Validation of CDISC specifications using VALSPEC utility macro
Hrideep Antony, Syneos Health, Morrisville, North Carolina, USA
Aman Bahl, Syneos Health, Ontario, Canada

ABSTRACT
The define.xml is the cover section of the electronic common technical document (eCTD) submission and it provides a high-level summary of the metadata for all the data submitted for an NDA (new drug application). Define.xml will not only aid in the ease of regulatory review but also conveys a message on the overall quality of the submitted work to the reviewers and a good quality one can augment their trust in the results that are being submitted. Define.xml document quality is widely driven by the data and the specifications that are used to generate the file.

The accuracy of the SDTM/ADaM specifications is determined by parameters such as data compliance and the usage of controlled terminology. However, the majority of the discrepancies are only identified after the pinnacle validation. Specification updates based on the compliance findings while creating the final CRT package may result in post-lock updates to the SDTM/ADaM and TFLs. These post-lock updates can cause significant submission delays.

The process of identifying the issues in the specification document does not have to wait that long! This paper introduces an approach and a utility that will alert the author about the issues at an early stage of the creation of the data transformation process. Identifying the issues at an early stage will not only improve the quality of the overall submission but will also reduce the rework that is required and further ease the submission process.

INTRODUCTION
The lifecycle of the NDA process begins with the data collection stage to the submission stage as shown in figure 1 below. Most of these stages are well regulated by predetermined standards set by the regulatory agency to facilitate the streamlining of the approval process. Non-conformance and deviations to the set compliance standards at any stage will have a ripple effect on the overall compliance requirements, resulting in undue delays in the approval process. Such deviations from the standards if identified at an early stage will avoid any undue revisions and rework.

![Figure 1. Drug submission process lifecycle](image-url)
VALSPEC utility that is introduced in this paper is used to validate SDTM and ADaM specifications along with the data and streamline standard implementation at an early data transformation stage. The mission of this utility is to find potential compliance conformance issues at an earlier stage and thus avoiding any late-stage revisions.

**VALSPEC UTILITY**

This utility consists of three key components as shown in the flowchart below:

- Validation of specification metadata to check for compliance with CDISC standards.
- Validation of the data to check for compliance
- Categorization and generation of findings based on the severity of findings

The key functionality of this utility will be further discussed in this paper.

![Flowchart of VALSPEC Utility](image)

**Figure 2. Process flowchart**

**VALSPEC SUMMARY OF FINDINGS OVERVIEW**

One of the compliance checks that the utility performs is to check controlled terminology (CT) that is being referenced in the specification metadata and validate it against the standards.

Controlled terminology (CT) is the set of code lists and valid values used with data items within CDISC-defined datasets. Controlled terminology provides the values required for submission to FDA and PMDA in CDISC-compliant datasets. CDISC, in collaboration with the National Cancer Institute's Enterprise Vocabulary Services (EVS), supports the controlled terminology needs of CDISC foundational and therapeutic area standards. The references to the most current-controlled terminology can be referenced and downloaded at the [CDISC website](https://www.cdisc.org).

The summary of findings generated by the VALSPEC utility in this example of DM domain is classified as the error and warnings as shown in figure 3 below.
Figure 3. Sample output from VALSPEC utility

In the example below, it can be noted that the terminology used in the DM domain is not consistent with data as shown in figure 4 below, causing an error to be displayed.

Figure 4. The terminology used for ARM and ARMCD variables in DM

The warnings issued by the VALSPEC are due to the excess country codes that are not being removed from the DM domain but not used in the study as shown in figure 5 below.
Figure 5. Terminology list used in the DM domain having excess country codes

These excess country codes, if not removed will cause define.xml to populate unnecessary country codes that are not relevant to this study.

VALSPEC FUNCTIONALITY

The SAS code below is used in the VALSPEC utility to read in the terminology data from the DEFINE and SELECT terminology metadata.

```sas
/*Using the Tables tab information to select the domains that are needed to be validated*/
data AllData;
set &xml.tables and=ends;
  count=1;
/*if dataname macro variable has ALL then validate all the domains in that study*/
/* else validate only the specific domain*/
  %if %str(&dataname) = ALL then %do;
    where REMOVE='';
    %end;
  %else %do;
    where REMOVE=''; and DATASET in({&datanames});
    %end;

    filename_base=compress(DATASET,'-');
    call symputx('domain',put(count,1.-1),scan(filename_base,1,'.'));
    /*Max macro variable will have the count of domains that need to be checked*/
    if end then call symputx('max',count);
run;
/*read in the Define terminology*/
%put adomain1;
data define_termmiology_;
set &spec.define_termmiology;
  SRC='DEFINE';
r
/*read in the Select terminology*/
data select_termmiology_;
set &spec.select_termmiology;
  SRC='SELECT';
    where REMOVE='';
r
```

The macro variable “max” keeps a counter of all the domains that need to be checked using the utility. The parameter ‘dataname’ has a default value of “ALL” which indicates that the utility needs to check all
the domains used in that study. If the VALSPEC only has to check one specific domain, then the corresponding domain name needs to be assigned to the “dataname” parameter.

Once the select and define terminology is being read, they are combined to create a repository of controlled terminology that is used in the study for each of the domains as shown in the code below. Note that the macro variable “i” will be used to distinguish the corresponding domains against the controlled terminology.

```plaintext
data final_summary_of_spec;
run;

/*read in define terminology for each of the domains*/
%do i = 1 %to nmax;
  data define_terminology;
  set define_terminology;
  %let defn = DEFINE;
  %let domain = %scan(\&domains\&i);
  where upcase(strip(dataset)) = strip(upcase("\&domain"));
  run;
/*read in select terminology for each of the domains*/
data select_terminology;
  set select_terminology;
  %let select = SELECT;
  where upcase(strip(dataset)) = strip(upcase("\&domain"));
  run;
/*combine the define and select terminology*/
data terminology;
  length CONDITIONAL_VALUE $19 CONDITIONAL_DATASET $20 CONDITIONAL_VARIABLE $20 CONDITIONAL_VALUE $200;
  set define_terminology define_terminology;
  %let i = 1;
  %do j = 1 %to nmax;
    %let temp = \&variables\&j;
    %let temp = trim(upcase("\&domain\&i"));
    %if %str(\&define) = 'DEFINE' %then \&ctlist = \&variables\&j;
    %else \&ctlist = \&ctlist \&temp;
    %let i = \&i + 1;
  %end;
  %let ctlist = \&ctlist;
run;

/*get CT list that is not part of the Terminology*/
proc sort data=spec_\&domain out = spec_\&domain__list (keep = \&ctlist) nodupkey;
  by \&ctlist;
run;
proc sort data=terminology out = \&domain__term (keep = \&ctlist) nodupkey;
  by \&ctlist;
run;
data \&domain__term_list(drop=\&ctlist);
  length DOMAIN CTLIST_REF MSG $200;
  merge \&domain__list(list) \&domain__term (in-term);
  by \&ctlist;
  length msg $200;
  %if term and not list then msg = "Format reference in Terminology not used in \&domain spec ";
  %else if list and not term then msg = "Format reference in \&domain spec not found in Terminology list";
  %let domain = \&domain;
  %let CTLIST_REF = \&ctlist;
run;
data final_summary_of_spec;
  set final_summary_of_spec \&domain__term_list;
  if DOMAIN ne "" and msg ne "";
  label msg = 'Message' CTLIST_REF = 'CTLIST Name';
run;
%end;
```

The controlled terminology references that are used in the specification document are not checked against the CT and corresponding messages are issued using the code below.
If a required variable is being removed in the specification an error message is created using the code below.

data spec_sdomain1;
set spec_sdomain1;
length MSG3 206;
if VARIABLE_REQUIRED='Y' and REMOVE ne '' then MSG3='ERROR: Variable ' || compress(VARIABLE) || ' is a required variable but is currently removed '
run;

%do i = 1 %to 100:
/*read in each domains*/
data admid_sdomain1;
set admid_sdomain1;
run;
/*Select the variables that has references to CT*/
data spec_sdomain1;
set spec_sdomain1;
if CTLIST ne '' and remove=' ' and CTCONFIG ne 'FIXED';
  length cclist_n 2000;
  if index(CLIST, ', ') then cclist_n= scan(CLIST, ', '); else cclist_n= CTLIST ;
run;
/*get the unique variable names that need to be checked for compliance*/
proc sort data= spec_sdomain1 out= varlist_sdomain1 (keep=VARIABLE cclist_n SUPPRESS_FLAG) nodupkey;
  by VARIABLE cclist_n;
run;
/*Check the CT references */
data defn_sdomain1;
set defn_sdomain1;
SRC='DEFN';
where upcase(strip(dataset)) = strip(upcase("sdomain1"));
run;
data Select_termncology;
set Select_termncology;
SRC='SELECT';
where upcase(strip(dataset)) = strip(upcase("sdomain1"));
run;
data termid_sdomain1(keep= DATASET VARIABLE SUBMISSION_VALUE cclist_n);
length CONDITIONAL_VALUE 19 CONDITIONAL_DATASET 10 CONDITIONAL_VARIABLE 80 CONDITIONAL_VALUE1 6000;
put select_termncology define_termncology;
ivariable='I' || strip(variable);
if upcase(strip(dataset)) = strip(upcase("sdomain1"));
if SRC='DEFN' then CTLIST=VARIABLE;
length cclist_n 2000;
if index(CLIST, ', ') then cclist_n= scan(CLIST, ', '); else cclist_n= CTLIST ;
run;
proc sort data= termid_sdomain1 ;
  by VARIABLE cclist_n;
run;
data termid_sdomain1;
merge termid_sdomain1(in=a) varlist_sdomain1 (in=b);
  by VARIABLE cclist_n;
  if a and b;
run;
proc sort data=termid_sdomain1 cut=common_sdomain1 (keep= DATASET VARIABLE) nodupkey;
  by DATASET VARIABLE;
run;
/*maxvar macro variable will have the count of variables that need to be checked against the CT*/
%let maxvar=0;
data varsct;
set common_sdomain1 end=end;
  count=1;
filename_base=compress(VARIABLE, "-'");
call symputx('vars',input(count, 4-1), scan(filename_base, 1, ', '));
  if end then call symput('maxvar', count);
rungo
The VALSPEC utility macro is invoked using the macro call `%valspec` with the parameters as shown below. Dataloc and specloc parameters are used to locate the data and specification location.

```
%valspec(dataloc=, specloc=, ckdata=, dataname=);
```

This utility can be expanded based on the requirement of individual organizational requirements and evolving compliance requirements.
CONCLUSION
The quality of the SDTM/ADaM specifications has a key role in determining the quality of the define.xml document, which is a key part of the CRT package submitted to the regulatory agencies for approval.

Specification updates based on the compliance findings while creating the final CRT package may result in post-lock updates to the SDTM/ADaM and TFLs. These post-lock updates can cause significant submission delays.

VALSPEC utility is designed to alert such non-compliance aspects at an earlier stage thereby streamlining the submission process and minimizing any rework risks.

REFERENCES
How can we ensure our study data is FAIR (Findable, Accessible, Interoperable, and Reusable)?

Automate Process to Ensure Compliance with FDA Business Rules in SDTM Programming for FDA Submission:

Introduction to the CDISC Standards:

ACKNOWLEDGMENTS
Sincere thanks to Steve Benjamin, Director Statistical Programming, Biostatistics and Global Contracts and Aman Bahl, Associate Director, Statistical Programming, Clinical Division for their vision, great leadership, persistent support, and encouragement throughout and for their valuable assistance in reviewing this paper.

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

Hrideep Antony
Principal Statistical Programmer, Clinical Division, Syneos Health
Work Phone: +1-984 459 4785
Email: hrideep.antony@syneoshealth.com
Web: http://www.syneoshealth.com

Aman Bahl
Associate Director, Statistical Programming, Clinical Division, Syneos Health
Phone: +1-905 399 6715
E-mail: Aman.Bahl@syneoshealth.com
Web: http://www.syneoshealth.com