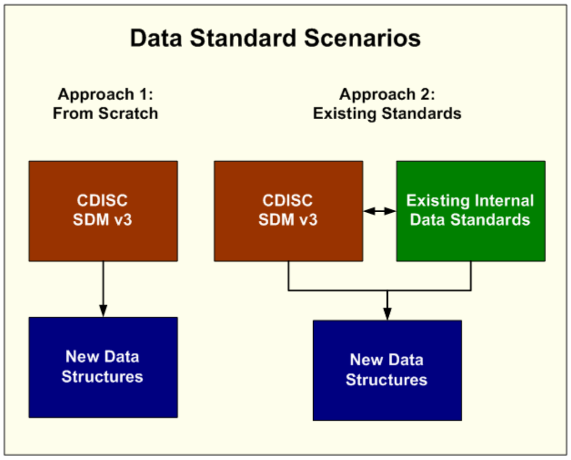
This paper will present some of the challenges presented in transforming large sets of data into one standard format. It will then demonstrate methodologies and automation tools such as Transdata™ and CDISC Builder™ which make the process less prone to errors. The paper will explore approaches that create an abstract of what the transformation is doing. It then captures this in a model that can generate code to implement the transformation.

INTRODUCTION

Transforming data from one structure to another is the most rudimental and oldest database task since the beginning relational database. One of the reasons why data needs to be transformed is that source data is in a non-standard form as compared to a target structure. It would make sense that all data should originally be set up with a standardized target structure so that no transformations are needed. In the ideal world, that would be the case. However, in reality, many scenarios arise that create a plethora of non-standard data structures that need to be transformed.

For example, a small biotech company discovers a potential blockbuster drug and starts clinical trials to gain approval. When they start out, the company does not have a data management or a statistical analysis and reporting department. They start out with Phase I trials outsourcing all of its work to Contract Research Organizations (CROs). The project lead of this project is a Microbiologist and has no background in computer sciences or database design. He place trust upon the expertise of the CROs to setup the data structure to capture the information needed. As the Phase I trials prove to be successful, Phase II trials were added. The original CRO used did not have enough resources, so the new studies were outsourced to other CROs to perform similar tasks. Once these studies prove to be successful, the biotech gained funding to form an internal team. The goal was to bring the data in-house to perform integrated safety summaries (ISS) and also prepare for electronic submission using CDISC Standards. Each CRO had their own quirks and standards. When the data manager and SAS programmer were hired to do the job, they were presented with large sets of data stored in different structures from each CRO. The job now was to transform all the data into a uniform standard to meet the objective of generating summary reports for the electronic submission. This is a common example of how good intentions can still lead to non-standard structures which requires data transformations.

There are two general approaches towards achieving data standards. If you are starting from scratch, it makes sense to use a suggested standard such as CDISC. In this case, the effort will be in ensuring that new data created adheres to this standard. A second approach is when you already have existing data that is structured very different from CDISC standards. In this case, the task is to make sure that all existing data structures follow an internal standard. Any new data created would then need to adhere to this new standard. It is more common for the second scenario to occur.



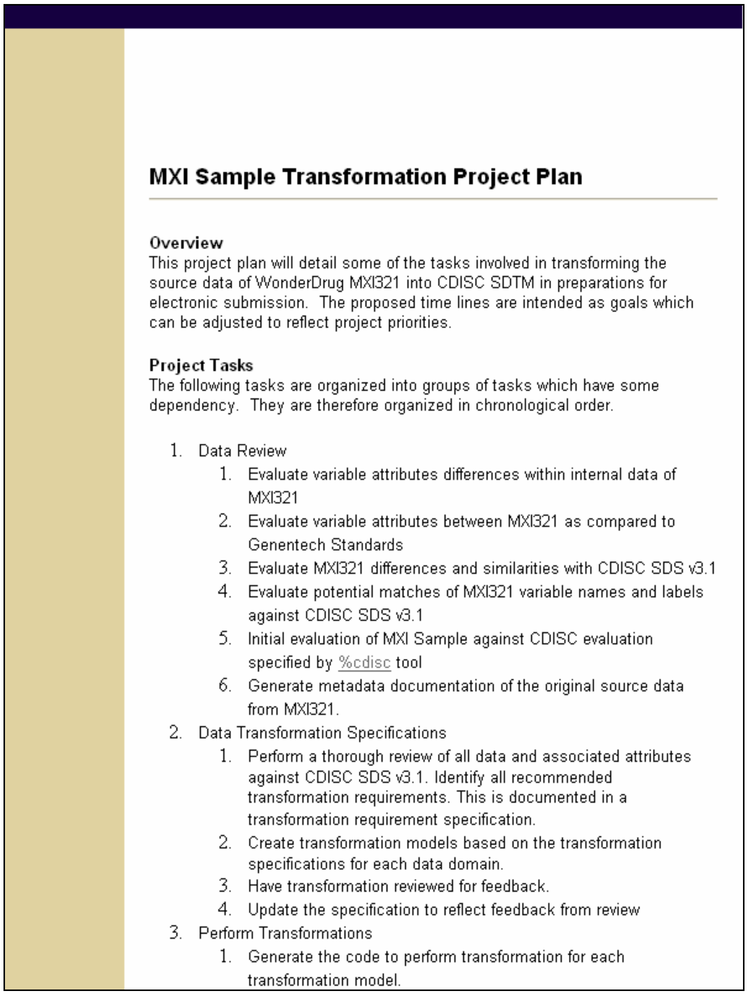
WHY STANDARDIZE?

Establishing data standards and applying the standards across all studies and projects can be resource intensive. It is reasonable to ask the question whether the effort is worth it. One of the key benefits is that the programs associated with this data become more portable. They can be moved from one study to the next with minor modifications. Not only are the programs more portable, the programmer and statistician working on one study can understand a new study with the same structure relatively quickly compared to learning a new set of programs, macros and data structures. The standard data structure allows for performing analysis across studies that would not be feasible if each study had their own data structure. This is also true for the FDA. If they need to perform aggregate summaries across different submissions from different companies, it would be impossible unless the companies follow an industry standard structure such as CDISC. The benefits of standardized datasets are truly realized when cross studies analysis can be performed where it was not once possible. The productivity gain is sometimes difficult to measure but, in the long run, it will outweigh the efforts invested in standardizing.

IMPLEMENTING CDISC

Applying the data transformation is a pivotal step but it is one among many tasks that need to be performed in order to have a successful CDISC or data standards implementation. This section describes a recommended step by step methodology towards implementing this standards transformation.

STEP 1: Before any code is written or even any specifications are drawn up, it is important to define a transformation plan. This does not have to be formal but a project plan will help the entire team work together cohesively towards a common objective. The project plan should include the list of tasks and an estimated time line for the tasks at hand. An example of an abbreviated test plan is shown here:



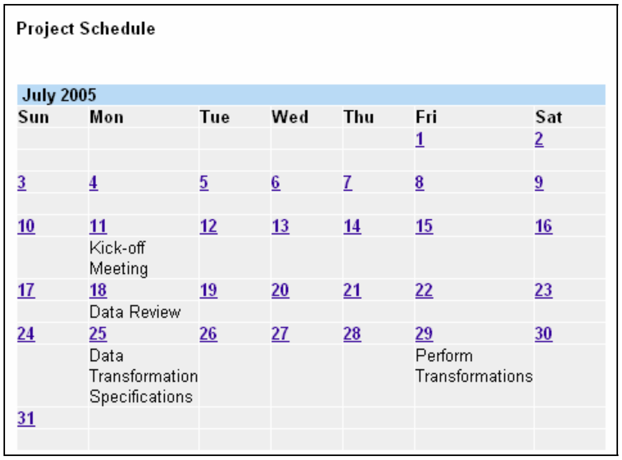
MXI Sample Transformation Project Plan

Overview
This project plan will detail some of the tasks involved in transforming the source data of WonderDrug MXI321 into CDISC SDTM in preparations for electronic submission. The proposed time lines are intended as goals which can be adjusted to reflect project priorities.

Project Tasks
The following tasks are organized into groups of tasks which have some dependency. They are therefore organized in chronological order.

1. Data Review
   1. Evaluate variable attributes differences within internal data of MXI321
   2. Evaluate variable attributes between MXI321 as compared to Genentech Standards
   3. Evaluate MXI321 differences and similarities with CDISC SDS v3.1
   4. Evaluate potential matches of MXI321 variable names and labels against CDISC SDS v3.1
   5. Initial evaluation of MXI Sample against CDISC evaluation specified by %cdisc tool
   6. Generate metadata documentation of the original source data from MXI321.
2. Data Transformation Specifications
   1. Perform a thorough review of all data and associated attributes against CDISC SDS v3.1. Identify all recommended transformation requirements. This is documented in a transformation requirement specification.
   2. Create transformation models based on the transformation specifications for each data domain.
   3. Have transformation reviewed for feedback.
   4. Update the specification to reflect feedback from review
3. Perform Transformations
   1. Generate the code to perform transformation for each transformation model.

The test plan contains high level tasks which would then be placed on the projected schedule. These milestones are target dates for projected completion.



Project Schedule

July 2005

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6	7	8	9
10	11 Kick-off Meeting	12	13	14	15	16
17	18 Data Review	19	20	21	22	23
24	25 Data Transformation Specifications	26	27	28	29 Perform Transformations	30
31						

STEP 2: Perform data review of the existing source structure. This will capture deviations which will be resolved before any transformation occurs. It also highlights findings between differences and similarities between the source and the target CDISC structures. These findings will help identify what needs to be transformed. Some of the recommended review tasks

include:

Tasks	Description
Source Data Review	Evaluate variable attributes differences within internal data and Source Data.
Source Data and Internal Standards	Evaluate variable attributes between Source Data as compared to Internal Standards.
Source Data and CDISC SDS 3.1	Evaluate Source Data differences and similarities with CDISC SDS v3.1.
Source Data and CDISC SDS 3.1	Evaluate potential matches of Source Data variable names and labels against CDISC SDS v3.1.
CDISC Evaluation	Evaluate Source Data against CDISC evaluation from a set of specified rules.
Source Data Attributes	Review PROC CONTENTS and PROC PROC PRINT of Source Data.

The review can be performed manually with the use of PROC REPORT. Considerable attention has to be paid to the details of the attributes in order to capture the discrepancies through visual inspection. Some example findings are shown here:

1. The most significant finding is that there are 125 incidences of variable labels being inconsistent for the same variable name. Some of these were how the subject was labeled. For example, the variable PATNUM has these two different labels across different datasets:

Data name	Variable	Label
ae	patnum	**PROTOCOL CASE NO.**
pat	patnum	**Subject ID (Num)**

2. There were three incidences where the variable lengths differ for the same variable.

Data name	Variable	Type	Length	Label
death	comment	C	**300***	Death comment*
txsumm	comment	C	**200***	Comments*
pat	tythmod	C	**25***	Type of Dose Mod to Protocol Therapy*
txsumm	tythmod	C	**100***	Type of modified therapy*
resps	comment	C	**300***	Comment*
txsumm	comment	C	**200***	Comments*

3. When compared with CDISC SDS v.31, the findings were revealing. For example, there were 120 instances found of variable lengths that were different between MXI213 and the CDISC data model yet they share the same variable name. There were multiple occurrences of the same variables such as SEX, RACE and VISIT.

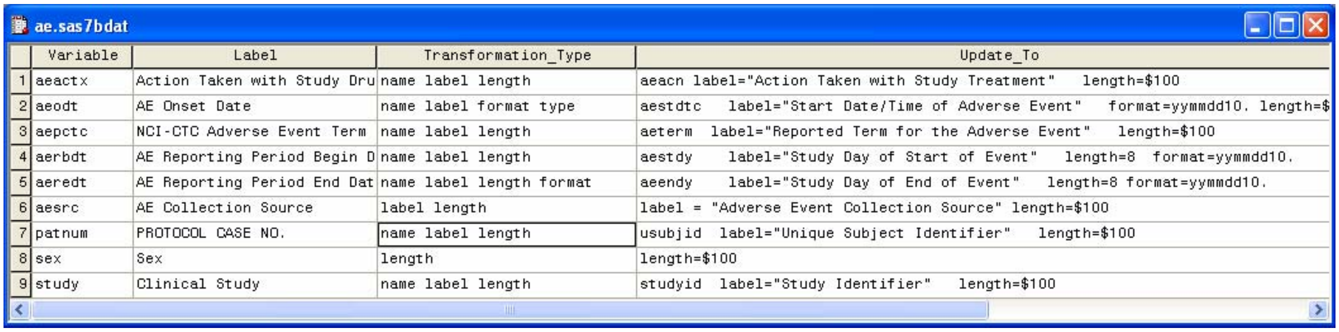
Data name	Variable	Type	Length	Label
ae	race	C	**40***	Race/ethnicity*
Dm	race	C	**100***	Race*
ae	sex	C	**7***	Sex
Dm	sex	C	**100***	Sex
undocfu	visit	N*	**8***	PROG BASED PHYS EXAM*
Tv	visit	C*	**100***	Visit Name*

4. Other findings between CDISC and the source data includes 14 instances where there were the same variables defined as CDISC but the variable types were different.

Variable	Type	Length	Label	Format
undocfu	visit	**N***	8*	PROG BASED PHYS EXAM*
tv	visit	**C***	100*	Visit Name*

These examples are representative of a review that was done on a study. These results are useful to help identify ways of making the data structure more consistent. They can also help identify attributes that may need to be changed to meet target CDISC standards.

STEP 3: Define the transformation specification. This is the result of the review of the source data compared with the destination CDISC data structure. It is also a result of careful review of each variable and how it would fit into the data structure of CDISC. Note that the related records, supplemental qualification and comments are treated separately. This will be described in the next step.



ae.sas7bdat

	Variable	Label	Transformation_Type	Update_To
1	aeactx	Action Taken with Study Dru	name label length	aeacn label="Action Taken with Study Treatment" length=$100
2	aeodt	AE Onset Date	name label format type	aestdtc label="Start Date/Time of Adverse Event" format=yymmdd10. length=$
3	aepctc	NCI-CTC Adverse Event Term	name label length	aeterm label="Reported Term for the Adverse Event" length=$100
4	aerbdt	AE Reporting Period Begin D	name label length	aestdy label="Study Day of Start of Event" length=8 format=yymmdd10.
5	aeredt	AE Reporting Period End Dat	name label length format	aeendy label="Study Day of End of Event" length=8 format=yymmdd10.
6	aesrc	AE Collection Source	label length	label = "Adverse Event Collection Source" length=$100
7	patnum	PROTOCOL CASE NO.	name label length	usubjid label="Unique Subject Identifier" length=$100
8	sex	Sex	length	length=$100
9	study	Clinical Study	name label length	studyid label="Study Identifier" length=$100

The transformation model is captured in a SAS dataset. The variable and label columns identify the source data. The transformation type indicates what attributes are to be transformed. The "Update_to" column is the actual code example of the actual transformation. There are other types of transformations such as transpose and value change that are beyond the scope of this paper. The subset of transformation types described here demonstrates how transformations can be expressed in a series of columns which is referred to as a transformation model.

You can choose to capture the transformation model in an Excel spreadsheet but in this example, the Transdata utility expects the transformation model to be captured in a SAS dataset format.

STEP 4: Identify supplemental qualifiers, relational records and comment fields. The CDISC data model is restrictive in what variables can be included. It however has structures which can capture information from your source data in separate datasets. The definition of these three structures mentioned is explained in more detail in the SDTM definition document found at the website: cdisc.org. Once you have identified these variables, you can then transform your source dataset into these structures.

STEP 5: Develop SAS programs to apply the transformation. This is based on the transformation model which functions as specifications to your transformation. A simple transformation program is shown below:

```
************************************************;
* Program: trans_ae.sas
* Description: Transform Adverse Events data
*              from inlib.ae to outlib.ae
* By: %trans, 06/16/2005,  8:11:57 pm
************************************************;

libname inlib "C:\sample\location\source\data";
libname outlib "C:\sample\location\cdisc\data";

data outlib.ae (label="Adverse Events");
   attrib aesrc label = "Adverse Event Collection Source" length=$100;
   set inlib.ae;
   *** Define a new variable aeacn  to replace old variable: aeactx ***;
   attrib aeacn label="Action Taken with Study Treatment"   length=$100;
   aeacn  = left(trim(aeactx));
   drop aegn;

   *** Define a new variable aestdtc  to replace old variable: aeodt ***;
   attrib aestdtc   label="Start Date/Time of Adverse Event"    length=$100;
   aestdtc  = put(aeodt,yymmdd10. );
   drop aeodt;

   drop aeodtf;
   *** Define a new variable aeterm  to replace old variable: aepctc ***;
   attrib aeterm  label="Reported Term for the Adverse Event"   length=$100;
   aeterm  = left(trim(aepctc));
   *** Define a new variable studyid  to replace old variable: study ***;
   attrib studyid  label="Study Identifier"   length=$100;
   studyid  = left(trim(study));
   drop study;
   drop toxcat;
   drop toxcd;
run;
```

STEP 6: Verify the transformation. Depending on your SOP, this can be formally applied with requirements, functional specifications and test plan. Or it can be a series of SAS programs which confirms the results. Some of the verification tasks include:

Code Review	Systematic review of SAS programs according to a predetermined checklist of verification criteria.
Code Testing	Perform testing on SAS programs or macros supplying valid and invalid inputs and verify expected output.
Log Evaluation	Evaluate the SAS log for error, warning and other unexpected messages.
Output Review	Visual or programmatic review of report outputs as compared to expected results.
Data Review	Review attributes and contents of output data for accuracy and integrity.
Duplicate Programming	Independent programming to produce the same output for comparison.

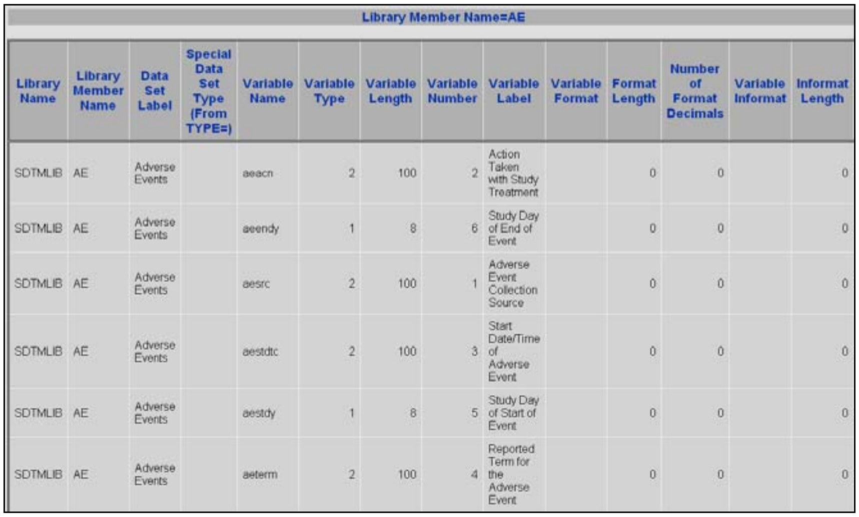
STEP 7: Document the metadata of the transformed data. Metadata is information about the data. This can be attributes of the variables such as variable label and type. It can also include the origin of the variable such as whether it is a source or a derived variable. This is commonly referred to as the DEFINE.PDF file or domain documentation. There are two sections to this documentation. The first lists all the datasets and the second details all the variables for each dataset.

Dataset Name	Location	Keys	Number of Variables	Number of Records	Comment
ae	ae.xpt	usubjid	9	20	
cm	cm.xpt	usubjid	4	35	
co	co.xpt	usubjid	15	42	
dm	dm.xpt	usubjid	7	20	
ds	ds.xpt	usubjid	5	20	

ae (ae.xpt)								
Variable Name	**Type**	**Length**	**Variable Label**	**Format**	**Decode Formats**	**Origins**	**Role**	**Comment**
Aeacn	Character	100	Action Taken with Study Treatment	actfmt.	1 = None 2 = Dose Increase 3 = Dose Decrease	Derived		
Aeendy	Numeric	8	Study Day of End of Event			Derived		

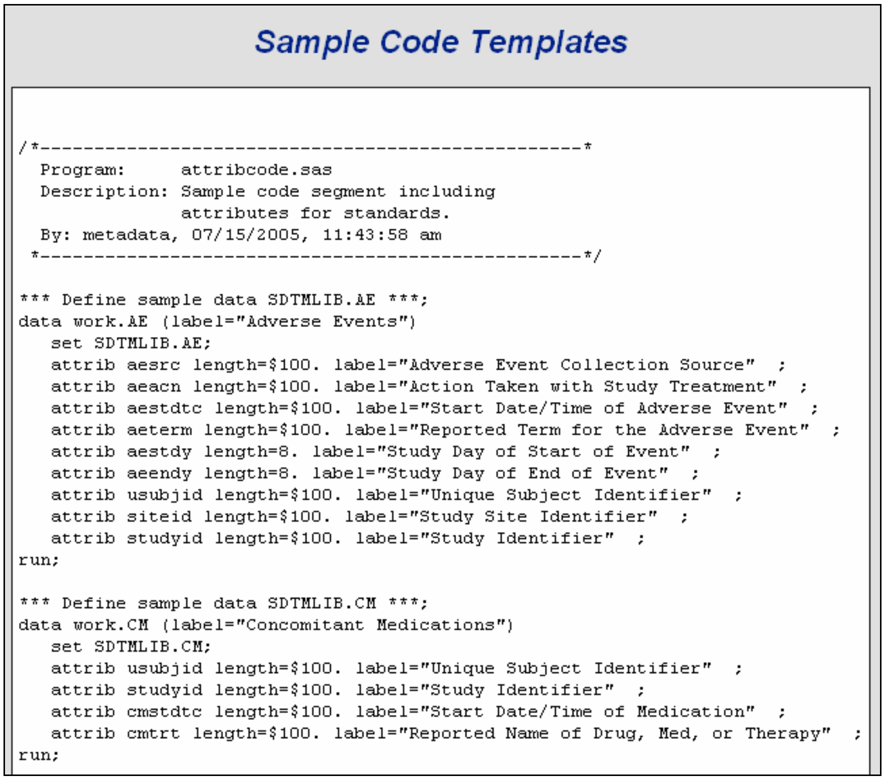
ae (ae.xpt)								
Variable Name	Type	Length	Variable Label	Format	Decode Formats	Origins	Role	Comment
Aesrc	Character	100	Adverse Event Collection Source			Derived		
Aestdtc	Character	100	Start Date/Time of Adverse Event			Derived		
Aestdy	Numeric	8	Study Day of Start of Event			Derived		

STEP 8: Make the data available for users as a standard for future uses. The domain documentation can be published to users. In addition to the domain documentation, a full PROC CONTENTS of the transformed data and template code can also be included to give users a starting point if they need to create the same data standards for their next study.



Library Member Name=AE

Library Name	Library Member Name	Data Set Label	Special Data Set Type (From TYPE=)	Variable Name	Variable Type	Variable Length	Variable Number	Variable Label	Variable Format	Format Length	Number of Format Decimals	Variable Informat	Informat Length
SDTMLIB	AE	Adverse Events		aeacn	2	100	2	Action Taken with Study Treatment		0	0		0
SDTMLIB	AE	Adverse Events		aeendy	1	8	6	Study Day of End of Event		0	0		0
SDTMLIB	AE	Adverse Events		aesrc	2	100	1	Adverse Event Collection Source		0	0		0
SDTMLIB	AE	Adverse Events		aestdtc	2	100	3	Start Date/Time of Adverse Event		0	0		0
SDTMLIB	AE	Adverse Events		aestdy	1	8	5	Study Day of Start of Event		0	0		0
SDTMLIB	AE	Adverse Events		aeterm	2	100	4	Reported Term for the Adverse Event		0	0		0



Sample Code Templates

```
/*---------------------------------------------------*
  Program:     attribcode.sas
  Description: Sample code segment including
               attributes for standards.
  By: metadata, 07/15/2005, 11:43:58 am
 *---------------------------------------------------*/

*** Define sample data SDTMLIB.AE ***;
data work.AE (label="Adverse Events")
   set SDTMLIB.AE;
   attrib aesrc length=$100. label="Adverse Event Collection Source"  ;
   attrib aeacn length=$100. label="Action Taken with Study Treatment"  ;
   attrib aestdtc length=$100. label="Start Date/Time of Adverse Event"  ;
   attrib aeterm length=$100. label="Reported Term for the Adverse Event"  ;
   attrib aestdy length=8. label="Study Day of Start of Event"  ;
   attrib aeendy length=8. label="Study Day of End of Event"  ;
   attrib usubjid length=$100. label="Unique Subject Identifier"  ;
   attrib siteid length=$100. label="Study Site Identifier"  ;
   attrib studyid length=$100. label="Study Identifier"  ;
run;

*** Define sample data SDTMLIB.CM ***;
data work.CM (label="Concomitant Medications")
   set SDTMLIB.CM;
   attrib usubjid length=$100. label="Unique Subject Identifier"  ;
   attrib studyid length=$100. label="Study Identifier"  ;
   attrib cmstdtc length=$100. label="Start Date/Time of Medication"  ;
   attrib cmtrt length=$100. label="Reported Name of Drug, Med, or Therapy"  ;
run;
```

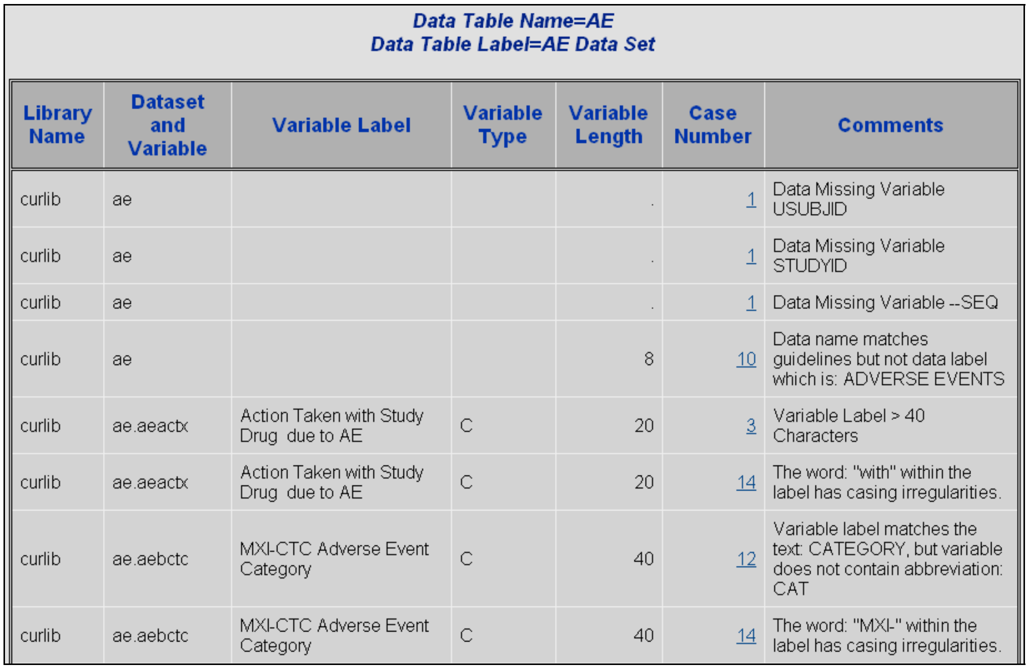
As seen in the eight steps, the actual data transformation portion is only one step. The other accompanying steps are essential in making the transformation accurate and useful.

AUTOMATING TRANSFORMATION

The eight steps described above can be very labor intensive if they are done manually. Some of the tasks are tedious and require great attention to detail since there are many variables and attributes related to a data transformation. Each step will once again be described with the aid of tools such as the CDISC Builder and Transdata to automate certain tasks.

STEP 1: The test plan has to be manually edited. In the example used, an HTML version was created using FrontPage as a tool.

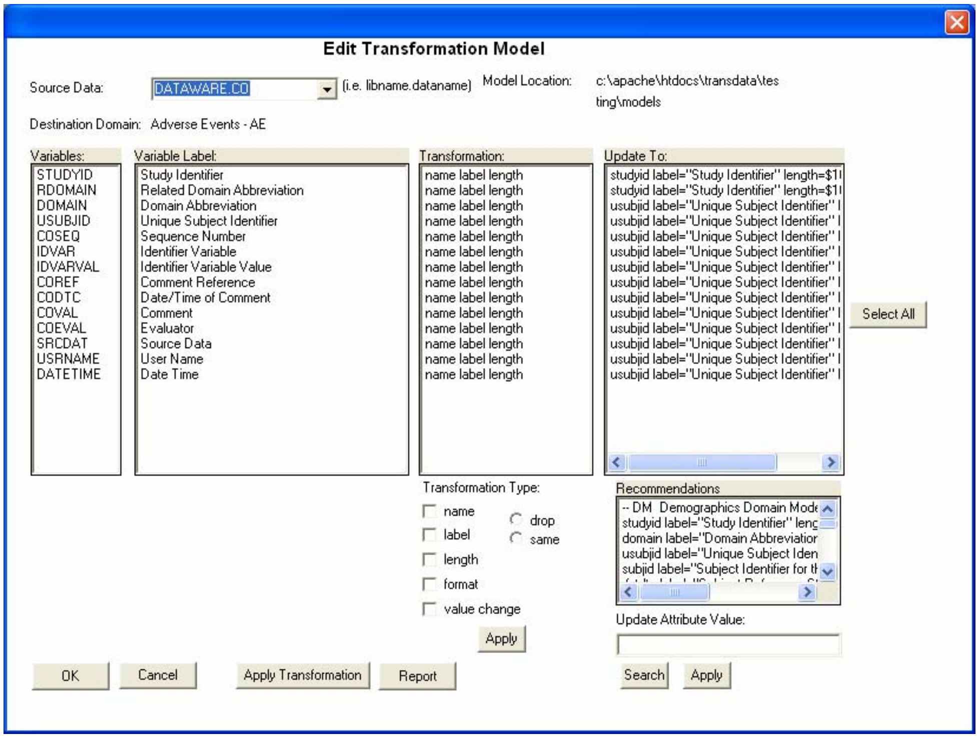
STEP 2: This step included a data review of the existing source structure. The findings in step 2 utilizes %difftest which is a tool provided within CDISC Builder. It evaluates each variable and the associated attributes to determine if there are any differences. This automation captures detailed attributes which would otherwise be missed through visual inspection. In addition, the %cdisc macro within CDISC Builder was used to generate the following report. This helps identify deviations from rules established by CDISC.



Data Table Name=AE
Data Table Label=AE Data Set

Library Name	Dataset and Variable	Variable Label	Variable Type	Variable Length	Case Number	Comments
curlib	ae			.	1	Data Missing Variable USUBJID
curlib	ae			.	1	Data Missing Variable STUDYID
curlib	ae			.	1	Data Missing Variable --SEQ
curlib	ae			8	10	Data name matches guidelines but not data label which is: ADVERSE EVENTS
curlib	ae.aeactx	Action Taken with Study Drug due to AE	C	20	3	Variable Label > 40 Characters
curlib	ae.aeactx	Action Taken with Study Drug due to AE	C	20	14	The word: "with" within the label has casing irregularities.
curlib	ae.aebctc	MXI-CTC Adverse Event Category	C	40	12	Variable label matches the text: CATEGORY, but variable does not contain abbreviation: CAT
curlib	ae.aebctc	MXI-CTC Adverse Event Category	C	40	14	The word: "MXI-" within the label has casing irregularities.

STEP 3: This step defines the transformation specification. The Transdata tool is used to automate the transformation model definition. It captures the variables from the source and then provides a mechanism to populate the transformed attributes through recommendations. The recommendations are derived from the CDISC SDS v3.1 structure.


Edit Transformation Model

Source Data: DATAWARE.CO (i.e. libname.dataname)

Model Location: c:\apache\htdocs\transdata\testing\models

Destination Domain: Adverse Events - AE

Variables:	Variable Label:	Transformation:	Update To:
STUDYID	Study Identifier	name label length	studyid label="Study Identifier" length=$1
RDOMAIN	Related Domain Abbreviation	name label length	studyid label="Study Identifier" length=$1
DOMAIN	Domain Abbreviation	name label length	usubjid label="Unique Subject Identifier" l
USUBJID	Unique Subject Identifier	name label length	usubjid label="Unique Subject Identifier" l
COSEQ	Sequence Number	name label length	usubjid label="Unique Subject Identifier" l
IDVAR	Identifier Variable	name label length	usubjid label="Unique Subject Identifier" l
IDVARVAL	Identifier Variable Value	name label length	usubjid label="Unique Subject Identifier" l
COREF	Comment Reference	name label length	usubjid label="Unique Subject Identifier" l
CODTC	Date/Time of Comment	name label length	usubjid label="Unique Subject Identifier" l
COVAL	Comment	name label length	usubjid label="Unique Subject Identifier" l
COEVAL	Evaluator	name label length	usubjid label="Unique Subject Identifier" l
SRCDAT	Source Data	name label length	usubjid label="Unique Subject Identifier" l
USRNAME	User Name	name label length	usubjid label="Unique Subject Identifier" l
DATETIME	Date Time	name label length	usubjid label="Unique Subject Identifier" l

Transformation Type:

□ name
□ label
□ length
□ format
□ value change

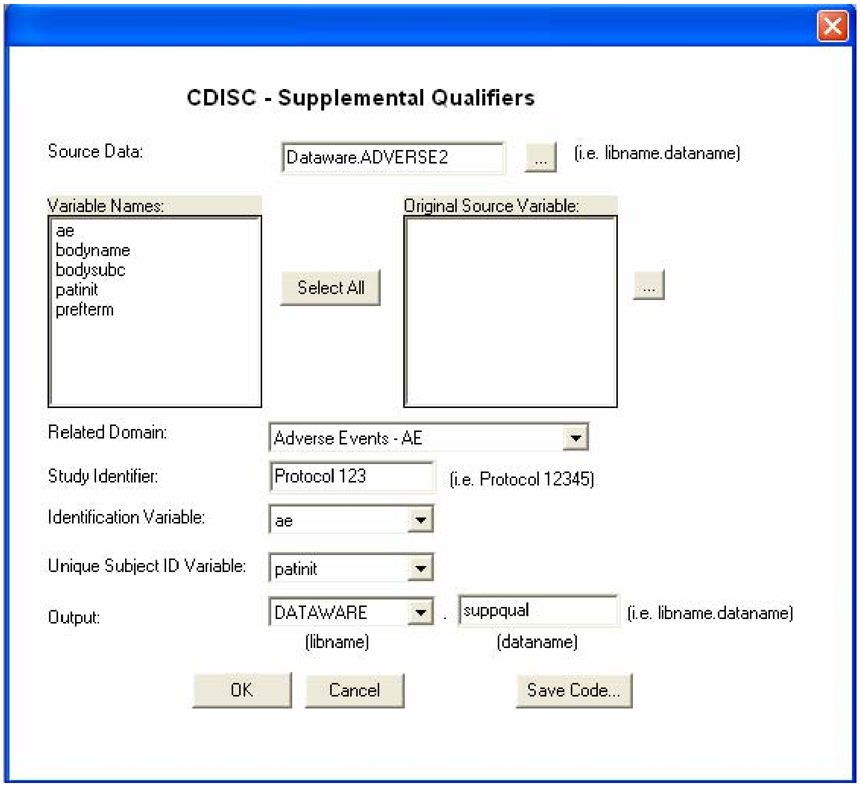
○ drop
○ same

Recommendations

-- DM Demographics Domain Mode
studyid label="Study Identifier" lenc
domain label="Domain Abbreviatior
usubjid label="Unique Subject Iden
subjid label="Subject Identifier for tl

Update Attribute Value:

STEP 4: This step identifies supplemental qualifiers, relational records and comment fields and then imports them into the new structure defined by CDISC. The CDISC Builder tools automate this process by helping you identify which fields pertain to related records. Once identified, it would transpose your source data into the specified relational record data structure as defined by CDISC. It also has similar tools for the supplemental qualifier and comment fields.


CDISC - Supplemental Qualifiers

Source Data: Dataware.ADVERSE2 (i.e. libname.dataname)

Variable Names:

ae
bodyname
bodysubc
patinit
prefterm

Original Source Variable:

Related Domain: Adverse Events - AE

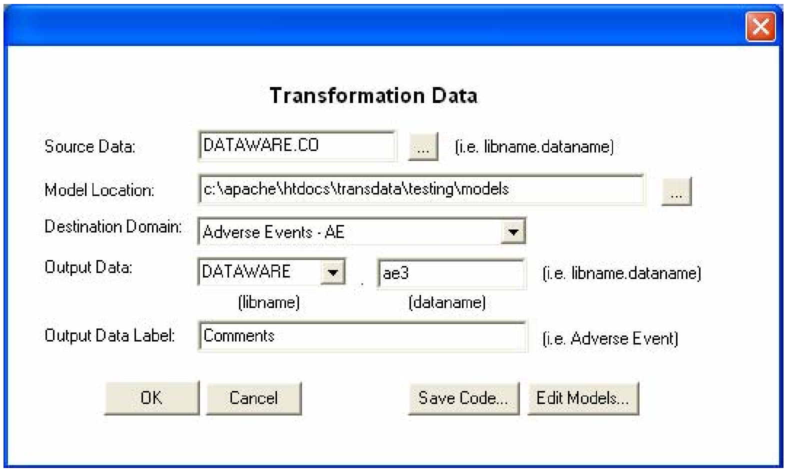
Study Identifier: Protocol 123 (i.e. Protocol 12345)

Identification Variable: ae

Unique Subject ID Variable: patinit

Output: DATAWARE . suppqual (i.e. libname.dataname)
(libname) (dataname)

STEP 5: This step generates the SAS programs used to apply the transformation. The Transdata automates this step by reading the transformation model and applying these rules to generate the transformation code.



Transformation Data

Source Data: DATAWARE.CO (i.e. libname.dataname)

Model Location: c:\apache\htdocs\transdata\testing\models

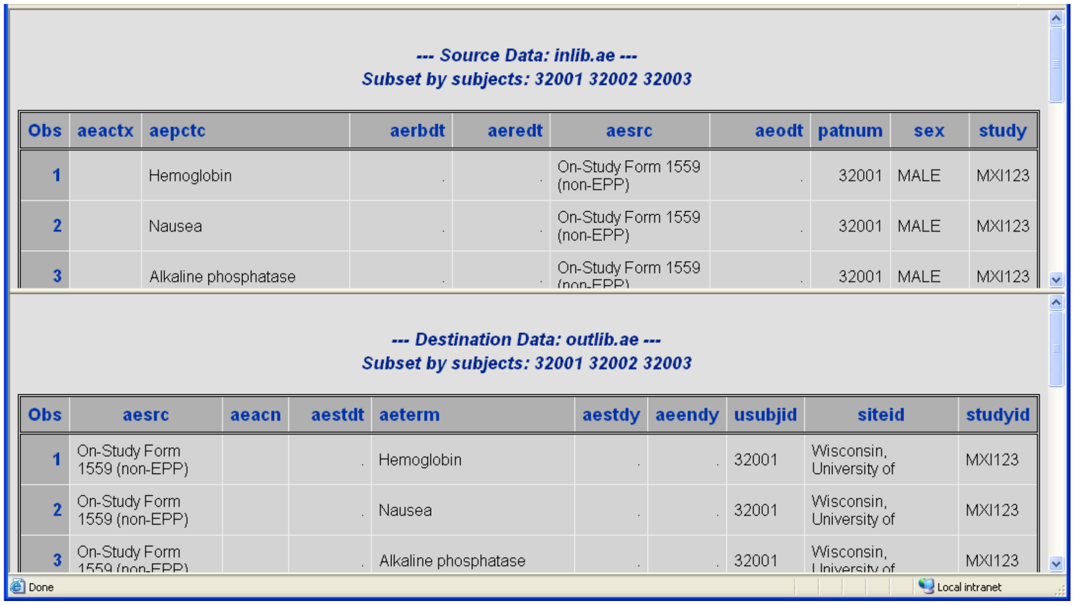
Destination Domain: Adverse Events - AE

Output Data: DATAWARE (libname) . ae3 (dataname) (i.e. libname.dataname)

Output Data Label: Comments (i.e. Adverse Event)

This handles most of the transformation needs. However, for more customized transformations, the code can be saved and edited to fulfill the requirements of any specific transformation requirements.

STEP 6: This step verifies the transformation. The Transdata tool assists in the verification task by generating reports to help verify the source against the transformed data. The verification report is an HTML report which has a dual frame screen with a subset of three subjects for fast review.



--- Source Data: inlib.ae ---
Subset by subjects: 32001 32002 32003

Obs	aeactx	aepctc	aerbdt	aeredt	aesrc	aeodt	patnum	sex	study
1		Hemoglobin	.	.	On-Study Form 1559 (non-EPP)	.	32001	MALE	MXI123
2		Nausea	.	.	On-Study Form 1559 (non-EPP)	.	32001	MALE	MXI123
3		Alkaline phosphatase	.	.	On-Study Form 1559 (non-EPP)	.	32001	MALE	MXI123

--- Destination Data: outlib.ae ---
Subset by subjects: 32001 32002 32003

Obs	aesrc	aeacn	aestdt	aeterm	aestdy	aeendy	usubjid	siteid	studyid
1	On-Study Form 1559 (non-EPP)		.	Hemoglobin	.	.	32001	Wisconsin, University of	MXI123
2	On-Study Form 1559 (non-EPP)		.	Nausea	.	.	32001	Wisconsin, University of	MXI123
3	On-Study Form 1559 (non-EPP)		.	Alkaline phosphatase	.	.	32001	Wisconsin, University of	MXI123

This allows visual inspection of the "before" and "after" to verify if the transformation is being performed correctly. In addition to this PROC PRINT report, there is also a PROC FREQ for categorical variables. The frequency report helps to verify values in a summarized manner to help identify potential discrepancies in the transformation.

STEP 7: This step makes the data available for users as a standard for future uses. The CDISC builder captures all the metadata using PROC CONTENTS and stores this in a centralized database. It then generates a report in HTML format along with template code. This can be published on an intranet or emailed to users so that they can use the same standard on their next study. The metadata database also has search capabilities for users to easily find attributes among the large set of metadata that has been transformed.

There are many efforts that are needed to perform data transformation such as the CDISC example described in this section. Automating each step can help ensure that the transformation is done with accuracy. It also saves time since it elevates the tedium of many small but labor intensive tasks.

CONCLUSION

Data transformation is an essential and unavoidable task when working with clinical trials data. It can be used to move towards a new data standard such as CDISC or an internal standard used for an integrated safety summary. Accomplishing standards leads to portability of data between studies, while also increasing the mobility of team members between projects. It also gives you capabilities of performing analysis across studies that would not be available otherwise. Performing a successful transformation of data from one structure to another is not limited to just programming the transposition. There are other steps in managing and verifying the transformation to ensure the integrity and accuracy of the transformed data. These tasks can be resource intensive and time consuming if done manually. Automation tools can add value in removing the tedium while increasing efficiency and accuracy of the work. This can ultimately be the difference between a time consuming failed project and one of accuracy and success.

REFERENCES

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

CDISC Builder, Transdata and all other Meta-Xceed, Inc. product names are registered trademarks of Meta-Xceed, Inc. in the USA.

Other brand and product names are registered trademarks or trademarks of their respective companies.

ABOUT THE AUTHOR

Sy Truong is a Systems Developer for Meta-Xceed, Inc. They may be contacted at:

Sy Truong

48501 Warm Springs Blvd. Ste 117

Fremont, CA 94539

(510) 226-1209

sy.truong@meta-x.com