

**Paper SI09**  
**Standardizing SDTM Laboratory (LB) Programming**  
**PHUSE Connect 2023**  
**Murali Marneni, Biogen, RTP, USA**  
**Stephanie Olexovitch, Biogen, Cambridge, USA**

Lab data plays a crucial role in clinical trials. The source data can be very complex and is commonly received from multiple sources, which adds to the intricacy of mapping the SDTM.LB domain.

Standardizing and developing CDISC compliant SDTM.LB to facilitate a Clinical Study Report is the end goal and proves to be time consuming and hard to maintain if it is not done efficiently from the start.

# Programming Challenges

- ☐ Multiple vendors with differing metadata
- ☐ Volume of data
- ☐ Unique programming styles
- ☐ Varying level of experience

## Issues to address

- ☐ Lengthy programs
- ☐ Inefficient programs that lead to long run time
- ☐ Limited reusability
- ☐ Hard to maintain throughout the life of the study

# LB Program Standardization

In order to resolve the challenges and issues, we created a solution to standardize the programs, making the programs more user-friendly and significantly reduced programming hours. We identified that developing new global macros and embedding them in a standard template program allows the team to create the study level programs with greater efficiency, maintainability, reusability, and understandability irrespective of vendor file structure.

# Standardization Steps

EDC Data

Step 1

- Map lab data captured in the EDC by using existing macro: **%edc\_mapping**

Step 2

- Utilize the **%biibsdtn\_create\_lb\_testcd** macro to read in vendor data and populate LB.LBTESTCD, LB.LBTEST by merging with an internally maintained TESTCD file

Step 3

- Identify the standard unit and conversion factor by using the **%biibsdtn\_create\_lbstd\_units** macro

Step 4

- Populate standard variables like: LBSTRESU, LBSTRESN, LBSTRESC, LBSTNRLO, LBSTRNHI, LBSTNRC by using the **%biibsdtn\_create\_lbvars** macro

Step 5

- Standardize the LBCAT by using controlled terminology and map the one-to-one variables like LBNAM and LBFAST

Step 6

- Combine EDC and Vendor data and populate the EPOCH, slot unscheduled visits and derive baseline flag by using existing global macros and output the final SDTM.LB

Vendor Data

All Data

# Step 1

Set up the program header and use an existing macro to map the EDC data to and interim LB dataset

```
/* **Setup macro call** */
```

```
%let chrProgram=biibxyz;  
%let chrProtocol=study123;  
%let chrAnatype=sdtm;
```

```
%include "/setup.sas";
```

```
*****  
Step1: EDC data mapping  
*****;
```

```
/**mapping lab data that captured in EDC **/
```

```
%edc_mapping(sdtmdomain = lb);
```

1

# Step 2

## %biibsdtm\_create\_lb\_testcd

- Used to merge external vendor Lab data with internally maintained TESTCD mapping guidance file to
- Pulls the respective SDTM\_TEST, SDTM\_TESTCD information which is later mapped to SDTM.LB.LBTEST, LB.LBTESTCD.
- Can be used on any vendor file that has variable names in any naming convention.
- Outputs excel file with tests that do not have values in the current TESTCD mapping guidance file, this will be reported to the governance team

### ★ New Macro

```
*****  
VENDOR data mapping  
*****;  
  
/* **Step2: Call Lab testcd macro to pull the SDTM_TESTCD, SDTM_TEST** */  
  
%biibsdtm_create_lb_testcd(dsin= lb_vendor  
                           ,dsout= lb_vendor_testcd  
                           ,lbtestcd=lbtestcd  
                           ,lbtest=lbtest  
                           ,lbcac=lbcat  
                           ,stdlib=stdlib  
                           ,stdsin=biib_master_testcd_test_map_guid  
                           ,provider_name="LAB VENDOR"  
                           );
```

2

# Step 3

## %biibsd\_tm\_create\_lbstd\_units

- Developed to get the conversion factor and standard unit with respect to the original collected unit by merging with an internal Biogen resource file (unit conversion mapping file).
- Input: dataset generated from Step 2
- Output:
  1. Dataset with standard unit and conversion factors populated
  2. Excel file with units that do not have standard unit and conversion factor, this is used to create a governance request

### ★ New Macro

```
/* **Step3: Unit Conversions ** */
```

```
data lb_vendor_testcd;  
set lb_vendor_testcd;
```

```
** Standardize LBSPEC into ct:specimen submission values **;  
if index(lbspec, 'Whole Blood') > 0 then lbspec = 'BLOOD';  
else lbspec = strip(upcase(scan(lbspec, 1, '')));
```

```
run;
```

```
%biibsd_tm_create_lbstd_units(dsin=lb_vendor_testcd  
                             ,dsout=lb_vendor_all  
                             ,lborresu=lborresu  
                             ,lbspec=lbspec  
                             ,stdlib=stdlib  
                             ,stdsin=BIIB_LAB_CONVERSION_FACTORS  
                             );
```

3

# Step 4

## %biibsdm\_create\_lb\_vars

- Uses step 3 output dataset as an input dataset
- This macro was developed for the purpose of populating the standard variables by using the conversion factor and standard unit populated in Step 3

★ New Macro

```
/* ***Step4: Creating SDTM.LB.LBORRESU, LBORNRL0, LBORNRI, LBSTRESN,  
LBSTRESC, LBSTRESU, LBSTNRLO, LBSTNRHI, LBSTNRC** */
```

```
%biibsdm_create_lb_vars(dsin= lb_vendor_all  
                        ,dsout=vendordata  
                        ,lbtestcd=lbtestcd  
                        ,lbtest=LBTEST  
                        ,lborres=LBORRES  
                        ,LBORRESU=LBORRESU  
                        ,LBORNRL0=_lbornrlo  
                        ,LBORNRI=_LBORNRI  
                        ,LBSTNRC=  
                        );
```

4

# Step 5

Create the one-to-one mapping variables as per source data collection and mapping specification

```
/* ***Step5: mapping one on one mapping variables based on specification****/
data all_vendor;
    %attrib(vars = DOMAIN SUBJID LBCAT LBORRES LBREFID LBTSTDTL LBNRIND
    LBNAM LBSPEC LBDTC LBLOINC LBDTC LBSPCCND LBSTAT LBREASND);
    set vendordata;

    /* LBNRIND*/
        if alrtfl ne '' then lbnrind=strip(put(alrtfl,$alrtfl.));
        if alrtfl ne '' and lbnrind = '' then put "warning: Please check
the format of alrtfl. where more values need to be added";

    /*one to one mapping variables*/
        LBNAM = strip(lbnam);
        lbfast = strip(lbfast);
        if _lbrefid ne . then lbrefid = strip(put(_lbrefid,best.));

    ** LBLOINC **;
        LBLOINC = strip(LBLOINC);

    ** LBDTC **;
        LBDTC = STRIP(LBDTC);

    ** LBSPCCND **;
        LBSPCCND = STRIP(SPECCND);

run;
```

5

# Step 6

- Set the EDC and Vendor intermediate datasets together
- Populate the EPOCH variable
- Slot unscheduled visits
- Create the baseline flag
- Populate SEQ
- Output the LB dataset to output library

```
/* ***Step6: Creating Baseline flag, slotting Unscheduled visits, epoch
variables*** */

***Combine EDC and Vendor datasets ***;

data edc_and_vendor;
set edc1b
    all_vendor;
run;

***Merging with dm dataset to get reference start date for deriving variables
***;

%biibsdm_merge_dm_v2(dsin= edc_and_vendor, dsout=lb_dm, dmloc=crtedir.dc,
dmvars=usubjid rficdtc rfstdtc rfxstdtc rfxendtc rfendtc, dmsubj=Y, debug=N);

***Calling EPOCH MACRO** *;
%epoch(date = lbdtc);

** **UNSCEDULED slotting based on scheduled visit macro
%biibsdm_unscheduled_visit_v2(DSIN=epoch
,DSOUT=lb_uns
,UNSCTEXT=%STR(UNS)
,VISITVAR=visit_
,BASENUM=100000
,DATE= lbdtc
,STDY=lbdy
,SORTVAR=
);

*****Assign baseline flag using standard macro. update timevars or groupby
variables as needed*****/

%biibsdm_create_b1f1(dsin= lb_uns , dsout=lb_base, xx=lb, timevars= lbdtc ,
groupby=usubjid lbtstdc lbcac lbspec lbmethod lbtstdt1,
date=lbdtc, result= lborres, rule=, refdsn=crtedir.dm, refdt=rfxstdtc);

*****Derive sequence variable ***;

%seq(dsin=lb_base,dsout=final_lb, byvars=%attrib_dssort., seqby=usubjid,
seqvar=lbseq, dropvars=);

***Output SDIM.LB*****;

data output.lb;
set final_lb;
run;
```

# Conclusion

As Lab dataset programming is often more complex, this programming technique along with a standard programming template helps to resolve some of the programming challenges and utilizes a highly efficient program that is easy to maintain. This technique also helps to achieve our goals and shorten the programming timelines.

1. CDISC, SDTM controlled terminology.
2. CDISC, SDTM IG 3.3 package.
3. FDA. Study Data Technical Conformance Guide - Technical Specification Document. March 2019
4. FDA CDER/CBER. Position on Use of SI Units for Lab Tests. October 25, 2013
5. FDA. Recommendations for the Submission of LOINC® Codes in Regulatory Applications to the U.S. Food and Drug Administration. November 2017

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.

## ACKNOWLEDGEMENTS

We would like to acknowledge all team members and leadership team for the support and endorsement.

## CONTACT INFORMATION

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

Name: Murali Marneni

Email: [murali.marneni@biogen.com](mailto:murali.marneni@biogen.com)

Name: Stephanie Olexovitch

Email: [stephanie.olexovitch@biogen.com](mailto:stephanie.olexovitch@biogen.com)