

How to effectively deal with hardcoding and CDISC Controlled Terminology in clinical studies

Lennert van der Zee, OCS Consulting, 's-Hertogenbosch, the Netherlands

ABSTRACT

When a large number of clinical studies is added to one integrated database or is transformed into SDTM and ADaM standards, the values of many variables may be inconsistent across studies. This paper describes how to get consistency of the variables within all studies by using a single Excel spreadsheet for all value updates, effectively separating the hard-coding from the actual programming and from the program validation. The paper will go in-depth on the advantages of this approach on how to use a SAS macro to update the values of the variables, and on how to implement CDISC Controlled Terminology in the SAS macro and the advantages of this.

INTRODUCTION

Integrated databases or clinical data warehouses, and data sets that are transformed into SDTM and ADaM standards, require consistent values of all variables across studies. The values of the variables are usually consistent within a study, but may often differ across multiple studies. This can be caused by a changing Case Report Form over the years, diversity in the format of the same findings described by the investigators and different data sources, e.g. Electronic Data Capture (EDC) systems.

In order to solve these inconsistencies, it was often chosen to simply apply hardcoding in the individual programs. Hardcoding consisted of adding one or a few lines of program code in these programs. This was a time-consuming activity, because many programs had to be checked whether the values of a variable and which values of a variable had to be adapted. The chance of a mistake was significant, because a clear overview could be quickly lost. Moreover, all the programs which were modified had to be validated again.

The disadvantages which are described above can be avoided by applying both of the following points:

- Collecting all original values of variables and their associated values that are converted to uniform values or SDTM and ADaM standards. All these values are stored in one document, e.g. an Excel spreadsheet.
- Including one or more SAS macro calls in every individual SAS program. This SAS macro ensures that the values are read from the document and the values of the variables are updated in the data sets.

The advantage of this method is that the values can be modified at one central location and the need of hardcoding the values into the programs is eliminated. As a result, revalidation of the SAS programs is not required, if the values need to be changed. The overview of all values updates is maintained and time is saved on validation and implementation of hardcoding.

In this paper an effective SAS macro UPDATE_VALUES is described that reads a single Excel spreadsheet with values updates of the variables and updates the values of the variables, so that all values of these variables are consistent or comply with the CDISC Controlled Terminology. The whole process of this SAS macro is illustrated by using sample data.

The macro presented in this paper is available for download from the OCS Consulting website, see at the end of this paper for the download location.

THE ORIGINAL DATA

In this paper the following data is used as an example of an integrated database in which the values from adverse event data set have to be converted to CDISC Controlled Terminology. The variables AESEV for 'Severity or Intensity' and AEREL for 'Relationship to Study Product or Causality' are used in this example, see *Table 1*.

PhUSE 2016

STUDYID	SUBJID	AESEQ	AETERM	AESEV	AEREL
MARS	1001	1	NAUSEA	MILD SEVERITY	UNLIKELY REL
MARS	1002	1	DIZZINESS	MODERATE SEVERITY	POSSIBLE REL
MARS	1003	1	FEVER	MODERATE SEVERITY	UNLIKELY REL
NEPTUNE	1001	1	ABDOMINAL PAIN	MOD	UNRELATED
NEPTUNE	1002	1	COUGH	SEV	PROBABLE
NEPTUNE	1003	1	VOMITING	MOD	DEFINITE
PLUTO	1001	1	COLD	1	3
PLUTO	1001	2	NAUSEA	1	1
PLUTO	1002	1	DIARRHOEA	1	2
SATURN	1001	1	LOSS OF APPETITE	SEVERE	PROBABLY
SATURN	1001	2	HEADACHE	MILD	UNLIKELY
SATURN	1001	3	NAUSEA	MILD	NOT RELATED

All other variables

Table 1 – A number of observations of Severity and Causality in the integrated database for Adverse Events

THE UNDESIRABLE WAY OF HARDCODING

A method which is still regularly used to update the values in the integrated database is hardcoding. Based on the sample data from *Table 1*, the values of variable AESEV need to be updated to the following CDISC Controlled values: "MILD", "MODERATE" and "SEVERE". This means that within a data step a number of lines of undesirable SAS program code may be added to a SAS program that creates an adverse event data set. If this was applied to study MARS, the following SAS program code was probably used:

UNDESIRABLE PROGRAM CODE

```
if upcase(studyid)="MARS" then
do; if upcase(aesev)="MILD SEVERITY" then aesev="MILD";
    else if upcase(aesev)="MODERATE SEVERITY" then aesev="MODERATE";
end;
```

This method of updating the values of variables is not desirable because of a lack of flexibility in order to revise the values of variables. In addition to this, validation of the programs is very time-consuming. In this paper a more desirable and effective method for updating values of variables by using Controlled Terminology will be described.

UPDATING VALUES OF VARIABLES BY USING CONTROLLED TERMINOLOGY

The most effective method for updating values of variables is using the SAS macro UPDATE_VALUES described in this paper. The functioning of this macro will be explained on the basis of the following steps:

- The steps which the user of the macro has to do: the two manual steps
- The steps that are done by the macro: the six macro steps

MANUAL STEP 1: CREATE AN EXCEL SPREADSHEET FOR ALL VALUE UPDATES

In order to update the values of the variables in the integrated database or clinical data warehouse, all unique values of the variables across all studies are entered in an Excel spreadsheet. You may use a custom piece of SAS code to obtain all unique values in your variable. Based on the data from *Table 1* the Excel spreadsheet will be as follows, see *Table 2*. In this Excel spreadsheet the following items are required:

- The study name: STUDY
- A name or title that covers the contents of the variable: TERM
- The original value of the variable: ORIGINAL_VALUE
- The new value updates of the variable: NEW_VALUE

The values of variable TERM are names which covers the contents of the variables that needs to be updated. It is sensible to choose generic names for the variable TERM so that these values can be used for multiple variables from different data sets. An example is 'End Relative to Reference Time Point' which is the variable AEENRF in Adverse Events and the variable CMENRF in Concomitant Medication. As both variables have the same values "BEFORE", "DURING" and "AFTER", it is recommended to use one generic name for the variable TERM such as the value "TIMEPOINT" or "ENRF".

PhUSE 2016

STUDY	TERM	ORIGINAL_VALUE	NEW_VALUE
MARS	RELATIONSHIP	POSSIBLE REL	POSSIBLY RELATED
MARS	RELATIONSHIP	UNLIKELY REL	UNLIKELY RELATED
NEPTUNE	RELATIONSHIP	DEFINITE	DEFINITELY RELATED
NEPTUNE	RELATIONSHIP	UNRELATED	NOT RELATED
NEPTUNE	RELATIONSHIP	PROBABLE	PROBABLY RELATED
PLUTO	RELATIONSHIP	1	NOT RELATED
PLUTO	RELATIONSHIP	3	POSSIBLY RELATED
PLUTO	RELATIONSHIP	2	UNLIKELY RELATED
SATURN	RELATIONSHIP	NOT RELATED	NOT RELATED
SATURN	RELATIONSHIP	PROBABLY	PROBABLY RELATED
SATURN	RELATIONSHIP	UNLIKELY	UNLIKELY RELATED
MARS	SEVERITY	MILD SEVERITY	MILD
MARS	SEVERITY	MODERATE SEVERITY	MODERATE
NEPTUNE	SEVERITY	MOD	MODERATE
NEPTUNE	SEVERITY	SEV	SEVERE
PLUTO	SEVERITY	1	MILD
SATURN	SEVERITY	MILD	MILD
SATURN	SEVERITY	SEVERE	SEVERE

Table 2 – The value updates in the Excel spreadsheet of the observations of Severity and Causality in the integrated database of Adverse Events sorted by TERM, NEW_VALUE and STUDY

In *Table 2* only those values that actually occur in the data are presented. If the new values of the variable should meet the CDISC Controlled Terminology, it may be recommended to use the complete CDISC code list.

The Excel spreadsheet with the value updates is stored in a generic folder within the project.

MANUAL STEP 2: ENTER THE PARAMETERS IN THE SAS MACRO UPDATE_VALUES.

This SAS macro updates the values of one variable per study in a data set or an integrated database. For this purpose, the following six macro parameters are required:

- The input SAS data set name: INSET
- The output SAS data set name: OUTSET
- The study name or protocol number: STUDY
- A name or title that covers the contents of the variable: TERM
- The variable name containing the original values: ORGVAR
- The variable name containing the new value updates: NEWVAR

In the example below, the values of Severity or Intensity, variable AESEV from study MARS in data set AE01 will be updated. To avoid overwriting the original values of Severity or Intensity variable AESEV is renamed to AESEV_ORG. The SAS macro UPDATE_VALUES will return the value updates into variable AESEV_NEW in data set AE02. All original and new values of Severity or Intensity are obtained from the Excel spreadsheet under the term SEVERITY.

EXAMPLE FOR MANUAL STEP 2

```
%update_values(inset =work.ae01
               ,outset=work.ae02
               ,study =MARS
               ,term  =SEVERITY
               ,orgvar=AESEV_ORG
               ,newvar=AESEV_NEW
               );
```

In *Table 3* the blue frame shows which data in data set AE01 is selected.

PhUSE 2016

STUDYID	SUBJID	AESEQ	AETERM	AESEV_ORG	AEREL
MARS	1001	1	NAUSEA	MILD SEVERITY	UNLIKELY REL
MARS	1002	1	DIZZINESS	MODERATE SEVERITY	POSSIBLE REL
MARS	1003	1	FEVER	MODERATE SEVERITY	UNLIKELY REL
NEPTUNE	1001	1	ABDOMINAL PAIN	MOD	UNRELATED
NEPTUNE	1002	1	COUGH	SEV	PROBABLE
NEPTUNE	1003	1	VOMITING	MOD	DEFINITE
PLUTO	1001	1	COLD	1	3
PLUTO	1001	2	NAUSEA	1	1
PLUTO	1002	1	DIARRHOEA	1	2
SATURN	1001	1	LOSS OF APPETITE	SEVERE	PROBABLY
SATURN	1001	2	HEADACHE	MILD	UNLIKELY
SATURN	1001	3	NAUSEA	MILD	NOT RELATED

All other variables

Table 3 – The selected data in data set AE01 that needs to be updated

MACRO STEP 1: READ THE VALUE UPDATES IN THE EXCEL SPREADSHEET

To process the value updates, the SAS macro starts to read all data that are stored in an Excel spreadsheet as mentioned in *manual step 1*. The most effective way is to have a general and separate SAS macro that only reads data from Excel spreadsheets. SAS has various ways of reading data from Excel spreadsheets. It depends on your specific installation which one you have access to.

In this SAS macro the following parameters are required:

- The pathname where the Excel spreadsheet with the value updates can be found: PATHNAME
- The filename of the Excel spreadsheet: FILENAME
- The sheet name in the Excel spreadsheet: SHEET
- The output SAS data set name: OUTSET

In the example below the Excel spreadsheet UPDATE_VALUES.XLSX is read and the data is returned into data set UPDATEVALUES.

EXAMPLE FOR MACRO STEP 1

```
%read_xlsx_file(pathname=C:\PhUSE2016\DH11\Metadata
                ,filename=update_values.xlsx
                ,sheet   =Sheet1
                ,outset  =work.updatevalues
                );
```

MACRO STEP 2: MAINTAIN THE ROW ORDER IN THE DATA SET

To maintain the original sorting of the records in the data set during the value updates process, a temporary variable: ROWNUM will be defined in the data set. This variable is based on a sequence number of the records in the adverse event data set: AE01, recorded as macro parameter INSET. The result of this operation is stored in data set INSET_SORTED.

At the end of the value updates process variable ROWNUM can be used for restoring the original sorting of the records in data set AE01.

PROGRAM CODE FOR MACRO STEP 2

```
data work.inset_sorted;
  set &inset;
  rownum+1;
run;
```

In order to save space in the presentation of *Table 4*, variable AEREL is not displayed.

PhUSE 2016

STUDYID	SUBJID	AESEQ	AETERM	AESEV_ORG	AEREL + All other variables	ROWNUM
MARS	1001	1	NAUSEA	MILD SEVERITY		1
MARS	1002	1	DIZZINESS	MODERATE SEVERITY		2
MARS	1003	1	FEVER	MODERATE SEVERITY		3
NEPTUNE	1001	1	ABDOMINAL PAIN	MOD		4
NEPTUNE	1002	1	COUGH	SEV		5
NEPTUNE	1003	1	VOMITING	MOD		6
PLUTO	1001	1	COLD	1		7
PLUTO	1001	2	NAUSEA	1		8
PLUTO	1002	1	DIARRHOEA	1		9
SATURN	1001	1	LOSS OF APPETITE	SEVERE		10
SATURN	1001	2	HEADACHE	MILD		11
SATURN	1001	3	NAUSEA	MILD		12

Table 4 – Every record in data set AE01 gets a row number

MACRO STEP 3: DETERMINE THE TYPE OF THE VARIABLE

In order to process the values of the variable correctly it is necessary to know whether this variable has numeric or character values. For this purpose, the type of variable AESEV_ORG, recorded as macro variable ORGVAR in data set INSET_SORTED, is determined by using a generic SAS macro GETVARTYPE.

This macro has the following two macro parameters:

- The name of the data set: DSET
- The variable that needs to be updated: VAR

The power of the macro GETVARTYPE is that it returns one of the following values:

- The value “C”, if the variable contains character values
- The value “N”, if the variable contains numeric values

Subsequently, one of both values is assigned to the macro variable VARTYPE.

PROGRAM CODE FOR MACRO STEP 3

```

%macro getvartype(dset=
                ,var =
                );
    %let rc=%sysfunc(open(&dset, i));

    %if &rc>0 %then
    %do; %let varnum =%sysfunc(varnum(&rc, &var));
        %let vartype=%sysfunc(vartype(&rc, &varnum));
        %let rc      =%sysfunc(close(&rc));
    %end;
    %else %put WARNING: Macro getvartype cannot open data set &dset.;

    &vartype
%mend getvartype;

%let vartype=%getvartype(dset=inset_sorted
                        ,var =&orgvar
                        );

```

MACRO STEP 4: DETERMINE THE LENGTH OF THE VARIABLE

If the variable is a character value, it will be important to know the length of the variable. To prevent truncation of the values, this length is used to specify the length for the newly created variable AESEV_ORG. For this purpose, the length of variable AESEV_ORG, recorded as macro variable ORGVAR in data set INSET_SORTED, is obtained by using a generic SAS macro GETVARLENGTH.

This macro has the following two macro parameters:

- The name of the data set: DSET
- The variable that needs to be updated: VAR

PhUSE 2016

The power of the macro GETVARLENGTH is that it returns the length of the variable. This macro will be called, if the variable is character. This value is assigned to the macro variable VARLEN. For numeric values this macro is not called, because the length is set to 8, see *macro step 5*.

PROGRAM CODE FOR MACRO STEP 4

```
%macro getvarlength(dset=
                    ,var =
                    );

%let rc=%sysfunc(open(&dset, i));

%if &rc>0 %then
%do; %let varnum=%sysfunc(varnum(&rc, &var));
      %let varlen=%sysfunc(varlen(&rc, &varnum));
      %let rc    =%sysfunc(close(&rc));
%end;
%else %put WARNING: Macro getvarlen cannot open dataset &dset.;

&varlen
%mend getvarlength;

%if %upcase(&vartype)=C %then
%do; %let varlen=%getvarlength(dset=inset_sorted
                              ,var =&orgvar
                              );
%end;
```

MACRO STEP 5: SELECT THE DATA IN THE EXCEL SPREADSHEET WITH VALUE UPDATES

Now that the study name and term are known, and the type and length of the variable have just been determined, the required data in the Excel spreadsheet with value updates can be selected. The original values of the variable ORIGINAL_VALUE from the Excel spreadsheet stored in data set UPDATEVALUES are assigned to the variable AESEV_ORG, recorded as macro variable ORGVAR, and the new values of the variable NEW_VALUE from the Excel spreadsheet are assigned to the variable AESEV_NEW, recorded as macro variable NEWVAR. Additionally, the length of the macro variable ORGVAR can be set.

PROGRAM CODE FOR MACRO STEP 5

```
data work.updatevalues_selected;
  set work.updatevalues(where=(   upcase(study)=upcase("&study")
                                and upcase(term) =upcase("&term")));

%if %upcase(&vartype)=N %then
%do; length &orgvar 8;
      &orgvar=input(original_value, 8.0);
%end;
%else %if %upcase(&vartype)=C %then
%do; %let varlen=%getvarlength(dset=&inset
                              ,var =&orgvar
                              );
      length &orgvar $ &varlen;
      &orgvar=original_value;
%end;

rename new_value=&newvar;
drop term original_value;
run;
```

After processing this program code, the following data set containing the variables AESEV_ORG and AESEV_NEW is created.

STUDYID	AESEV_ORG	AESEV_NEW
MARS	MILD SEVERITY	MILD
MARS	MODERATE SEVERITY	MODERATE

Table 5 – The selected data in the Excel spreadsheet with value updates

PhUSE 2016

MACRO STEP 6: MERGE THE UPDATED VALUES DATA SET WITH THE INPUT DATA SET

Now the data set with the updated values can be merged with the input data set based on the study name or protocol number and the macro variable ORGVAR, in this example the variable AESEV_ORG. At the same time the records in the data set can be sorted based on the temporary variable ROWNUM which was determined in *macro step 2*.

PROGRAM CODE FOR MACRO STEP 6

```
proc sql;
  create table &outset(drop=rownum) as
  select  A.*
         ,B.&newvar
  from    work.inset_sorted      as A
  left join work.updatevalues_selected as B
  on      upcase(A.studyid)=upcase(B.study)
         and upcase(A.&orgvar)=upcase(B.&orgvar)
  order by rownum;
quit;
```

After processing this program code, the output data set AE02 is created. Compared with the original data set AE01, variable AESEV_ORG with the original values is kept and variable AESEV_NEW with the value updates is added. Retaining these two variables provides insight into how the Controlled Terminology is applied, see *Table 6*.

STUDYID	SUBJID	AESEQ	AETERM	AESEV_ORG	AESEV_NEW
MARS	1001	1	NAUSEA	MILD SEVERITY	MILD
MARS	1002	1	DIZZINESS	MODERATE SEVERITY	MODERATE
MARS	1003	1	FEVER	MODERATE SEVERITY	MODERATE
NEPTUNE	1001	1	ABDOMINAL PAIN	MOD	
NEPTUNE	1002	1	COUGH	SEV	
NEPTUNE	1003	1	VOMITING	MOD	
PLUTO	1001	1	COLD	1	
PLUTO	1001	2	NAUSEA	1	
PLUTO	1002	1	DIARRHOEA	1	
SATURN	1001	1	LOSS OF APPETITE	SEVERE	
SATURN	1001	2	HEADACHE	MILD	
SATURN	1001	3	NAUSEA	MILD	

Table 6 – The resulting data set AE02 after the merge of the value updates and one study in the integrated database

Ultimately, in the final integrated database variable AESEV_ORG should be dropped and variable AESEV_NEW should be renamed to AESEV.

APPLY CONTROLLED TERMINOLOGY TO THE ENTIRE INTEGRATED DATABASE

If the values of one variable for all studies in the integrated database need to be updated the selection based on study name should be removed in *macro step 5* or replaced by the following program code:

REPLACING PROGRAM CODE FOR MACRO STEP 5

```
data work.updatevalues_selected;
  set work.updatevalues(where=(%if &study^=%str() %then
                              %do; upcase(study)=upcase("&study") and %end;
                              upcase(term) =upcase("&term")));

  %if %upcase(&vartype)=N %then
  %do; length &orgvar 8;
        &orgvar=input(original_value, 8.0);
  %end;
  %else %if %upcase(&vartype)=C %then
  %do; %let varlen=%getvarlength(dset=&inset
                                , var =&orgvar
                                );
        length &orgvar $ &varlen;
        &orgvar=original_value;
  %end;
```

PhUSE 2016

```

rename new_value=&newvar;
drop term original_value;
run;

```

For this purpose, macro parameter STUDY in the SAS macro UPDATE_VALUES should be set to missing.

REPLACING EXAMPLE FOR MANUAL STEP 2

```

%update_values(inset =work.ae01
              ,outset=work.ae02
              ,study =
              ,term  =SEVERITY
              ,orgvar=AESEV_ORG
              ,newvar=AESEV_NEW
              );

```

If this macro call is applied, the following data set AE02 is created:

STUDYID	SUBJID	AESEQ	AETERM	AESEV_ORG	AESEV_NEW
MARS	1001	1	NAUSEA	MILD SEVERITY	MILD
MARS	1002	1	DIZZINESS	MODERATE SEVERITY	MODERATE
MARS	1003	1	FEVER	MODERATE SEVERITY	MODERATE
NEPTUNE	1001	1	ABDOMINAL PAIN	MOD	MODERATE
NEPTUNE	1002	1	COUGH	SEV	SEVERE
NEPTUNE	1003	1	VOMITING	MOD	MODERATE
PLUTO	1001	1	COLD	1	MILD
PLUTO	1001	2	NAUSEA	1	MILD
PLUTO	1002	1	DIARRHOEA	1	MILD
SATURN	1001	1	LOSS OF APPETITE	SEVERE	SEVERE
SATURN	1001	2	HEADACHE	MILD	MILD
SATURN	1001	3	NAUSEA	MILD	MILD

Table 7 – The resulting data set AE02 after the merge of the value updates and all studies in the integrated database

Ultimately, in the final integrated database variable AESEV_ORG should be dropped and variable AESEV_NEW should be renamed to AESEV.

RESULTING DATA SET

If the same procedure is followed for Relationship to Study Product or Causality, variable AEREL, the SAS macro UPDATE_VALUES should be applied as follows:

REPLACING EXAMPLE FOR MANUAL STEP 2

```

%update_values(inset =work.ae02
              ,outset=work.ae03
              ,study =
              ,term  =RELATIONSHIP
              ,orgvar=AEREL_ORG
              ,newvar=AEREL_NEW
              );

```

In this example the input data set AE02 contains the value updates of variable AESEV. If variable AEREL_ORG is dropped and variable AEREL_NEW is renamed to AEREL in the output data set AE03, the following data set is created:

PhUSE 2016

STUDYID	SUBJID	AESQ	AETERM	AESV	AEREL
MARS	1001	1	NAUSEA	MILD	UNLIKELY RELATED
MARS	1002	1	DIZZINESS	MODERATE	POSSIBLY RELATED
MARS	1003	1	FEVER	MODERATE	UNLIKELY RELATED
NEPTUNE	1001	1	ABDOMINAL PAIN	MODERATE	NOT RELATED
NEPTUNE	1002	1	COUGH	SEVERE	PROBABLY RELATED
NEPTUNE	1003	1	VOMITING	MODERATE	DEFINITELY RELATED
PLUTO	1001	1	COLD	MILD	POSSIBLY RELATED
PLUTO	1001	2	NAUSEA	MILD	NOT RELATED
PLUTO	1002	1	DIARRHOEA	MILD	UNLIKELY RELATED
SATURN	1001	1	LOSS OF APPETITE	SEVERE	PROBABLY RELATED
SATURN	1001	2	HEADACHE	MILD	UNLIKELY RELATED
SATURN	1001	3	NAUSEA	MILD	NOT RELATED

Table 8 – A number of observations of Severity and Causality in the integrated database for Adverse Events after using the macro to update values of the variables

ADVANTAGES OF USING AN EXCEL SPREADSHEET FOR ALL VALUE UPDATES

The application of SAS macro UPDATE_VALUES that reads the Excel spreadsheet with controlled terms and updates the values of the variables in an integrated database has the following benefits:

CLEAR DOCUMENTATION

Changes made in the values of the variables are clearly documented in one Excel spreadsheet. This means that adaptations in the values of the variables do not need to be done in the individual SAS programs, but in one Excel spreadsheet. This method provides a clear overview of all variables whose values are regulated by Controlled Terminology.

COMPLYING WITH THE CDISC CONTROLLED TERMINOLOGY

A clear overview of all the values of the variables in an integrated database reduces the risk that some values of variables in a study do not meet the CDISC Controlled Terminology. Applying the CDISC Controlled Terminology to every individual program increases the risk of deviations in apparently the same values of the variables.

GETTING CONSISTENCY

Applying Controlled Terminology ensures consistency of the variables within all studies. The values of the variables are defined at one central location. Value deviations are easy to be detected.

EASY TO MODIFY

If a value of a variable needs to be adapted, it is not necessary to make these modifications in one or more SAS programs that were used to create the integrated database. These modifications can be easily completed in the Excel spreadsheet with all value updates. If a new value of a variable needs to be added, a new line can be added to the Excel spreadsheet instead of adding a line of program code to one or more SAS programs.

PREVENTING HARDCODING

If a value of a variable needs to be corrected, only the value of the variable in the Excel spreadsheet needs to be updated. This prevents the writing of undesirable program code in a SAS program, known as hardcoding.

TIME-SAVING

If a value of a variable needs to be adapted or corrected in the Excel spreadsheet after validation, the SAS program doesn't have to be validated again. In particular, if this value occurs in many studies and SAS programs. The time-consuming validation of these SAS programs will be avoided.

POTENTIAL EXTRA FEATURES

The SAS macro UPDATE_VALUES as described in this paper assumes that all values in the data set can be mapped with the values which are entered in the Excel spreadsheet with value updates. In practice, it may happen that some original values are not entered in the Excel spreadsheet with value updates. To obtain an overview of the values of the variables that are not mapped, extra lines of program code can be added after *macro step 6* that issue statements in the LOG file to raise awareness of values occurring in the data but missing from the list.

PhUSE 2016

CONCLUSION

Using a SAS macro and an external document to apply Controlled Terminology in clinical studies has many advantages. Hardcoding is avoided, the values of variables can easily be adapted and an overview of adapted values can be presented without difficulties. Moreover, this method is very time-saving.

The SAS macro as described in this paper can handle either one type of variable per clinical study or one type of variable in all clinical studies. Hereby, it is recommended to create an Excel spreadsheet with the complete CDISC code list for every term.

CONTACT INFORMATION

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

Lennert van der Zee
OCS Consulting
Ruwekampweg 2G
5222 AT 's-Hertogenbosch
The Netherlands
Work Phone: +31 73 523 60 00
Email: sasquestions@ocs-consulting.com
Web: www.ocs-consulting.nl

The SAS macros described in this paper can be downloaded from:

<http://ocs-consulting.nl/phuse2016-dh11>