DM 06

Using CDISC Lab Terminology and Determining Standard Units

Mat Davis, Omnicare Clinical Research, King of Prussia, PA

ABSTRACT

In any clinical trial submission, laboratory test results are always necessary. However, there are many challenges that can face programming teams working through lab data. Lab test names can differ depending on what lab handles the testing. Units can vary from lab to lab, causing the need for standardization among units and conversions. Even within a submission, a need arises to maintain as much standardization as possible in lab test names, units and conversions. In addition to these difficulties, groups can also struggle with achieving the level of standardization required by CDISC (Clinical Data Interchange Standards Consortium) for SDTM (Study Data Tabulation Model) submissions.

When submitting CDISC compliant Lab SDTMs, it is important you use CDISC compliant terminology for lab tests. This paper will discuss the help that this terminology can afford in both individual and ISS studies. It will also provide practical SAS® techniques to ensure that LB submissions have LBTEST and LBTESTCD parameters that are SDTM compliant. In addition, standardized lab values will be discussed (--STRES variables) for use in individual and ISS studies. Particular attention will be paid to examples of SAS® code that help determine if the values for – STRES variables are unique for each LBTESTCD throughout a submission.

The ideas discussed in this paper will be applicable to SAS® users on any platform. This paper is intended for those who are interested in creating CDISC compliant LB SDTMs and should be accessible to SAS® users of all levels of skill and experience.

INTRODUCTION

Over time, the rules dictated by regulatory agencies have become both more numerous and more strict. Consider the SDTM IG (Implementation Guidelines) that have been produced over the year. Each guideline is more strict that the last, attempting to regulate as much as possible. While this can cause problems for pharmaceutical companies and programmers alike, you could argue that the standardization of clinical trial data helps the SAS® programmer just as much as the FDA. Consider an ISS (Integrated Summary of Safety) which covers the safety aspects of a number of studies dealing with the same drug, potentially conducted over a number of years. Programming issues, such as laboratory test names, can be difficult to regulate across studies. However, following SDTM guidelines prove to be helpful tools to walk us through this process.

LB SDTM

This paper will deal with two issues particularly related to LB SDTM domains. First, the use of controlled lab test terminology will be discussed; along with methods of ensuring that all appropriate controlled terminology has been implemented. Controlled lab test terminology is different from other types of controlled terminology which may consist of a simple ‘Y’ or ‘N’. The list of lab terminology is long, yet not comprehensive. There can be a great deal of difficulty in mapping the reported test name to the corresponding controlled terminology name, as well as confusion in the determination of whether controlled terminology even exists for a certain test. While this may take some time to determine the appropriate lab test to map to, the methods in this paper will not only ensure that all terminology is CDISC compliant, but will also serve as a quality control check to see that there are no lab tests left non-compliant.

STANDARD LAB VALUES

Secondly, the use of standardized lab values will be discussed. There appears to be a considerable amount of confusion about the place of these values. Programmers can be unsure of which units are appropriate to use for certain lab tests. In reality, according to CDISC guidelines, standard units are only intended to ensure a standard within an FDA submission and not a regulatory standard. Keeping that in mind, a sponsor could ask for any feasible unit to be considered the standard unit for a specific lab test. Again, the difficulty comes in studies that may have used different labs in the same submission. You will see, through certain processes, that you can be sure that both all lab tests have the same lab terminology and the same conversion factors.

CONTROLLED LAB TEST TERMINOLOGY

When programming the LB SDTM domain, two variables are used to describe lab test information: LBTEST and LBTESTCD. LBTEST is simply the name of the lab test performed. LBTESTCD, however, is constrained to be eight characters long and must be unique per LBTEST. Thus, an LBTESTCD of 'NEUT' could not be used to describe both LBTESTs 'ABSOLUTE NEUTROPHILS' and 'DIFFERENTIAL NEUTROPHILS'. Both LBTEST and LBTESTCD are subject to CDISC controlled terminology. For the consideration of this paper, SDTM Labtest Package 3 will be used throughout. This information can be found through the CDISC website (www.cdisc.org). This document is a list of all lab tests that CDISC has controlled terminology for. If a laboratory test is not found on this spreadsheet then, at this time, CDISC does not have controlled terminology. According to the SDTM IG, if a lab test is not found on this spreadsheet, then a new LBTEST/LBTESTCD can be created provided that the LBTESTCD is unique to the LBTEST and that both the LBTEST and LBTESTCD values are consistent throughout the submission when describing the specific lab test. The guidelines indicate that there are still procedures that need to be followed when specifying your own LBTESTCD values. Interestingly, this event does not occur often as the list of lab tests is extensive.

IMPLEMENTING CONTROLLED LAB TERMINOLOGY

There are obviously many methods that can be used to implement this controlled terminology, including developing macros that could map every conceivable tab test name to the CDISC controlled lab test name. However, this can get tedious and almost impossible to anticipate. For example, consider the LBTEST "MONOCYTES". To code for this, you would have to consider all possible lab tests:

```
SELECT(UPCASE(lbtest));
WHEN ('MONOS' 'MONO' 'MONOCYTES' 'ABS. MONOCYTES' 'ABS MONOS'…) lbtest='MONOCYTES';
```

Due to the variation that can be seen in the way that laboratories report their data, specifically local labs, this method can be almost impossible to implement with precision. Thus, the following method is recommended for data quality and assurance of compliance.

STEP 1: MAP YOUR LBTESTs

As previously stated, very rarely will labs report their lab tests in ways that are perfectly CDISC compliant. Thus, the need to manipulate the data by mapping the lab reported test names to CDISC compliant LBTESTs are necessary. Sometimes, this mapping can be obvious. For instance, a reported lab test of "Chol" can easily be mapped to LBTEST "Cholesterol". In other cases, some research is required to map to the appropriate lab test. In these cases, you may want to speak with someone who is an expert in labs to determine what the appropriate mapping is. There are many good textbooks as well that will help determine these mappings. For those in the CRO industry, it is recommended that you consult with your sponsor about these mappings as they are a manipulation of the raw data. An instance of one of these difficulties is mapping "UR Clarity". At first glance, there does not appear to be a lab test that maps to "UR Clarity". Most of the urinalysis tests deal with sediments or granulocytes and are not applicable. Only after some research will you realize that this should map to the CDISC LBTEST "Turbidity", described as "A measurement of the opacity of a biological specimen." While some of these mappings are obvious, others can be difficult, thus presenting the need for expert advice and double programming for quality control.

NOTE: SOME HELPFUL HINTS IN MAPPING LBTEST VALUES

There a few items to keep in mind when mapping these LBTEST values that may be helpful in clearing up confusion:

1. Most tests that have units of '%' may not map as they appear

At first glance, a reported lab test of "BASO" might easily map to "Basophils" when in actuality it depends on the reported units. If the units reported are '10^9/L', "Basophils" is absolutely appropriate. However, if the reported units are '%', "Basophils" would not be appropriate. Most lab tests reported as percentages are actually ratios of two different measurements. In the case of "Basophils (%)", this is the ratio of Basophils to Leukocytes found in a subject. Since this is a different test than determining the amount of "Basophils" in a subject, the appropriate LBTEST is "Basophils/Leukocytes" with the corresponding LBTESTCD of "BASOLE". When mapping anything with '%' units, be careful that they are mapped to the appropriate LBTEST that specifies a ratio instead of an absolute result.

2. Beware of urinalysis tests as they can be mapped to the same tests as general hematology and chemistry

Most labs will report their urinalysis lab tests with the word “Urine” in the lab test. For example, a lab test of RBC’s done through urinalysis will most likely be reported something like “Urine RBC’s”. Because of this, it can be difficult for programmers to learn that these lab tests should be mapped to the appropriate test regardless of the method used of testing. Therefore, in this example, both “Urine RBC’s” and “RBC’s” would get mapped to the LBTEST “Erythrocytes”. The natural question that follows is how, then, do we differentiate between LBTEST values that are from blood draws from those obtained through urinalysis. The variable LBSPEC and LBCAT can be used to specify what type of sample was collected. The description of LBCAT indicates ‘Used to define a category of related records across subjects. Examples: such as HEMATOLOGY, URINALYSIS, CHEMISTRY.’ Since LBCAT is also an expected variable, this can be helpful to differentiate between records of the same lab tests. In a similar fashion, LBSPEC is defined as ‘Defines the type of specimen used for a measurement. Examples: SERUM, PLASMA, URINE.’ As these different tests are mapped to the same LBTEST values, the LBSPEC and LBCAT values can be helpful in separating these records apart. Due to the nature of SDTM, and the fact that LBSPEC is one of the keys that is used by define.xml, it is recommended that LBSPEC be used to distinguish one LBTEST from another.

3. The CT list is extendable

After going through the process of mapping each lab test to the CDISC controlled LBTEST value, the rare occasion can arise where a lab test will not map to any of the controlled lab tests. According to CDISC, this code list is extendable. Consider the lab test ‘Agranulocytosis’. A thorough examination of the CT spreadsheet shows no test that is appropriate to be mapped to. Thus, an LBTEST of ‘Agranulocytosis’ is permissible. The only restriction on this lab test is that the name must be less than 40 characters. In a similar way, the LBTESTCD can be specified as long as it is 8 characters or less, such as “AGRAN”. This LBTESTCD value must be unique for the LBTEST specified, either in the submission or in the CT spreadsheet.

STEP 2: BRING IN LBTEST VALUES

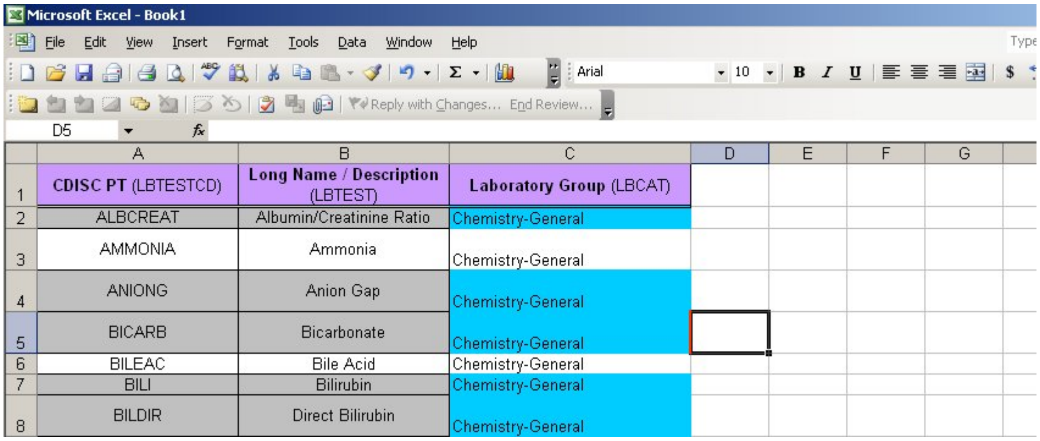
If CDISC is kind enough to provide an excel spreadsheet of all of the LBTEST / LBTESTCD combinations they expect, why not simply use their spreadsheet as it appears and simply import the information into our code? This way, we will actually check our LBTEST and LBTESTCD values against the CDISC specified spreadsheet to ensure that there are no missing or non-compliant values. This method will differ depending on the environment you are working in.

For Windows users, my recommendation is to call in the spreadsheet as is using the following macro:

```
DATA cdisc_ct;
       INPUT lbtestcd $ lbtest $;
       CARDS;
       ;
       RUN;

%MACRO lbimport(fileloc=);
       %DO i=1 %TO 14;
              %IF &i=1 %THEN %LET sheet=LB Chemistry - General;
              %ELSE %IF &i=2 %THEN %LET sheet=LB Chemistry - Enzymes;
              %ELSE %IF &i=3 %THEN %LET sheet=LB Chemistry - Hormones;
              %ELSE %IF &i=4 %THEN %LET sheet=LB Chemistry - Peptides;
              %ELSE %IF &i=5 %THEN %LET sheet=LB Chemistry - Proteins;
              %ELSE %IF &i=6 %THEN %LET sheet=LB Coagulation;
              %ELSE %IF &i=7 %THEN %LET sheet=LB Drug Screen;
              %ELSE %IF &i=8 %THEN %LET sheet=LB Flow Cytometry;
              %ELSE %IF &i=9 %THEN %LET sheet=LB General Observations;
              %ELSE %IF &i=10 %THEN %LET sheet=LB Immunology;
              %ELSE %IF &i=11 %THEN %LET sheet=LB Hematology;
              %ELSE %IF &i=12 %THEN %LET sheet=LB Sediment Analysis;
              %ELSE %IF &i=13 %THEN %LET sheet=LB Viral Loads;
              %ELSE %IF &i=14 %THEN %LET sheet=LB Vitamins;

       PROC IMPORT
       DATAFILE="&fileloc.\1_SDTMLabtest_Package3_Terminology_PublicReviewFINAL.xls"
       DBMS=excel REPLACE OUT=work.labtermx;
       SHEET='&sheet.';
       GETNAMES=no;
       RANGE='A6:B110';
       RUN;QUIT;
```

```
        DATA labtermx;
              LENGTH lbtest lbtestcd lbgroup $200;
              SET labtermx;
              lbtestcd=strip(f1);
              lbtest=strip(f2);
              lbgroup="&sheet.";
        RUN;

        DATA cdisc_ct (DROP=f1 f2);
              LENGTH lbtest lbtestcd lbgroup $200;
              SET cdisc_ct labtermx;
              WHERE NOT MISSING(lbtestcd);
        RUN;
        %END;
%MEND;
```

For UNIX users (and for Windows users who would like a different method), this can be a little more complicated and will take a little more work, but this work only needs to be done once. With UNIX environments, the PROC IMPORT macro shown above will not work in importing excel files, so another method needs to be used. To do this, simply begin a new spreadsheet, copy and paste the information from each individual spreadsheet into a master excel spreadsheet. Now, all LBTEST / LBTESTCD values should be contained in the same worksheet within an excel file. I recommend making a third group, LBGROUP, simply stating which spreadsheet each LBTEST originated, simply for ease in case a question comes un about which LBCAT an LBTEST value should be mapped to, or for ease in personal quality control. When this spreadsheet has been completed, save the spreadsheet as a .CSV file, which SAS® can read in on a UNIX machine. The document will look something like the following:



	A	B	C	D	E	F	G
1	CDISC PT (LBTESTCD)	Long Name / Description (LBTEST)	Laboratory Group (LBCAT)				
2	ALBCREAT	Albumin/Creatinine Ratio	Chemistry-General				
3	AMMONIA	Ammonia	Chemistry-General				
4	ANIONG	Anion Gap	Chemistry-General				
5	BICARB	Bicarbonate	Chemistry-General				
6	BILEAC	Bile Acid	Chemistry-General				
7	BILI	Bilirubin	Chemistry-General				
8	BILDIR	Direct Bilirubin	Chemistry-General				

When this has been completed, PROC IMPORT can be used to call in the .CSV file.

```
%MACRO csvimport (fileloc=);
DATA cttests;
            INFILE &fileloc. DELIMETER=',' MISSOVER DSD  FIRSTOBS=2 ;
               INFORMAT lbtestcd $8. ;
                INFORMAT lbtest $200. ;
               INFORMAT group $200. ;
               FORMAT lbtestcd $8. ;
               FORMAT lbtest $200. ;
               FORMAT lbgroup $200. ;
            INPUT
               lbtestcd $
               lbtest $
               group $
            ;
            RUN;
DATA cdisc_ct (KEEP=lbtest lbtestcd lbgroup);
        LENGTH lbtest lbtestcd lbgroup $200;
        SET cttests;
RUN;
%MEND;
```

Notice that only the values LBTEST, LBTESTCD and LBGROUP are retained. When checking our data for compliance, the other information (synonyms, descriptions, etc.) are not useful and can be discarded.

STEP 3: Assign LBTESTCD and check CT values

At this point in the process, all lab tests should have been assigned to their appropriate LBTEST value and the CDISC controlled lab terminology should have been brought in as a SAS® dataset. Remember, purposefully, no LBTESTCD values have been assigned, only LBTEST values. This third step is simple. Simply sort both the LB dataset and the CDISC CT dataset and do the following:

```
DATA lb;
      MERGE lb (IN=X) cdisc_ct;
      BY lbtest;
      IF x;
RUN;

PROC SORT DATA=lb OUT=check_lb NODUPKEY;
      BY lbtest lbtestcd;
RUN;

PROC PRINT DATA=check_lb;
      WHERE lbtestcd =’’;
      VAR lbtest lbtestcd;
RUN;
```

While this may seem simple enough, this step will apply all CDISC LBTESTCD values to their appropriate LBTEST values in LB. Secondly, it will go ahead and print out all LBTEST values that are not appropriately mapped according to CDISC. Once all LBTEST values have been appropriately assigned, no blank LBTESTCD values should print. If the case arose where you needed to specify your own LBTEST value (where the lab test did not appropriately map to any of the CDISC CT values), this will serve as a check in your program to show that these are the only LBTEST values that were not mapped.

Following these steps should ensure that all LBTEST values are on the CDISC CT list. In addition, it will serve to inform whoever looks at the program what LBTEST values did not map appropriately to ensure that all of these values are documented and appropriate for the submission.

STANDARDIZED LAB UNITS

In addition to mapping lab tests to their appropriate CDISC compliant names, another issue with the LB domain can be determining the appropriate LBSTRESU values to use. The LB domain is unique in that it is required that the original units reported by the labs must be converted to a standardized lab units, including both the original result and the upper and lower limits of normal. This can sometimes be difficult as lab values may come in many different names and units, making the standard units that much more difficult to determine. There are many misconceptions about these units, some of which should be cleared up shortly.

ASSIGNING STANDARD UNITS

There is a simple key to remember when assigning standardized lab units (LBSTRESU) to each lab test:

Each individual lab test (based on LBTESTCD, LBCAT or LBSPEC) in a submission must have one and only one LBSTRESU value assigned to it.

Now, to clear up one misconception, this does not mean that each LBTEST should get assigned one LBSTRESU value to it. Consider the following example. Within one lab transfer, more than likely the transfer would include lab tests of “Total Protein” and “Urine Protein” with corresponding units of “g/dL” and “mg/24 hr”. Using CDISC CT, both of these tests would be mapped to LBTEST=”Protein” with LBTESTCD=”PROT”. If we believed that each LBTEST should have the same LBSTRESU attached to it, we would have a serious issue with these lab tests. However, that does not mean that the key phrase has led us in error. Remember that, when assigning these lab tests, we would also assign either the LBSPEC variable to “BLOOD” and “URINE” or LBCAT variable to “HEMATOLOGY” and “URINALYSIS”. Therefore, to determine the individual lab tests, it is recommended that both LBTESTCD and either LBSPEC or LBCAT are used in determining what the individual lab tests are. Thus, the LBTESTCD of “PROT” and LBSPEC of “BLOOD” would be different than the LBTESTCD of “PROT” and LBSPEC of “URINE”. Therefore, these two lab tests can have separate LBSTRESU variables and still be compliant with the fact that each lab test must have one standard lab unit attached to it.

Specifically involving urinalysis tests, another issue can arise with these units. Many labs will report the units for these tests as “NO UNITS” or “NONE”. It is acceptable, in these cases, to have a blank value for the LBSTRESU variable, assuming once again that this value is consistent for each individual lab test.

There are a couple of issues that can make assigning these LBSTRESU values more difficult. The first issue is the case where different labs are processing tests for different patients. This happens in cases where patients are found in different countries or different areas where the same lab company does not handle all patients' lab data. The second case where this can be difficult is in an ISS submission. These submissions are comprised of more than one clinical trial and put together to test the safety of a certain drug after combining the results of many different studies about the same drug. In these cases, the labs will almost certainly come from more than one lab vendor and, thus, will require the same lab with many different units to be mapped to the same LBSTRESU value. In these cases, planning the mapping ahead of time is of utmost importance. There are a couple of ways that these can be mapped.

1. STANDARD OR CONVENTIONAL UNITS

One strategy, although not necessary, is to map each lab test to the appropriate SI units (International System of Units) or conventional units. The conversion factors for these units are well documented and can be found many places. The benefit of using these units is that is guarantees that each lab test, regardless of originally reported value, will all convert to the same standard unit. This can be helpful when receiving lab tests from many different vendors reported in many different lab units. It can also be helpful in determining department standards for STRES variables since all lab tests will be able to be mapped. The downside of using either SI or conventional units is the determination of clinically significant lab values at the ADaM (Analysis Data Model) level. Many companies have their values for clinical significance pre-specified in specific lab units. Doing two conversions (from original units to SI to clinically significant units) can lose precision and, potentially, distort whether lab values are truly clinically significant or not.

2. SPONSOR / STUDY UNITS

A second strategy, more highly recommended, is to use the units specified by either the client or the study specifications for the LBSTRESU values. This ensures that the issues of SI for determining standard units do not arise, including the issue of imprecision in determining clinically significant values. The problem with using these units, however, is that they can be difficult to find conversion factors for. There is a great deal of documentation for converting units to SI. However, for converting from non-SI to non-SI units, it may be more difficult to find the correct conversion factors, particularly in studies where there may be more than 10 separate units for the same lab tests. However, if it is possible to make these conversions, the reporting of the clinically significant values can be easier to program for and more precise.

UNIQUE LBSTRESU VALIDATION

In a study containing hundreds of lab tests at many different time-points, it can be hard to determine if all lab tests have the exact same LBSTRESU variables. To do so, macros can be used to validate that no lab test has two separate LBSTRESU values. It is important, as designated by the `%unikey` variable, to set the keys that determine a unique lab test. It is recommended that LBTESTCD and LBSPEC be used as these keys since they are considered two of the main keys by CDISC. Consider the following macro:

```
%MACRO validstresu(unikey=lbtestcd lbspec);

PROC SORT DATA=lb OUT=strescheck nodupkey;
       BY &unikey. lbstresu;
RUN;

PROC FREQ DATA=strescheck noprint;
       TABLES %SCAN(&unikey,1,' ')*%SCAN(&unikey,-1,' ')/OUT=stres;
RUN;

DATA stres;
       SET stres;
       WHERE count gt 1;
run;

%GLOBAL nobs;
%LET dataid=%SYSFUNC(OPEN(stres,IS));
```

```
%LET nobs=%SYSFUNC(ATTRN(&dataid,NLOBS));
%IF &dataid > 0 %THEN %LET rc=%SYSFUNC(CLOSE(&dataid));

%IF &nobs. GE 1 %THEN %DO;

PROC SORT DATA=stres;
       BY &unikey.;
RUN;

DATA stres;
       MERGE stres(in=a) strescheck;
       BY &unikey.;
       IF a;
RUN;

PROC PRINT DATA=stres;
       VAR &unikey. lbstresu;
       TITLE "Non-unique standardized lab test units determined by &unikey.";
RUN;
%END;
%MEND;
```

If a non-unique LBSTRESU is found, the program will output the affected tests and assigned standardized units, allowing you to isolate the issue and fix the problem. A fully compliant LB domain will produce no output from this macro, providing another check that the SDTM is CDISC compliant.

CONCLUSION

While lab datasets can be difficult enough, the introduction of CDISC regulation into the production of labs can add more work and more difficulty to the process. As has been shown, there are checks that can be put in place to ensure that your LB is compliant. Ensuring that the appropriate lab tests are used are vital to any submission, and the uniqueness of the LBSTRESU variable is crucial for a cohesive trial, particularly when more than one lab vendor is used. The using the processes shown here will assist you in your programming and give you comfort in the knowledge that a difficult part of your dataset is perfectly compliant.

REFERENCES

CDISC Inc. 2009. "SDTM & Labtest Package 3 Terminology”. http://www.cdisc.org/standards/index.html.
CDISC Inc. 2009. “The SDTM Implementation Guide, Version 3.1.2”. http://cdisc.org/models/sdtm/v1.2/index.html.

ACKNOWLEDGEMENTS

I would like to thank Elaine Dempsey for recommending this opportunity for me. I would like to thank PharmaSUG for affording me a scholarship to attend this conference. I would also like to thank everyone at Omnicare Clinical Research for their support of this paper.

CONTACT INFORMATION

Mat Davis
Omnicare Clinical Research
630 Allendale Road
King of Prussia, PA
(484)-679-2727
Matthew.Davis@omnicarecr.com