PharmaSUG 2012 - Paper CC03

Automatic Detection and Identification of Variables with All Missing Values in SDTM/ADaM Datasets for FDA Submission

Min Chen, Vertex Pharmaceuticals, Cambridge, MA
Xiangchen (Bob) Cui, Vertex Pharmaceuticals, Cambridge, MA

ABSTRACT

When submitting clinical study data in electronic format to the FDA, it is preferable to submit as few as possible unnecessary variables which have all missing values. This kind of variable is called empty variable. It is highly desirable to automate the process of detecting and identifying empty variables in the submitted datasets. Handling these identified empty variables is very critical for FDA submission. CDISC introduced a concept of core variable in an SDTM domain and an ADaM dataset. A variable is categorized as **Required**, **Expected**, and **Permissible** in an SDTM domain and as **Required**, **Conditionally Required**, and **Permissible** in an ADaM dataset. Applying the information of core variable categories to these empty variables provides a better decision to handle these empty variables in an FDA submission. Hence it ensures the technical accuracy and submission quality.

This paper introduces a SAS macro to automatically detect and identify empty variables in SDTM domains and ADaM datasets. Five scenarios of empty variables are illustrated and their corresponding resolutions are provided to the readers as a reference based on the information of core variable categories in SDTM/ADaM datasets.

INTRODUCTION

Submitting clinical study data to the FDA with variables having all missing values should be avoided as they provide little information about the study to the Agency. This kind of variable is called empty variable. It is highly desirable to automatically detect and identify empty variables at any stage before the submission in order to ensure the submission quality.

The concept of core variable defined in CDISC SDTM and ADaM can be used to handle the empty variables in an FDA submission. CDISC Submission Data Standards Team categorizes SDTM variables as **Required**, **Expected**, and **Permissible**. A required variable must be included in the dataset and cannot be null for any record. An expected variable may contain some null values, and an empty variable should still be included in the dataset when no data has been collected for an expected variable. A permissible variable should be used in a domain as appropriate when collected or derived. The sponsor can decide whether an empty permissible variable should be included in the submitted dataset. FDA CDER suggests that all permissible variables for which data were collected or for which derivations are possible should be submitted for traceability. The examples include: baseline flags, EPOCH designators, and --DY or --STDY variables.

Similarly, three core variable categories are defined for ADaM variables as **Required**, **Conditionally Required**, and **Permissible**. A required variable must be included in the datasets. A conditionally required variable must be included in the dataset in certain circumstances, and a permissible variable is not required to be included in the datasets. Unless otherwise specified, missing values are allowed in all ADaM variables. Therefore, it is appropriate to have empty variables for required ADaM variables or conditionally required ADaM variables, and appropriate to drop any empty permissible variables.

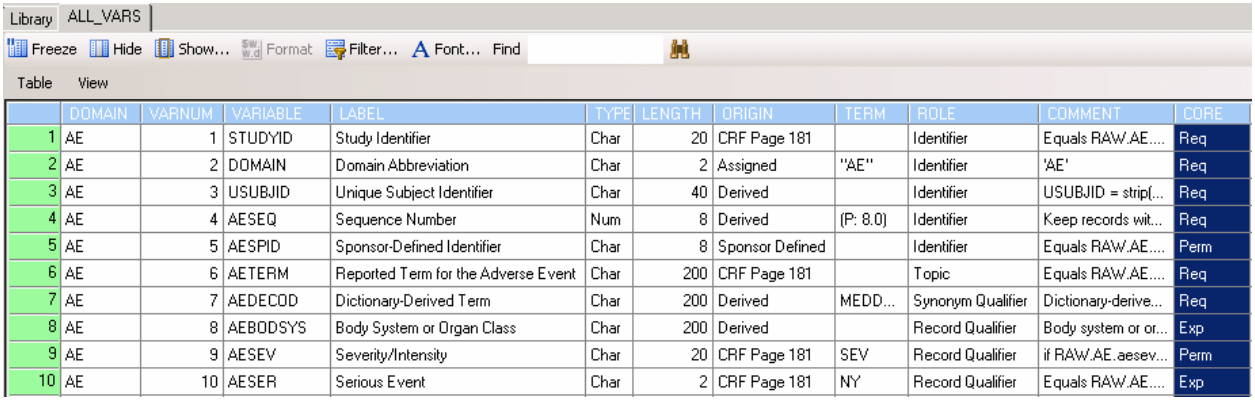
In this paper, the automatic detection and identification of the empty variables will be combined with the information of core variable categories of SDTM/ADaM to better handle them for an FDA submission, for an empty variable would not be necessarily dropped from FDA submission. A macro **%emtpty_var_checking** is called to automatically detect the empty variables in SDTM or ADaM domain(s). A report of empty variables will be generated from the macro. It lists domain name, variable names, information of core variable categories, and comments for the possible resolution to the user for a better decision. An error message will be output for SDTM required variables if they are empty. Five scenarios of the empty variables are illustrated and their corresponding resolutions are provided in the paper to the readers as a reference. Reviewing the report of empty variables is another validation of SDTM or ADaM programming to double check the variables with missing values. Hence the macro provides the user another validation tool.

Since the automation of empty variable checking is conducted from the beginning of ADaM programming to the end of FDA submission, the high quality of the submissions can be achieved in a cost-effective and efficient way.

AN INTRODUCTION OF PROGRAMMING SPECIFICATION FOR SDTM/ADAM DATASET(S)

An individual programming specification for SDTM is in Excel format®, and an individual programming specification for ADaM is in MS Word® format in our SDTM and ADaM programming process, respectively. It will be retrieved by

the in-house Macros during SDTM or ADaM Programming, and a SAS data named ALL_VARS will restore the variable information from each programming specification. Example SAS datasets from an SDTM specification and an ADaM specification are shown in Display 1 and Display 2 respectively.



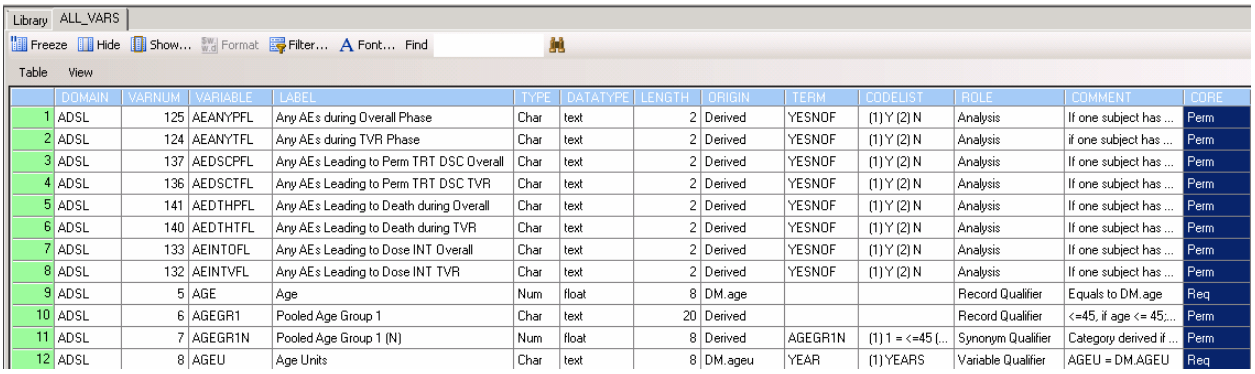
Library ALL_VARS

Freeze Hide Show... Format Filter... Font... Find

Table View

	DOMAIN	VARNUM	VARIABLE	LABEL	TYPE	LENGTH	ORIGIN	TERM	ROLE	COMMENT	CORE
1	AE	1	STUDYID	Study Identifier	Char	20	CRF Page 181		Identifier	Equals RAW.AE....	Req
2	AE	2	DOMAIN	Domain Abbreviation	Char	2	Assigned	"AE"	Identifier	'AE'	Req
3	AE	3	USUBJID	Unique Subject Identifier	Char	40	Derived		Identifier	USUBJID = strip(...	Req
4	AE	4	AESEQ	Sequence Number	Num	8	Derived	(P: 8.0)	Identifier	Keep records wit...	Req
5	AE	5	AESPID	Sponsor-Defined Identifier	Char	8	Sponsor Defined		Identifier	Equals RAW.AE....	Perm
6	AE	6	AETERM	Reported Term for the Adverse Event	Char	200	CRF Page 181		Topic	Equals RAW.AE....	Req
7	AE	7	AEDECOD	Dictionary-Derived Term	Char	200	Derived	MEDD...	Synonym Qualifier	Dictionary-derive...	Req
8	AE	8	AEBODSYS	Body System or Organ Class	Char	200	Derived		Record Qualifier	Body system or or...	Exp
9	AE	9	AESEV	Severity/Intensity	Char	20	CRF Page 181	SEV	Record Qualifier	if RAW.AE.aesev...	Perm
10	AE	10	AESER	Serious Event	Char	2	CRF Page 181	NY	Record Qualifier	Equals RAW.AE....	Exp

Display 1. SDTM Programming Specifications in SAS Format



Library ALL_VARS

Freeze Hide Show... Format Filter... Font... Find

Table View

	DOMAIN	VARNUM	VARIABLE	LABEL	TYPE	DATATYPE	LENGTH	ORIGIN	TERM	CODELIST	ROLE	COMMENT	CORE
1	ADSL	125	AEANYPFL	Any AEs during Overall Phase	Char	text	2	Derived	YESNOF	(1) Y (2) N	Analysis	If one subject has ...	Perm
2	ADSL	124	AEANYTFL	Any AEs during TVR Phase	Char	text	2	Derived	YESNOF	(1) Y (2) N	Analysis	if one subject has ...	Perm
3	ADSL	137	AEDSCPFL	Any AEs Leading to Perm TRT DSC Overall	Char	text	2	Derived	YESNOF	(1) Y (2) N	Analysis	If one subject has ...	Perm
4	ADSL	136	AEDSCTFL	Any AEs Leading to Perm TRT DSC TVR	Char	text	2	Derived	YESNOF	(1) Y (2) N	Analysis	If one subject has ...	Perm
5	ADSL	141	AEDTHPFL	Any AEs Leading to Death during Overall	Char	text	2	Derived	YESNOF	(1) Y (2) N	Analysis	If one subject has ...	Perm
6	ADSL	140	AEDTHTFL	Any AEs Leading to Death during TVR	Char	text	2	Derived	YESNOF	(1) Y (2) N	Analysis	If one subject has ...	Perm
7	ADSL	133	AEINTOFL	Any AEs Leading to Dose INT Overall	Char	text	2	Derived	YESNOF	(1) Y (2) N	Analysis	If one subject has ...	Perm
8	ADSL	132	AEINTVFL	Any AEs Leading to Dose INT TVR	Char	text	2	Derived	YESNOF	(1) Y (2) N	Analysis	If one subject has ...	Perm
9	ADSL	5	AGE	Age	Num	float	8	DM.age			Record Qualifier	Equals to DM.age	Req
10	ADSL	6	AGEGR1	Pooled Age Group 1	Char	text	20	Derived			Record Qualifier	<=45, if age <= 45;...	Perm
11	ADSL	7	AGEGR1N	Pooled Age Group 1 (N)	Num	float	8	Derived	AGEGR1N	(1) 1 = <=45 (...	Synonym Qualifier	Category derived if ...	Perm
12	ADSL	8	AGEU	Age Units	Char	text	8	DM.ageu	YEAR	(1) YEARS	Variable Qualifier	AGEU = DM.AGEU	Req

Display 2. ADaM Programming Specifications in SAS Format

The programming specifications will be applied to both checking of empty variables and decision making on handling of empty variables based on the CORE column.

A MACRO FOR AUTOMATIC DETECT AND IDENTIFY EMPTY VARIABLES

A macro to detect and identify empty variables is developed to facilitate the finalization of SDAM and ADaM programming, and further ensure the submission quality. It can be performed at any stage of the programming cycle.

Macro **%empty_var_checking** calculates the number of observations with missing value for each variable. If the count is equal to the number of the observations in the dataset, then the variable will be flagged as the empty variable. A report will be generated for all the empty variables which include error messages for SDTM required variables or selected ADaM required variables, warning messages for SDTM expected variables or ADaM required/conditionally required variables, and warning messages for SDTM permissible variables or ADaM permissible variables.

The macro call of %empty_var_checking is as follows.

```
%macro empty_var_checking(cdisc   = ,    /* SDTM or ADaM                          */
                          specdir = ,    /* a folder for programming specs.       */
                          datadir = ,    /* a folder for SDTM or ADaM datasets    */
                          domain  = _ALL_/*SDTM or ADaM domain name for checking */
                         );
```

Where,

CDISC: SDTM or ADaM.

SPECDIR: Full Path for SDTM /ADaM Programming Specifications.

DATADIR: Full Path for SDTM /ADaM datasets.

DOMAIN: An SDTM /ADaM domain, If assigned _ALL_, or blank, all SDTM /ADaM domains will be checked.

Empty variable checking can be performed during the development of individual SDTM/ADaM dataset by assigning an SDTM/ADaM dataset name to the macro variable &DOMAIN.

The following is some notes about the logic of the macro.

1. Get all variable names for selected domain(s) from the contents of SDTM/ADaM dataset(s).

```
*** get the variables from specifications;
data specs_allvars;
     set speclib.all_vars;
     %if %upcase(&domain.) ne _ALL_ %then where domain = upcase("&domain.");;
run;
proc sort data = specs_allvars; by domain variable; run;

*** get the variables from dataset;
proc contents data=datalib.&domain. noprint
              out=data_allvars(rename=(memname=domain name=variable));
run;
proc sort data=data_allvars; by domain variable; run;
data allvars;
     merge specs_allvars(in=a keep=domain variable) data_allvars(in=b);
     by domain variable;
     if a;
run;
data _null_;
     set allvars(keep=domain variable) end=done;
     by domain variable;
     retain numdomain numvar 0;
     if first.domain then do;
        numdomain + 1; numvar = 0;
        call symput('dsn'||strip(put(numdomain,best.)),strip(domain));
     end;
     numvar + 1;
    call symput('var_'||strip(put(numdomain,best.))||'_'||strip(put(numvar,best.)),
                strip(variable));
 call symput('varmfl_'||strip(put(numdomain,best.))||'_'||strip(put(numvar,best.)),
                'mfl_'||strip(variable));
     if last.domain then call symput('numvar'||strip(put(numdomain,best.)),
                strip(put(numvar,best.)));
     if done then do; call symput('numdomain',strip(put(numdomain,best.))); end;
run;
```

2. Missing records for each variable are counted and empty variables (if the number of missing records is equal to the total number of the observations) are output as vertical structure.

```
data all;if 0;run;
%do i=1 %to &numdomain.;
    data &&dsn&i;
         set datalib.&&dsn&i end=eof;
         retain %do j=1 %to &&numvar&i.;&&varmfl_&i._&j. %end; 0;
         %do j=1 %to &&numvar&i.;
             if missing (&&var_&i._&j.) then &&varmfl_&i._&j. + 1;
         %end;
         if eof then do; nobs = _n_; output; end;
    run;
    data _tmp;
         set &&dsn&i;
         %do j=1 %to &&numvar&i.;
             domain = "&&dsn&i";
             variable = "&&var_&i._&j.";
             nmissing = &&varmfl_&i._&j.;
             if nmissing = nobs then output;
         %end;
```

```
          keep domain variable nmissing nobs;
      run;
      data all; set all _tmp; run;
  %end;
```

3. Reports are generated based on empty variables in SDTM or ADaM datasets, and the core variable categories of the variables.

```
data final;
     merge all(in=a) specs_allvars(keep=domain variable varnum label core);
     by domain variable;
     if a;
     length COMMENT $100;
     source = "&CDISC.";
     %if %upcase(&CDISC) = SDTM %then %do;
        if core = 'Req' then do; coren = 1;
           comment='Error: Required Variable is Empty. Correct/Check the SAS
                    Program!'; end;
        if core = 'Exp' then do; coren=2;
           comment='Warning:Expected Variable is Empty.Check the SAS Program!';
        end;
        if core = 'Perm' then do; coren = 3;
           comment='Warning: Permissible Variable is Empty. Delete it?'; end;
     %end;
     %if %upcase(&CDISC) = ADAM %then %do;
        if core = 'Req' and variable in ('STUDYID', 'SITEID', 'USUBJID', 'SUBJID',
           'PARAMCD', 'PARAM', 'SEX', 'RACE', 'ARM', 'COUNTRY') then do; coren = 1;
           comment='Error: Required Variable is Empty. Correct/Check SAS Program!';
        end;
        else if core = 'Req' then do; coren = 2;
             comment='Warning: Required Variable is Empty. Check the SAS Program!';
        end;
        if core = 'Cond' then do; coren = 3;
           comment='Warning:Conditionally Required Variable is Empty.Check the SAS
                    Program!'; end;
        if core = 'Perm' then do; coren = 4;
           comment='Warning: Permissible Variable is Empty. Delete it?';
        end;
     %end;
run;
proc sort data = final;by domain coren varnum;run;
```

Display 3 and 4 shows reports of empty variables for different CORE categories in SDTM or ADaM datasets respectively.

The Following Variables in SDTM Have All Missing Values

Domain	Variable Order	Variable	Variable Label	Total Number of Observations	Core	Comment
DM	18	COUNTRY	Country	62	Req	Error: Required Variable is Empty. Correct/Check the SAS Program!
DS	12	DSDTC	Date/Time of Collection	1368	Perm	Warning: Permissible Variable is Empty. Delete it?
LB	26	LBTOX	Toxicity	50684	Perm	Warning: Permissible Variable is Empty. Delete it?
	27	LBTOXGR	Standard Toxicity Grade	50684	Perm	Warning: Permissible Variable is Empty. Delete it?
SUPPDM	4	IDVAR	Identifying Variable	1019	Exp	Warning: Expected Variable is Empty. Check the SAS Program!
	5	IDVARVAL	Identifying Variable Value	1019	Exp	Warning: Expected Variable is Empty. Check the SAS Program!
SUPPMH	10	QEVAL	Evaluator	4680	Exp	Warning: Expected Variable is Empty. Check the SAS Program!

Display 3. A Report of Empty Variables with Different CORE Variable Categories in SDTM Datasets

The Following ADaM Variables in Study xxx Have All Missing Values

Domain	Variable Order	Variable	Variable Label	Total Number of Observations	Core	Comment
ADAE	73	DCRASHFL	Discontinuation due to Rash SSC	561	Perm	Warning: Permissible Variable is Empty. Delete it?
	74	DCPRURFL	Discontinuation due to Pruritus SSC	561	Perm	Warning: Permissible Variable is Empty. Delete it?
	76	DCANORFL	Disc. due to Anorectal Disorder	561	Perm	Warning: Permissible Variable is Empty. Delete it?
	77	DCINJSFL	Disc. due to Injection Site Reaction	561	Perm	Warning: Permissible Variable is Empty. Delete it?
ADSL	15	COUNTRY	Country	62	Req	Error: Required Variable is Empty. Correct/Check SAS Program!
	36	TRT01P	Planned Treatment for Period 01	62	Req	Warning: Required Variable is Empty. Check the SAS Program!
	46	TRTSDT	Date of First Exposure to Treatment	62	Cond	Warning: Conditionally Required Variable is Empty. Check the SAS Program!

Display 4. A Report of Empty Variables with Different CORE Variable Categories in ADaM Datasets

DECISION MAKING ON THE EMPTY VARIABLES

There are 5 scenarios of empty SDTM/ADaM variables.

1. Empty SDTM Required Variables or Empty Specially-defined ADaM Required Variables, such as USUBJID, STUDYID, SEX, COUNTRY, and etc.

SDTM required variables and specially-defined ADaM required variables such as USUBJID, STUDYID, SEX, COUNTRY, and etc. cannot be null for any record. If they are empty, the SAS programs must be double checked to address the issue. Here is an example we encountered in a real project: the information of COUNTRY was not collected in eCRF, so that the required variable COUNTRY in DM domain was empty as shown in Display 3, and the required variable COUNTRY in ADSL dataset was empty as shown in Display 4. After the investigation, the variable was populated from the protocol in SDTM Mapping as the study was conducted in a single country.

2. Empty ADaM Required Variables Other Than USUBJID, SITEID, SEX, COUNTRY, and etc.

Null values are allowed for these ADaM required variables. Therefore, empty variables are acceptable. Nevertheless, a warning message is generated for these variables for programmers to check and make sure that the empty variables are not due to programming errors. Action will be taken to correct and check ADaM SAS programs only if there are any programming errors. An example can be seen in a Virology Follow-up Study where there is no investigational product provided, therefore TRT01P in ADSL is all missing as shown in Display 4, and no action will be taken since the missing of the variable is due to the study design, not programming errors.

3. Empty SDTM Expected Variables

Empty SDTM expected variables are allowed for SDTM datasets. Nevertheless, a warning message is generated for these variables for programmers to double check and make sure that the empty variables are not due to programming errors. If a programming error is found out, action must be taken to correct SDTM SAS programs. In Display 3, expected variables IDVAR and IDVARVAL are all missing in SDTM SUPPDM domain, which is one of the CDISC SDTM rules to follow. No action would be taken for this case. On the contrary, expected variable QEVAL was all missing in SDTM SUPPMH domain while it should be populated as “PRINCIPAL INVESTIGATOR” for QORIG = ‘CRF’, instead of missing. SDTM mapping program for MH would be corrected to incorporate this rule.

4. Empty ADaM Conditionally Required Variables

All empty ADaM conditionally required variables are allowed for ADaM datasets. Nevertheless, a warning message is generated for these variables for programmers to double check and make sure the empty variables are not due to programming errors. If a programming error is found out, action will be taken to correct ADaM SAS programs. In Display 4 TRTSDT in ADSL is missing for all, for a study where there is no investigational product, it is unnecessary to keep this variable in the dataset. This variable will be dropped from ADaM dataset ADSL.

5. Empty SDTM/ADaM Permissible Variables

It is recommended that permissible variables should not be included in the data set if the information was not collected. If the information for SDTM/ADaM permissible variables is specified in CRF, but it is never collected in the study, these variables should be included in the SDTM/ADaM datasets for the traceability. If a derived variable or a variable not specified in CRF is empty, such as DS.DSDTC in Display 3, the permissible variable should be dropped from the SDTM/ADaM datasets. Variables which are collected in CRF or needed in the analysis, such as

ADAE.DCRASHFL, ADAE.DCPRURFL, ADAE.DCANORFL, and ADAE.DCINJSFL in Display 4, should be submitted to FDA even though they are empty variables due to no occurrence in the study.

A summary of these 5 scenarios is shown in Table 1.

#	Scenario	Condition	Action Taken
1	Empty SDTM Required Variables; Empty Specially-defined ADaM Required Variables	Any Program Errors	Correct SAS Programs
		No Program Errors	Describe Rationale for Data Oddities in Reviewer Guide
2	Empty ADaM Required Variables Other Than USUBJID, SITEID, SEX, COUNTRY, and etc.	Any Program Errors	Correct ADaM SAS Programs
		No Program Errors	Keep the Variable for Submission, and Document in Reviewer Guide
3	Empty SDTM Expected Variables	Any Program Errors	Correct SDTM SAS Programs
		No Program Errors	Keep the Variable for Submission, and Document in Reviewer Guide
4	Empty ADaM Conditionally Required Variables	Any Program Errors	Correct ADaM SAS Programs
		No Program Errors, Needed in Analysis	Keep the Variable for Submission, and Document in Reviewer Guide
		No Program Errors, Not Needed in Analysis	Drop the Variable
5	Empty SDTM/ADaM Permissible Variables	A Variable Derived or Not Specified in CRF	Drop the Variable
		A Variable Collected or Needed in Analysis	Keep the Variable for Submission, and Document in Reviewer Guide

Table 1. Summary of 5 Scenarios of Empty SDTM/ADaM Variables

DOCUMENTATION OF EMPTY VARIABLES IN REVIEWER GUIDE

If the final ADaM Datasets still contains empty variables, the rationale to keep these empty variables in the ADaM datasets should be explained in the reviewer guide for FDA reviewers. An example of rationale to keep empty variables in ADAE dataset is shown in Table 2.

Domain	Variable Order	Variable	Variable Label	Core	Comment
ADAE	73	DCRASHFL	Discontinuation due to Rash SSC	Perm	The event did not occur in the study, but the variable is needed for analysis.
	74	DCPRURFL	Discontinuation due to Pruritus SSC	Perm	The event did not occur in the study, but the variable is needed for analysis.
	76	DCANORFL	Disc. due to Anorectal Disorder	Perm	The event did not occur in the study, but the variable is needed for analysis.
	77	DCINJSFL	Disc. due to Injection Site Reaction	Perm	The event did not occur in the study, but the variable is needed for analysis.

Table 2. The Rationale to Keep Empty Variables in ADaM Datasets – in Reviewer Guide

CONCLUSION

In summary, this paper introduces an efficient macro-based tool to automatically detect and identify the SDTM/ADaM variables with all missing values. This tool is handy and easy to use. Based on the information of CORE variable categories defined in CDISC SDTM and ADaM better decisions can be made for finalizing the SDTM/ADaM programming for FDA submission. The macro provides the user another validation tool of SDTM or ADaM programming. Since it can be used at any stage of the programming cycle, the submission quality can be improved in a cost-effective way. We hope this tool can help you make better decisions on SDTM/ADaM variables for an FDA submission.

REFERENCES

CDISC Submission Data Standards Team. “Study Data Tabulation Model Implementation Guide: Human Clinical Trials”, November 2008. http://www.cdisc.org/sdtm

CDISC Analysis Data Model Team. “Analysis Data Model (ADaM) Implementation Guide”. December 2009. http://www.cdisc.org/adam

“CDER Common Data Standards Issues Document”, May 2011. http://www.fda.gov/downloads/Drugs/DevelopmentApprovalProcess/FormsSubmissionRequirements/ElectronicSubmissions/UCM254113.pdf

Ellen Xiao. (2010) ”SDTM Attribute Checking Tool”, SAS Global Forum 2010.

Xiangchen (BoB) Cui, Min Chen, and Tathabbai Pakalapati. “An Innovative ADaM Programming Tool for FDA Submission”, PharmaSUG, May 2012

ACKNOWLEDGEMENTS

Appreciation goes to Kelly Blackburn, Stacy Surensky and Tathabbai Pakalapati for their review and comments.

CONTACT INFORMATION

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

Name: Min Chen**,** Ph.D.
Enterprise: Vertex Pharmaceuticals, Inc.
Address: 88 Sidney Street
City, State ZIP: Cambridge MA, 02139
Work Phone: 617-444-7134
Fax: 617-460-8060
E-mail: min_chen@vrtx.com

Name: Xiangchen (Bob) Cui**,** Ph.D.
Enterprise: Vertex Pharmaceuticals, Inc.
Address: 88 Sidney Street
City, State ZIP: Cambridge MA, 02139
Work Phone: 617-444-6069
Fax: 617-460-8060
E-mail: xiangchen_cui@vrtx.com