

## **The Benefits and Know-how of Building a Central CDISC-Terminology Dictionary with a Macro System**

Solomon Lee, Ph D, South San Francisco, CA  
Ronghai Bo, Medicine, WI

### **ABSTRACT/INTRODUCTORY**

As FDA pushes for submission in CDISC format for NDAs, more and more companies start to pay more attention to handling the time consuming compliance issues in CDISC terminology. An example of the issue is that in the CM domain of a particular study, there are about 150 unique entries in the variable cmdosfrq. Less than one third of them are C-term compliant while the rest are not. This shows that there is a need for a systematic and efficient way to handle this issue. The benefits are:

First, a study level format dictionary can be efficiently established so as to map the non-compliant entries of those variables that fall into the covering range of the C-term (CDISC terminology).

Second, the study level C-term dictionary can be used as a building block for a central C-term dictionary either at the project level or at the therapeutic level. As a result, the processing efficiency will be further improved as more studies are processed. The same method and code framework could be applied to other kinds of dictionaries in clinical trial processing such as the building of a central labname dictionary or lab test unit conversion dictionary.

This paper would address the building of a central C-term forming dictionary step by step with easy to understand SAS code.

### **SHRINK THE OPEN-CDISC REPORT**

Start with running Open-Cdisc software on SDTM domains of a study. This could be done on a single domain to a few domains to the whole set of domains of a study.

The generation of a summary report from the original Open-cdisc report with thousands of entries of warnings, errors, and notes of issues. One can further shrink the summary excel list with only unique codelist numbers for C-term issues.

```
Proc sort data=xxx out=xxx_brf;  
  By codelist;  
Run;
```

### **GET THE C-TERM SHEETS OUT**

The generation of C-term non-compliant sheets for other functions such as the medical writing and clinicians to help

```
data dm_;  
  set dm;  
  race=put(race,$race.);  
run;
```

```
*** provide STDM domain name, varname, codelist code  
    to get what is on the terminology list and what is not on
```

Due to compatibility issues between SAS excel output and input files, I kick out 60 workbooks here to be updated at the excel side and pulled in later for format mapping.

```
***;  
ods listing;  
  
%macro ck_lst(dds,var,code,seq);  
  
proc freq data=&dds;  
  table &var./list missing out=frq;  
run;  
  
proc freq data=term_a;  
  table sub_value/list missing out=c_frq;  
  where c_code="&code";  
run;  
  
proc sql;  
  create table have_&var as  
  select distinct(upcase(a.&var)) as &var length=40 from frq as a ,c_frq as b  
  where upcase(a.&var)=upcase(b.sub_value)  
  ;  
  create table lack_&var as  
  select distinct(upcase(&var)) as &var length=40 from frq where upcase(&var) not in  
  (select upcase(sub_value) from c_frq)  
  ;  
quit;  
  
data comb_&var;  
  
retain &var codelist start label compliance fmtname type;  
  length label start $120 type $1 codelist $10 fmtname $40 &var $40;  
  set have_&var(in=a) lack_&var(in=b);  
  type='C';  
  fmtname="&var";  
  codelist=upcase("&code");  
  start=&var;  
  if a then do;  
    compliance=1;  
    label=&var;  
  end;  
  else do;  
    compliance=0;  
    label='to be filled';  
  end;  
  
run;  
%global n&seq;  
%let n&seq=&var;  
%put n&seq= &&n&seq;  
  
PROC EXPORT DATA= comb_&var  
  outfile= "c:\term\&var"  
  replace  
  dbms=xlsx ;  
  sheet="&var";  
run;
```

```
%mend;
```

```
%ck_lst(CM,cmdosfrq,C71113,1)
```

```
%ck_lst(DM,race,C74457,2)
```

## THE GENERATION OF THE STUDY LEVEL C-TERM FORMAT DICTIONARY

```
** Import the updated excel book for format cntlin statement  
Multiple sheets to be concatenated for the format file
```

```
***,  
,
```

```
option mprint mlogic symbolgen;  
%let lst=race cmdosfrq;  
%put &lst;
```

```
%macro sheets;  
%let i=1;  
%do %while (%scan(&lst,&i) ne);  
%let var=%scan(&lst,&i);
```

```
proc import datafile="c:\term_map\&var"  
out=work.nn&i(drop=&var)  
dbms=xlsx replace;
```

```
range="&var.$";  
getnames=yes;  
MIXED=NO;  
* SCANTEXT=YES;
```

```
run;  
%if %eval(&i) =1 %then %do;  
data fmtfile;  
length fn $30;  
set nn1;  
fn=fmtname;  
run;  
%end;
```

```
%if %eval(&i) >1 and %scan(&lst,&i) ne %then %do;
```

```
data fmtfile(drop=fn);  
length fmtname $30;  
set fmtfile nn&i(rename=(fmtname=fn));  
fmtname=fn;  
run;  
%end;  
%let i=%eval(&i+1);  
%end;
```

```
%mend;  
%sheets
```

```
*** the fmt call-in part ***;
```

```
proc format cntlin=fmtfile;  
run;
```

## **THE CENTRAL C-TERM FORMAT DICTIONARY**

The central format file should be a concatenation and sort for unique entries of several study specific format excel workbooks

## **HOW TO APPLY THE C-TERM DICTIONARY**

It can be done in a couple of ways.

- A. A column of format on the SDTM specs excel sheet.
- B. Code to map the relevant variable with C-term format dictionary in the title-footer macro or a separate macro called in the SDTM domain program.

## **CONCLUSION**

As a result of the application of such a C-term a dictionary system, the benefits are an increase of productivity and accuracy in handling the c-term compliance issues.

## **REFERENCES**

**CDISC Variable Mapping and Control Terminology Implementation  
Made Easy** by Balaji Ayyappan and Manohar Sure

## **CONTACT INFORMATION**

Solomon Lee, [Solomon.Lee@gilead.com](mailto:Solomon.Lee@gilead.com), 317-748-9504

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.

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