SDTM-ETL 3.0 User Manual and Tutorial

Author: Jozef Aerts, XML4Pharma

Last update: 2014-03-29

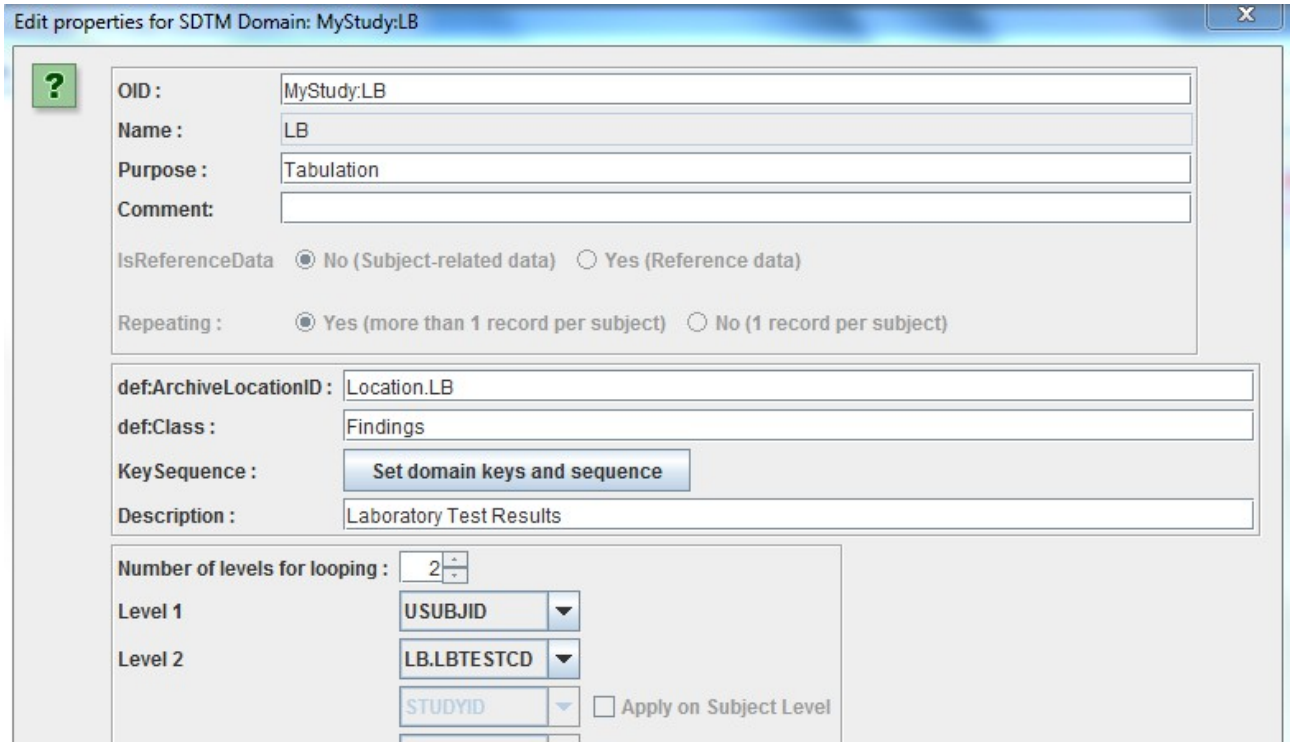
Tips for mapping the LB domain

Generating LB datasets is often one of the most difficult tasks, not only due to the large or even very large amout of data, but also because the lab data can have different providers (e.g. different labs), but also as lab data can appear in quite different forms.

Instead of providing a normal tutorial, describing the mapping for one specific situation, which is probably not your situation, we choose to describe a number of worthful tips for generating LB mappings and datasets.

Tip 1: Setting the looping variable

Keep it simple: in 90% of the cases, using only 2 looping variables (USUBJID and LBTESTCD) will be sufficient. You can check this by double clicking the first cell of the „LB row“ or using the menu „Edit – SDTM domain properties“:



Do not let confuse you by the SDTM-IG which states „One record per lab test per time point per visit per subject“. That statement is about the result[1], not about how you come to it. If, after a drag-and-drop, you select the correct visits (using the „Generalize for“ checkbox and the „Except for“ and „Only for“ buttons), forms and item groups, then you usually will automatically obtain the desired result. In some (but very few) cases you will want to add a third „looping variable“ like „VISITNUM“, but this is usually not necessary or even contraproductive.

1 It doesn't state either what variable corresponds to a „time point“

Tip 2: use several LB instances for different cases

Lab results can appear everywhere in a clinical study, and can have been generated by several labs.

This means that depending on the source (form, lab) you might need to map to a different set of LB variables. For example, you may have the case that a set of lab tests have been executed which have to do with pregnancy by lab A using form F1, and a set of blood lab tests by lab B collected on form F2.

In such a case, do not unnecessarily try to „push“ everything in a single LB instance, but generate two instances of the LB domain, and generate mappings for the „pregnancy case“ in one instance and mappings for the „hematology“ case in another LB instance.
The SDTM-IGs and „Metadata Submission Guides“ allow and even encourage such „splitting“ (essentially the term in incorrect in our case, as we even never had a single dataset).

In case you see you need or want to add additional variables which do not apply for all lab data, you should definitely generate a separate LB instance for those data that you need the additional variables. By doing so, you also make it easier for the reviewer by explicitely showing the reviewer in which cases (or subset of data) you have the additional variable and in which cases you don't.

Tip 3: reuse information

The LB table is a typical „hypervertical“ table, following the „entity-attribute-value“ model of databases. LBTESTCD is the „entity“ LBORRES the „value“, most of the others are „attributes“. Depending on which test is in the current row, the values for attributes like LBSTNRLO (Reference Range Lower Limit) and LBSTNRHI (Reference Range Higher Limit) will be different.

In SDTM-ETL, you can reuse variables values (but in read-only mode only) that you have defined before, i.e. more to the left in the same row. As LBTESTCD is pretty on the left, you can reuse its value in all variables that come after it (i.e. more to the right). Now suppose that you reference ranges are not the CRF, so you need to add them „by hand“. You can then e.g. have the following mapping script:

```
if($LB.LBTESTCD == 'GLUC') {
        $LB.LBSTNRLO = …;
} elsif ($LB.LBTESTCD == 'FRUCT') {
        $LB.LBSTNRLO =  ;
} …
```

As you have defined LBTESTCD before (more to the left) you do not need a drag-and-drop, you just can reuse the value from the mapping that you did before.

In case however that the reference range is also dependent on the method (LBMETHOD) you can not reuse the value from the LBMETHOD mapping as the latter comes after LBSTNRLO so you will probably need a drag-and-drop and create a temporary variable which has the same Xpath expression as fir the one you have for LBMETHOD itself.

For example:

```
$MYLBMETHOD = xpath(....);
if($LB.LBTESTCD == 'GLUC' and $MYLBMETHOD == '….' ) {
```

```
        $LB.LBSTNRLO = …;
} elsif($LB.LBTESTCD == 'GLUC' and $MYLBMETHOD == '….') {
        $LB.LBSTNRLO = …;
} elsif ($LB.LBTESTCD == 'FRUCT') {
        $LB.LBSTNRLO =  ;
} …
```