Proceedings

First SUGCDSC Conference

August 31, 2018

Silver Spring Civic Building at Veterans Plaza
8525 Fenton Street, Silver Spring, MD 20910
Washington DC Area, USA

SUG CDSC
SAS User Group for CDSC

sas
Official SAS® Users Group 2018

SAS User Group for Clinical Data Standards and Codes (SUGCDSC)
PO Box 1032, Rockville, MD 20849, USA
https://sugcdsc.org

Title: Macro to Automate SDTM & ADAM datasets Creation from Specification Template

Speaker: Vidya Muthukumar – Advanced Clinical

Author: Nirosha Reddy – Advanced Clinical

Key words:

SDTM**,** ADaM, Macro

Abstract:

As CDISC standards are required in clinical trials, many tools have been developed for automating SDTM and ADaM datasets creation. The main purpose of the macro is to increase the efficiency in building datasets, limiting errors in validation by creating a validated standard macro and also reducing the number of CDISC compliance findings. A macro program (called %make_empty_dataset) has been developed to create zero record SDTM & ADAM datasets by reading the information from its specification template. Macro first imports the specification template and will create a macro variable for number of datasets present in dataset tab (contains all the dataset names present in the spreadsheet) and will load the variable attributes into macro variables and finally creates zero observation datasets using the do loop, the datasets contains all the variables in the order of specification template and all variable attributes (i.e.name, type, label, length). In addition to creating all the datasets based on the order we have in template, this macro overcomes the challenge of inconsistencies between specification and the actual data sets. A benefit is efficiency for creating the datasets.

Outline of this article:

1. Introduction of macro % make_empty_dataset
   1.1. Import Specification template with datasets tab
   1.2. Create macro variable for number of datasets to be created based on template dataset tab
   1.3. Create do loop in order to create all datasets present in specification template.
   1.4. Import Specification template with each dataset sheet
   1.5. Load variable attributes into macro variables
   1.6. Create empty datasets
2. Discussion – the advantages
3. Conclusion

Macro to Automate SDTM & ADAM datasets Creation from Specification Template

1. Introduction of macro % make_empty_dataset

% make_empty_dataset is a SAS® macro program that generates SDTM & ADAM datasets by reading information from its specification template. The five major steps are described below:

1.1. Step 1- Import Specification template with datasets tab.

The specification template is usually saved in an Excel file. Following SAS® code demonstrates how to read from Excel file with dataset tab. "&file" is a macro parameter that indicates the location and file name of the specification template. An example of &file will be: Statistics_Non_Project_Items\Templates\CDISC SDTM and ADaM Spec templates\CDISC_ADaM_Spec_Template\ADaMIG v1.1 Spec Template.xls).

```
proc import
        datafile=&file.
        out=_list
        dbms=xls
        replace;
        sheet="datasets";
run;
```

1.2. Step 2- Creates macro variable for number of datasets to be created based on template dataset tab:

Following SAS® code demonstrates how to create macro variable for number of datasets and macro variable for names of the dataset. "&num" is a macro parameter that indicates the number of dataset and "&numdat" macro parameter indicates the name of the dataset i.e; each sheet has a dataset name in specification. An example of macro parameters is:&num=1,&numdat=adsl.

```
data _null_;
        set _list nobs=nobs end=eof;

        if _n_=1 then
        call symput("num", compress(put(nobs,3.)));
        call symputx('numdat'    || compress(put(_n_, 3.)), dataset);
run;
```

1.3. Step 3- Creating do loop:

Following SAS® code demonstrates how to create do loop using &num macro parameter such that the code runs for all datasets present in the specification template based on the order and will create the empty datasets.

```
%do k=1 %to &num;
```

1.4. Step 4- Import Specification template with each dataset sheet.

The specification template is usually saved in an Excel file. Following SAS® code demonstrates how to read from Excel file with each dataset sheet into SAS® dataset. "&file" is a macro parameter that indicates the location and file name of the specification template. "&numdat&k" is a macro parameter that indicates the name and number of each dataset sheet in the specification template. An example of &file will be: Statistics_Non_Project_Items\Templates\CDISC SDTM and ADaM Spec Templates\CDISC_ADaM_Spec_Template\ADaMIG v1.1 Spec Template.xls).

```
proc import
        datafile=&file.
        out=_temp
        dbms=xls
        replace;
        sheet="&&numdat&k";
run;
```

1.5. Step 5- Load variable attributes into macro variables:

Following SAS® code demonstrates how the macro variables &vars, &var, &label and &length are created such that the attributes are stored for all the variables in each dataset and will create the empty datasets.

```
data _null_;
        set _temp nobs=nobs end=eof;

        if _n_=1 then
        call symput("vars", compress(put(nobs,3.)));
        call symputx('var'    || compress(put(_n_, 3.)), variable_name);
        call symputx('label'  || compress(put(_n_, 3.)), variable_label);
  call symputx('length' || compress(put(_n_, 3.)),
                put(length, 3.));

        ** valid ODM types include CHAR, NUM, FLOAT, DATETIME,
        ** DATE, TIME and map to SAS numeric or character;
        if upcase(type) in ("NUM") then
                call symputx('type' || compress(put(_n_, 3.)), "");
        else if upcase(type) in ("CHAR", "DATE", "DATETIME",
                "DATE", "TIME") then
                call symputx('type' || compress(put(_n_, 3.)), "$");
        else put "ERR" "OR: not using a valid ODM type.  " type=;

run;
```

1.6. Step 6- Creating empty dataset:

Following SAS® code will demonstrates how to create empty datasets with all the variables and its attributes by using do loop and by using the macro parameters created in previous steps.

```
        data &&numdat&k;
                %do i=1 %to &vars;
                        attrib &&var&i label="&&label&i" length=&&type&i.&&length&i...
                        ;

                        %if &&type&i=$ %then
                                retain &&var&i '';
                        %else retain &&var&i .;
                        ;
                %end;
                if 0;
        run;
```

2. **Discussion**

There are three major advantages of using % make_empty_dataset macro. First, it increases the efficiency in building datasets i.e.; it creates empty datasets with the attributes. Second, limits errors in validation by creating a validated standard macro. Third, by reducing the number of CDISC compliance findings example length shouldn't exceed 200 characters. Once the variable attributes are read in to the macro parameters, the % make_empty_dataset macro will read in the macro parameters and will update the datasets variables accordingly.

3. **Conclusion**

The %make_empty_dataset macro uses the specification template to create empty SDTM & ADaM datasets. This new approach could improve the efficiency of datasets by limiting errors. Therefore, %make_empty_dataset macro can be used as an efficient tool for automating empty SDTM & ADaM datasets creation.

Contact Information:

Your comments and questions are valued and encouraged. Please contact author at:

Nirosha Reddy, Sr. SAS Programmer

Advanced Clinical

Email: nreddy@advancedclinical.com