SDTM-ETL 4.6 User Manual and Tutorial

Author: Jozef Aerts, XML4Pharma

Last update: 2024-08-12

Tutorial: Merging datasets

Table of Contents

Introduction

SDTM and regulatory authorities don't make it easy to us when it comes to generating and submitting comprehensive and especially logical, sets of data.
For example, even now in 2023, it forces us to "ban" non-standard variables (i.e. sponsor-defined variables) to a supplemental qualifier (SUPPxx) dataset. Also, when a data point value exceeds 200 characters, it must be split into chunks of not more than 200 characters and the second and further chunk must be banned to SUPPxx. Also, there is the famous 5GB file size limit, forcing us to "split" datasets[1] when the XPT file size grows beyond that limit.
Essentially, there are three reasons for all this:
The first is the mandated use of the outdated SAS Transport 5 (XPT) format, which is extremely inefficient in byte storage. Selecting XPT 30 years ago was a major error, even simple CSV would have been a better choice.
The second is the lack of SDTM understanding at the FDA: many of the reviewers are not able to distinguish between standard and non-standard SDTM variable, so CDISC decided that these need to go into SUPPxx datasets, which reviewers are supposed to then re-merge into the "parent" dataset. This could easily be solved e.g. by color-coding of non-standard variable columns in modern viewers (as the information about standard and non-standard is in the define.xml), but the relative primitive tools (third reason) at the regulatory authorities do not support this.

Modern viewers such as the "Smart Submission Dataset Viewer" have such features, e.g.:

[1] The interesting thing is that in such case, one still needs to submit the >5GB dataset, although FDA claims that it cannot read these ...

AGE	AGEU	SEX	RACE	ETHNIC	ARMCD	ARM	ACTARMCD	ACTARM	COUNTRY	DMDTC	DMDY	COMPLT16	COMPLT24	COMPLT8	EFFICACY	SAFETY	ITT
61	YEARS	F	WHITE	HISPANIC ...	Pbo	Placebo	Pbo	Placebo	USA	2013-12-26	-7	Y	Y	Y	Y	Y	Y
64	YEARS	M	WHITE	HISPANIC ...	Pbo	Placebo	Pbo	Placebo	USA	2012-07-22	-14						
71	YEARS	M	WHITE	NOT HISPA...	Xan_Hi	Xanomelin...	Xan_Hi	Xanomelin...	USA	2013-07-11	-8	Y	Y				
74	YEARS	M	WHITE	NOT HISPA...	Xan_Lo	Xanomelin...	Xan_Lo	Xanomelin...	USA	2014-03-10	-8						
77	YEARS	F	WHITE	NOT HISPA...	Xan_Hi	Xanomelin...	Xan_Hi	Xanomelin...	USA	2014-06-24	-7	Y	Y				
85	YEARS	F	WHITE	NOT HISPA...	Pbo	Placebo	Pbo	Placebo	USA	2013-01-22	-21						
59	YEARS	F	WHITE	HISPANIC ...	Scrnfail	Screen Fail...	Scrnfail	Screen Fail...	USA	2013-12-20							
68	YEARS	M	WHITE	NOT HISPA...	Xan_Lo	Xanomelin...	Xan_Lo	Xanomelin...	USA	2013-12-23	-9	Y	Y				
81	YEARS	F	WHITE	NOT HISPA...	Xan_Lo	Xanomelin...	Xan_Lo	Xanomelin...	USA	2012-08-25	-13						
84	YEARS	M	WHITE	NOT HISPA...	Xan_Lo	Xanomelin...	Xan_Lo	Xanomelin...	USA	2012-11-23	-7			Y	Y	Y	Y
52	YEARS	M	WHITE	NOT HISPA...	Pbo	Placebo	Pbo	Placebo	USA	2014-02-27	-13	Y	Y	Y	Y	Y	Y
84	YEARS	M	WHITE	NOT HISPA...	Pbo	Placebo	Pbo	Placebo	USA	2014-02-09	-6	Y	Y	Y	Y	Y	Y
81	YEARS	F	WHITE	NOT HISPA...	Xan_Hi	Xanomelin...	Xan_Hi	Xanomelin...	USA	2012-10-23	-5	Y	Y	Y	Y	Y	Y
57	YEARS	F	WHITE	NOT HISPA...	Scrnfail	Screen Fail...	Scrnfail	Screen Fail...	USA	2013-09-05							
75	YEARS	F	WHITE	NOT HISPA...	Xan_Hi	Xanomelin...	Xan_Hi	Xanomelin...	USA	2013-05-07	-13				Y	Y	Y
57	YEARS	M	WHITE	NOT HISPA...	Xan_Hi	Xanomelin...	Xan_Hi	Xanomelin...	USA	2013-08-14	-9	Y	Y	Y	Y	Y	Y
79	YEARS	F	WHITE	NOT HISPA...	Pbo	Placebo	Pbo	Placebo	USA	2013-09-06	-17	Y	Y	Y	Y	Y	Y
82	YEARS	F	WHITE	NOT HISPA...	Scrnfail	Screen Fail...	Scrnfail	Screen Fail...	USA	2013-04-18							
62	YEARS	F	AMERICAN...	NOT HISPA...	Scrnfail	Screen Fail...	Scrnfail	Screen Fail...	USA	2012-09-30							
56	YEARS	M	WHITE	NOT HISPA...	Xan_Hi	Xanomelin...	Xan_Hi	Xanomelin...	USA	2013-01-28	-15				Y	Y	Y
79	YEARS	F	WHITE	NOT HISPA...	Xan_Hi	Xanomelin...	Xan_Lo	Xanomelin...	USA	2013-11-26	-9				Y	Y	Y
71	YEARS	M	WHITE	NOT HISPA...	Xan_Lo	Xanomelin...	Xan_Lo	Xanomelin...	USA	2013-02-03	-12				Y	Y	Y
80	YEARS	F	WHITE	NOT HISPA...	Xan_Lo	Xanomelin...	Xan_Lo	Xanomelin...	USA	2012-07-08	-14	Y	Y	Y	Y	Y	Y
81	YEARS	F	BLACK OR ...	NOT HISPA...	Pbo	Placebo	Pbo	Placebo	USA	2013-01-25	-8	Y	Y	Y	Y	Y	Y
76	YEARS	F	WHITE	NOT HISPA...	Xan_Lo	Xanomelin...	Xan_Lo	Xanomelin...	USA	2012-10-30	-16			Y	Y	Y	Y
69	YEARS	M	WHITE	NOT HISPA...	Pbo	Placebo	Pbo	Placebo	USA	2013-03-20	-10	Y	Y	Y	Y	Y	Y
56	YEARS	M	WHITE	HISPANIC ...	Xan_Hi	Xanomelin...	Xan_Hi	Xanomelin...	USA	2013-12-28	-14	Y	Y	Y	Y	Y	Y
57	YEARS	F	BLACK OR ...	NOT HISPA...	Scrnfail	Screen Fail...	Scrnfail	Screen Fail...	USA	2013-09-22							
61	YEARS	M	AMERICAN...	NOT HISPA...	Xan_Hi	Xanomelin...	Xan_Hi	Xanomelin...	USA	2014-01-25	-13	Y		Y	Y	Y	Y
56	YEARS	F	WHITE	HISPANIC ...	Xan_Hi	Xanomelin...	Xan_Hi	Xanomelin...	USA	2014-01-17	-8	Y	Y	Y	Y	Y	Y
67	YEARS	M	WHITE	NOT HISPA...	Xan_Lo	Xanomelin...	Xan_Lo	Xanomelin...	USA	2013-03-17	-7			Y	Y	Y	Y
61	YEARS	M	WHITE	NOT HISPA...	Xan_Hi	Xanomelin...	Xan_Hi	Xanomelin...	USA	2013-08-20	-9			Y	Y	Y	Y
80	YEARS	F	WHITE	NOT HISPA...	Scrnfail	Screen Fail...	Scrnfail	Screen Fail...	USA	2013-12-08							
68	YEARS	M	WHITE	NOT HISPA...	Xan_Lo	Xanomelin...	Xan_Lo	Xanomelin...	USA	2014-05-10	-12	Y	Y	Y	Y	Y	Y
79	YEARS	M	WHITE	NOT HISPA...	Xan_Lo	Xanomelin...	Xan_Lo	Xanomelin...	USA	2012-09-16	-16	Y	Y	Y	Y	Y	Y
51	YEARS	M	WHITE	NOT HISPA...	Xan_Lo	Xanomelin...	Xan_Lo	Xanomelin...	USA	2012-12-22	-14				Y	Y	Y
63	YEARS	F	WHITE	NOT HISPA...	Pbo	Placebo	Pbo	Placebo	USA	2013-10-01	-7	Y	Y	Y	Y	Y	Y
54	YEARS	F	WHITE	HISPANIC ...	Scrnfail	Screen Fail...	Scrnfail	Screen Fail...	USA	2014-04-01							
67	YEARS	M	WHITE	NOT HISPA...	Xan_Hi	Xanomelin...	Xan_Lo	Xanomelin...	USA	2013-07-24	-7				Y	Y	Y
81	YEARS	F	BLACK OR ...	NOT HISPA...	Pbo	Placebo	Pbo	Placebo	USA	2013-05-20	-10	Y	Y	Y	Y	Y	Y
74	YEARS	M	WHITE	NOT HISPA...	Scrnfail	Screen Fail...	Scrnfail	Screen Fail...	USA	2013-09-18							

Define-XML metadata:
Name: COMPLT24
Label: Completers of Week 24 Population Flag
Mandatory: No
Datatype: text
Length: 1
CodeList: Y_BLANK
Non-Standard Variable (NSV)

(the DM file is from the famous LZZT CDISC-FDA SDTM pilot)

SDTM-ETL and 'split' domains/datasets

With SDTM-ETL, we never need to "split" datasets, we take care in advance that we never produce huge datasets (larger than 5GB) that we then later need to split. Even for large datasets smaller than 5GB, we need to think about whether such huge datasets are "usable" for reviewers. For example, an LB dataset with millions of rows spread over different categories of lab tests will be hard to analyze for reviewers - they will need to start to filter to make these usable.
Therefore, the better strategy, followed by SDTM-ETL, is to develop different instances of the same domain, usually one dataset per category. This also makes the mapping considerably easier, as data for different categories can come from different sources, e.g. some from EDC, other from electronic data transfers (e.g. in CSV format).

This strategy e.g. leads to different dataset definitions in SDTM-ETL, for example:

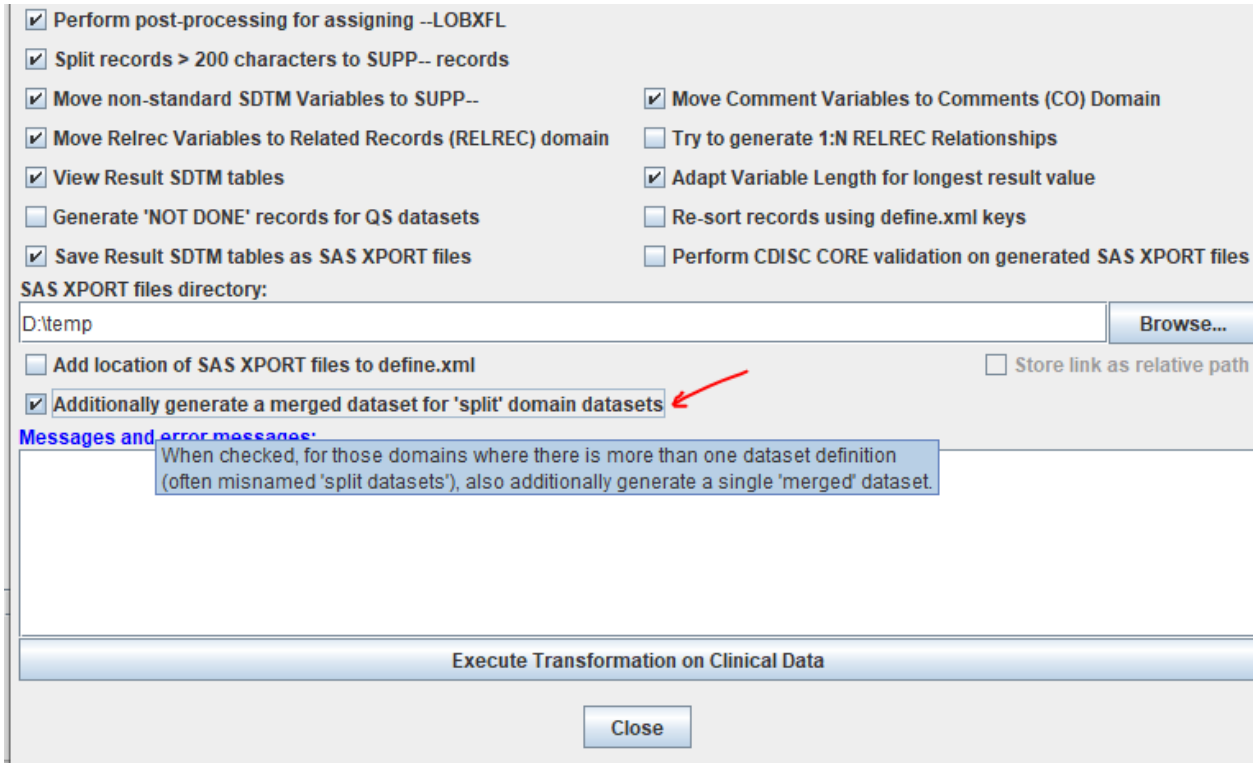
RELREC	STUDYID	RDOMAIN	USUBJID	IDVAR
SUPPQUAL	STUDYID	RDOMAIN	USUBJID	IDVAR
RELSUB	STUDYID	USUBJID	RELSUB.POOLID	RELSUB.RS
OI	STUDYID	DOMAIN	OI.NHOID	OI.OISEQ
[redacted]GLOBAL	STARTREF			
[redacted]LBUR	STUDYID	DOMAIN	USUBJID	LB.LBSEQ
[redacted]LBBL	STUDYID	DOMAIN	USUBJID	LB.LBSEQ
[redacted]LBCO	STUDYID	DOMAIN	USUBJID	LB.LBSEQ
[redacted]LBBR	STUDYID	DOMAIN	USUBJID	LB.LBSEQ

where it was decided to develop mappings and generate 4 instances of the LB (Laboratory) domain: one for urinalysis, one for all all blood tests (chemistry and hematology), one for coagulation tests, and one for breath alcohol tests.
Of course the choice of categories is arbitrary. For example, we could also have chosen to generate separate datasets for hematology and blood chemistry.

With the above setup, developing the mappings will be considerably easy, and lead to 4 submission datasets. In case SAS Transport is used: LBUR.xpt, LBBL.xpt, LBCO.xpt and LBBR.xpt. These will be considerably easier to analyze by the regulatory reviewers than would be the case with one huge LB dataset.

However, these authorities (or in case work is done as a service provider, the sponsor) will also want to see a single LB dataset.

In SDTM-ETL, this can be simply realized during execution of the mappings (when theXPT datasets are generated), by checking the checkbox "*Additionally generate a merged dataset for 'split' domain datasets*".



When it is checked, also a single dataset named "LB.xpt" is generated in the output folder, containing the combined content of the separate LBUR, LBBL, etc. datasets.

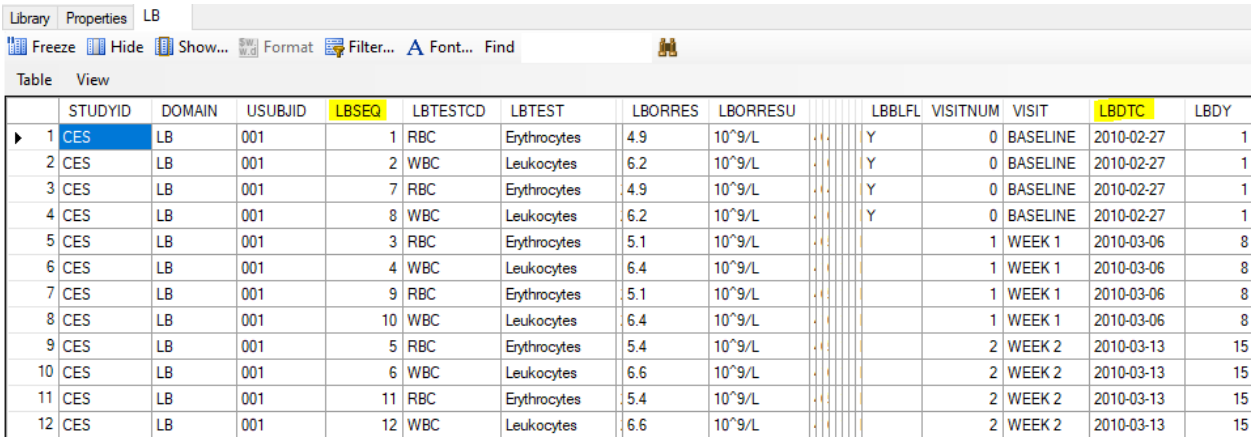
Remark that for QS (Questionnaires), one will always generate distinct datasets, one for each questionnaire, and will never merge these into a single dataset.

Merging datasets and --SEQ values

Often, one will want to have the rows in the datasets being sorted. For example, when the data in the ODM file is not chronologically ordered (though they should), one will often want to have them chronologically within each subject in the SDTM dataset.
in such a case, one will assign define.xml "keys" and, when executing the mappings, taking care that the checkbox "Re-sort records using define.xml keys" checked.

When also merging is requested, e.g. of different LB "instances", the system will try to also re-sort

the "merged" dataset using the same keys[2].
For the "merged" dataset, the result can then e.g. be:



Library Properties LB

Freeze Hide Show... Format Filter... Font... Find

Table View

	STUDYID	DOMAIN	USUBJID	LBSEQ	LBTESTCD	LBTEST	LBORRES	LBORRESU	LBBLFL	VISITNUM	VISIT	LBDTC	LBDY
1	CES	LB	001	1	RBC	Erythrocytes	4.9	10^9/L	Y	0	BASELINE	2010-02-27	1
2	CES	LB	001	2	WBC	Leukocytes	6.2	10^9/L	Y	0	BASELINE	2010-02-27	1
3	CES	LB	001	7	RBC	Erythrocytes	4.9	10^9/L	Y	0	BASELINE	2010-02-27	1
4	CES	LB	001	8	WBC	Leukocytes	6.2	10^9/L	Y	0	BASELINE	2010-02-27	1
5	CES	LB	001	3	RBC	Erythrocytes	5.1	10^9/L		1	WEEK 1	2010-03-06	8
6	CES	LB	001	4	WBC	Leukocytes	6.4	10^9/L		1	WEEK 1	2010-03-06	8
7	CES	LB	001	9	RBC	Erythrocytes	5.1	10^9/L		1	WEEK 1	2010-03-06	8
8	CES	LB	001	10	WBC	Leukocytes	6.4	10^9/L		1	WEEK 1	2010-03-06	8
9	CES	LB	001	5	RBC	Erythrocytes	5.4	10^9/L		2	WEEK 2	2010-03-13	15
10	CES	LB	001	6	WBC	Leukocytes	6.6	10^9/L		2	WEEK 2	2010-03-13	15
11	CES	LB	001	11	RBC	Erythrocytes	5.4	10^9/L		2	WEEK 2	2010-03-13	15
12	CES	LB	001	12	WBC	Leukocytes	6.6	10^9/L		2	WEEK 2	2010-03-13	15

As one can see, the data is also in chronological order (as requested).
However, one also sees that the values for LBSEQ are not subsequent anymore.
The reason is that they have been taken from the source datasets, i.e. copied from the separate instances. In this examples, rows with LBSEQ values from 1 to 6 come from the first instance, whereas the rows with LBSEQ values from 7 to 12 come from the second instance. But as the rows have been resorted according to LBDTC, the values for LBSEQ are not subsequent anymore.
This has caused some controversy.
One of our customers was of the opinion that in such case, the values of --SEQ in the "merged" dataset needs to be reassigned so that they are subsequent.
After a good amount of discussions with all the customers (especially sponsor companies) we decided not to have such a reassignment, for the following reasons:
- there is no rule in SDTM (nor by FDA) that values of --SEQ must be subsequent
- the --SEQ variables have always been meant as identifiers, having to be unique withing each subject within each domain. For reasons of outdated SAS-XPT, they have been defined to integer numbers, but they could essentially also e.g. be UUIDs such as "c18cf1c6-5518-4df2-b272-773bd715b1aa". The later would make more sense. Essentially they are just "artificial keys" and should have no meaning on their own.
- The values of --SEQ are often used as "foreign keys" used by other datasets such as SUPPxx datasets, and the RELREC and CO datasets.
If for the "merged" datasets, we would reassign the values of --SEQ, this would mean they would also be reassigned for the corresponding SUPPxx dataset, which can easily lead to errors.
Even worse, as also in RELREC and CO, the value of --SEQ is often used to point to the corresponding record in the "source" dataset, it would mean that one would need two different RELREC and two different CO datasets, one for use with the "individual" datasets, and one for use with the "merged" dataset. This would of course lead to a lot of confusion in a regulatory submission.
- The main reason however is that one would loose the relation between the "individual" datasets and the "merged" dataset. A record in e.g. LBCH (Lab-chemistry) with USUBJID=001 and LBSEQ=22 does not necessarily correspond anymore with the record with the same value of USUBJID and LBSEQ in the "merged" dataset when such reassignments would be done.
Essentially, this would break traceability.

Of course, if users want to have subsequent values for --SEQ in "merged" datasets, they are of free to do so using other software such as SAS or R, but at their own risk. They should then also be

---

[2] This of course requires that the same "keys" have been assigned to all the "instances" of the same domain.

aware that in such a case, values for IDVARVAL (when IDVAR is a "--SEQ") must also be reassigned in the SUPP--, CO and RELREC datasets.

Merging datasets that were generated separately

The above approach works very well when the different instances of the same domain have been developed together, or are loaded by merging (using the menu "File - Load Study define.xml" followed by "I want to merge with the existing define.xml").

There may be circumstances however, that also this approach does not work.
One such case is that we have two (or more) types of data in SUPPxx:
- SUPPxx dataset with "non-standard variables" automatic split off during mapping execution, and maybe containing values beyond the 200 character limitation.
- custom SUPPxx datasets (derived from the "SUPPQUAL" in the template)

Another example is the case that we have generated different CO (Comments) datasets, by automatic split of ("Comment variable in the dataset definition" generated by "Insert - New SDTM Variable for Comment" - see the tutorial "Auto-generation of comments and putting them in the Comments (CO) domain".

In this tutorial we will take the somewhat more complex example of a SUPPLB dataset that has the "Reason for Event" for each measurement (**one record per measurement** when the field on the CRF was filled), to be combined with a SUPPLB dataset that contains the "abnormality clinical significant" values, with **one record per visit** per group of records (i.e. CRF page).
So, in the first SUPPLB dataset, the identifying variable (IDVAR) will be LBSEQ (Sequence Number), whereas in the second, it will be VISITNUM (Visit Number).

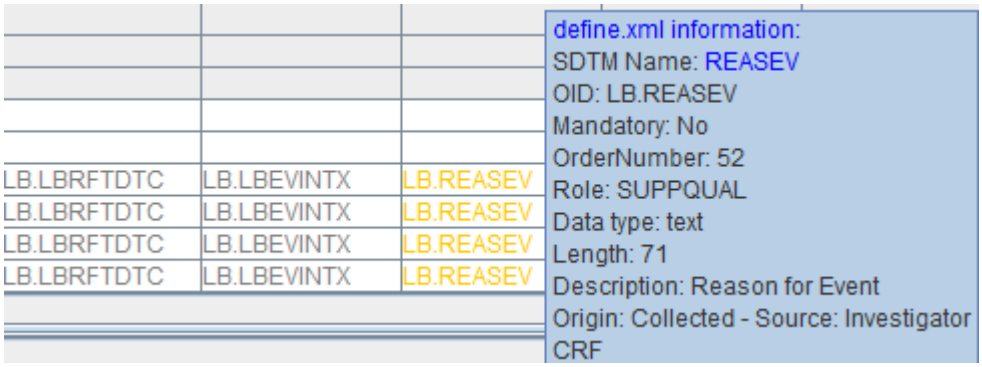
For this new feature, we made a separate application, named "XML2SASDatasetMerger".
It does not merge SAS-XPT datasets directly (that should be done in SAS), but allows to use an XML representation of the output of SDTM-ETL (even when generating XPT files) and merge these. So no (expensive) SAS license is necessary.

Suppose we have already loaded our mappings for LB:

RELSUB	STUDYID	USUBJID	RELSUB.POOLID	RELSUB.RSUB...	RELSUB
OI	STUDYID	DOMAIN	OI.NHOID	OI.OISEQ	OI.OIPAR
[redacted]DM	STUDYID	DOMAIN	USUBJID	SUBJID	DM.RFST
[redacted]GLOBAL	STARTREF				
[redacted]LBUR	STUDYID	DOMAIN	USUBJID	LB.LBSEQ	LB.LBGR
[redacted]LBBL	STUDYID	DOMAIN	USUBJID	LB.LBSEQ	LB.LBGR
[redacted]LBCO	STUDYID	DOMAIN	USUBJID	LB.LBSEQ	LB.LBGR
[redacted]LBBR	STUDYID	DOMAIN	USUBJID	LB.LBSEQ	LB.LBGR

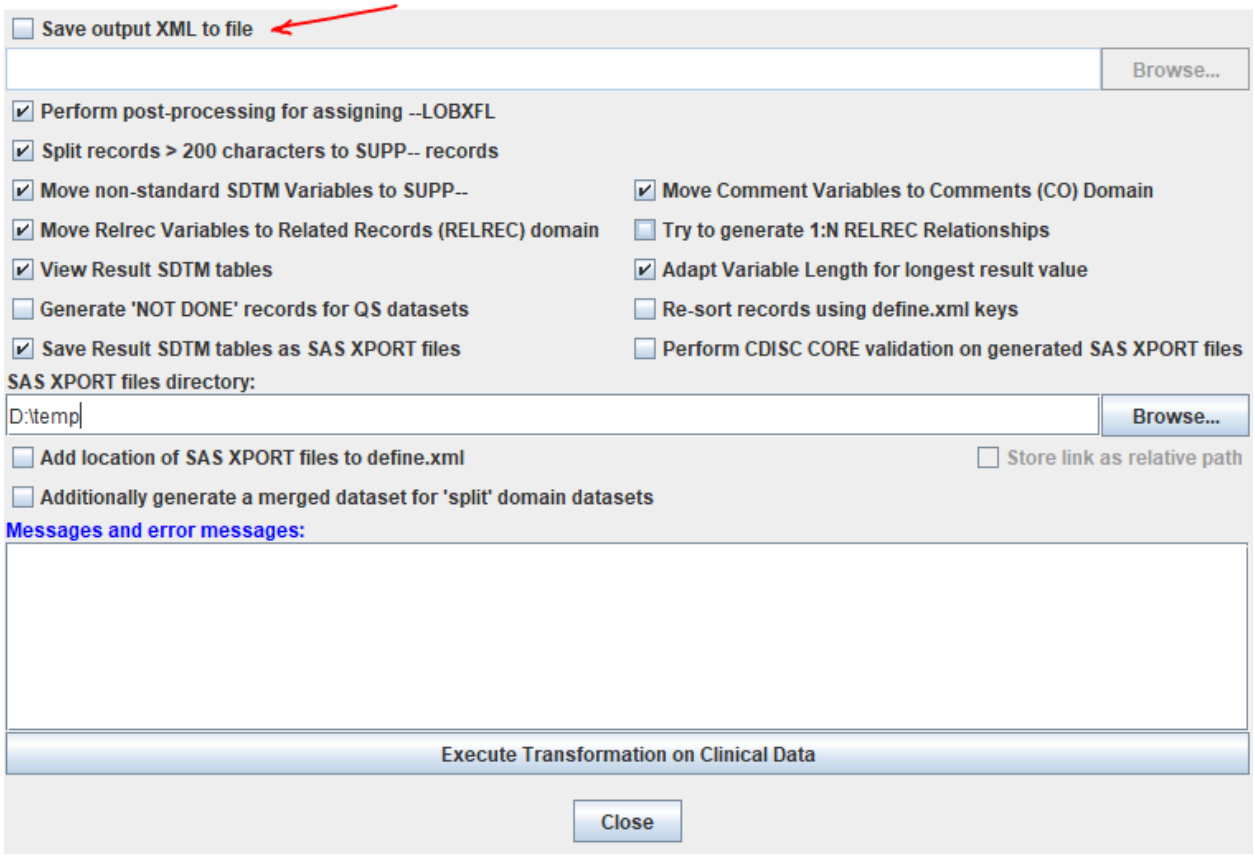
We also loaded the DM dataset, in order to derive the "Last Observation Before First Exposure Flag" (LBLOBXFL) values, as that requires the value of DM.RFXSTDTC (datetime of first exposure).

We also see that an "non-standard variable" (NSV) REASEV" (Reason for Event) has been added to each of the instances:

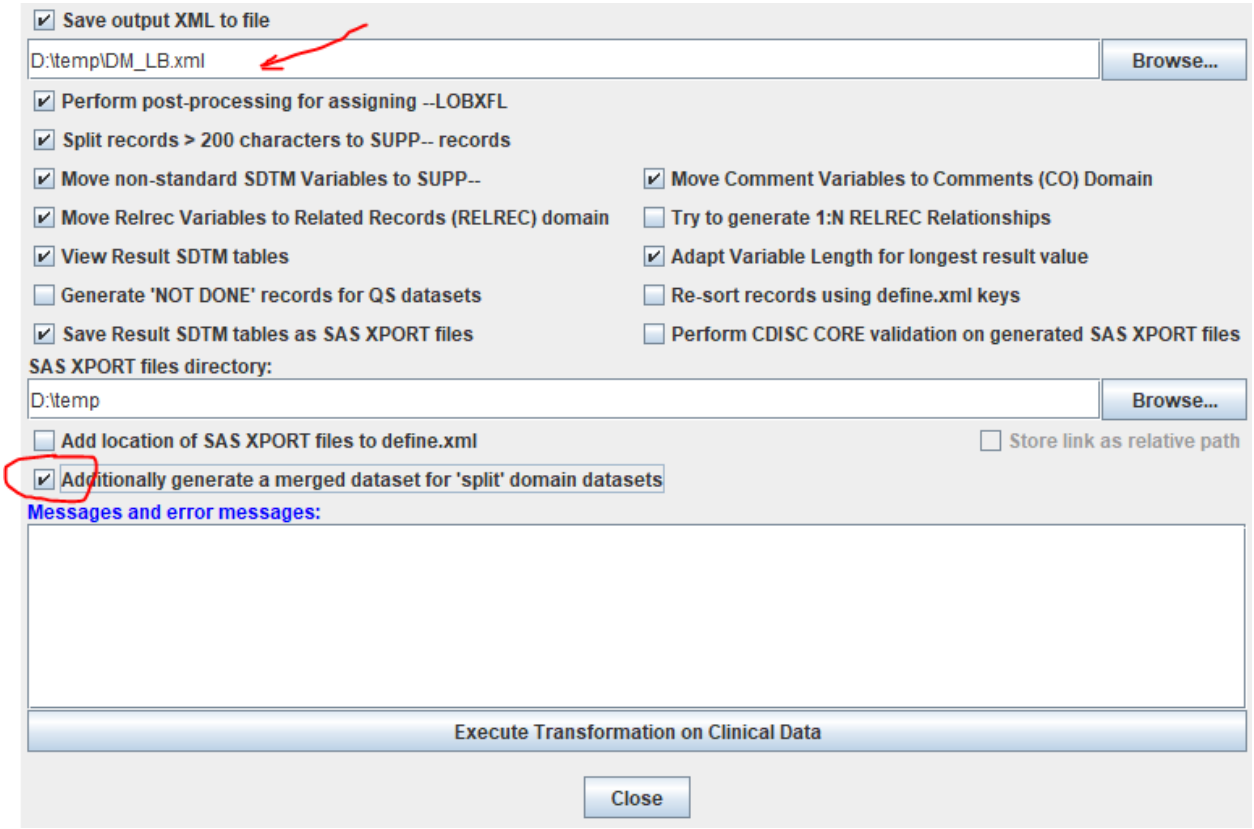


As it is an NSV, at least for regulatory submissions, it must be "banned" to SUPPLB.

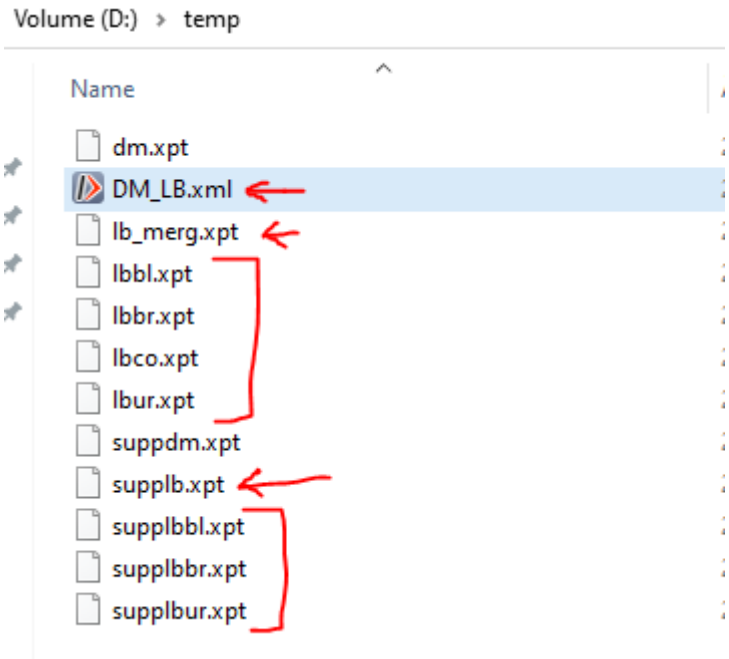
When executing the mappings (using the menu "Transform - Generate Transformation (XSLT) Code for SAS-XPT", and following the wizard, this leads to the dialog:



We see there is a checkbox and field "Save output XML to file:"
Though we are generating SAS-XPT files, we can also save all the datasets into a single XML file, which essentially is a CDISC Dataset-XML file. For example:

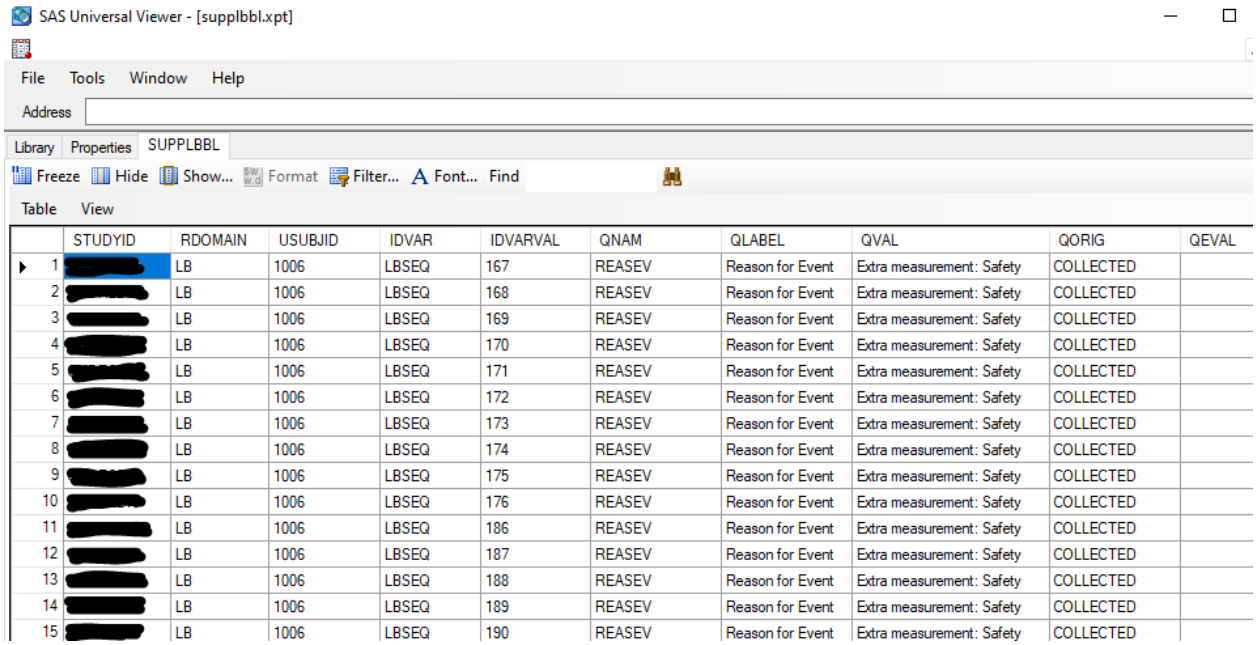


We also check the checkbox "Additionally generate a merged dataset for 'split' domain datasets". After execution, we find the following datasets:



We find the four (distinct) LB datasets: lbbl.xpt (blood tests), lbbr.xpt (breath alcohol tests), lbco.xpt (coagulation tests) and lbur.xpt (urinalysis), and the respective SUPPLB datasets supplbbl.xpt, supplbbr.xpt and supplbur.xpt. There is no supplbco.xpt dataset, as there were no "coagulation" records for which there was a value "Reason for Event".

For example, for supplbbl.xpt, we find:



	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	QNAM	QLABEL	QVAL	QORIG	QEVAL
1	[redacted]	LB	1006	LBSEQ	167	REASEV	Reason for Event	Extra measurement: Safety	COLLECTED	
2	[redacted]	LB	1006	LBSEQ	168	REASEV	Reason for Event	Extra measurement: Safety	COLLECTED	
3	[redacted]	LB	1006	LBSEQ	169	REASEV	Reason for Event	Extra measurement: Safety	COLLECTED	
4	[redacted]	LB	1006	LBSEQ	170	REASEV	Reason for Event	Extra measurement: Safety	COLLECTED	
5	[redacted]	LB	1006	LBSEQ	171	REASEV	Reason for Event	Extra measurement: Safety	COLLECTED	
6	[redacted]	LB	1006	LBSEQ	172	REASEV	Reason for Event	Extra measurement: Safety	COLLECTED	
7	[redacted]	LB	1006	LBSEQ	173	REASEV	Reason for Event	Extra measurement: Safety	COLLECTED	
8	[redacted]	LB	1006	LBSEQ	174	REASEV	Reason for Event	Extra measurement: Safety	COLLECTED	
9	[redacted]	LB	1006	LBSEQ	175	REASEV	Reason for Event	Extra measurement: Safety	COLLECTED	
10	[redacted]	LB	1006	LBSEQ	176	REASEV	Reason for Event	Extra measurement: Safety	COLLECTED	
11	[redacted]	LB	1006	LBSEQ	186	REASEV	Reason for Event	Extra measurement: Safety	COLLECTED	
12	[redacted]	LB	1006	LBSEQ	187	REASEV	Reason for Event	Extra measurement: Safety	COLLECTED	
13	[redacted]	LB	1006	LBSEQ	188	REASEV	Reason for Event	Extra measurement: Safety	COLLECTED	
14	[redacted]	LB	1006	LBSEQ	189	REASEV	Reason for Event	Extra measurement: Safety	COLLECTED	
15	[redacted]	LB	1006	LBSEQ	190	REASEV	Reason for Event	Extra measurement: Safety	COLLECTED	

also showing that each record is related to a single measurement, identified by LBSEQ, in the LB, and in the LBBL dataset.

We also see that a "merged" dataset named "lb_merg.xpt" has been created. This can be renamed to "lb.xpt" if wanted. And there is of course also a "merged" supplb.xpt dataset.

The additional XML dataset "DM_LB.xml" is a Dataset-XML file containing all the records of all the generated datasets, but then in XML format.
As once can see, it's file size is a bit higher than the sum of the file sizes of the XPT files, but not much more, which against contradicts the prejudice of the FDA that Dataset-XML files are much larger in size than XPT files[3].

The XML file "DM_LB.xml" contains all the records for DM, SUPPDM, LBBL, LBBR, LBCO, LBUR, SUPPLBBL, SUPPLBBR, SUPPLBUR, for example:

[3] Also, XML files can be compressed to less than 5% of their normal size, and do not need to be de-compressed in order to be ready by modern software. This was however fully ignored by FDA during the Dataset-XML pilot. Essentially, the file sizes were just an excuse not having to change anything ...

```
189859    <ItemGroupData ItemGroupOID="[redacted]SUPPDM" TransactionType="Insert">
189860      <ItemData ItemOID="STUDYID" Value="[redacted]"/>
189861      <ItemData ItemOID="RDOMAIN" Value="DM"/>
189862      <ItemData ItemOID="USUBJID" Value="4102"/>
189863      <ItemData ItemOID="IDVAR" Value=""/>
189864      <ItemData ItemOID="IDVARVAL" Value=""/>
189865      <ItemData ItemOID="QNAM" Value="SCRNUM"/>
189866      <ItemData ItemOID="QLABEL" Value="Screening Number"/>
189867      <ItemData ItemOID="QVAL" Value="75"/>
189868      <ItemData ItemOID="QORIG" Value="COLLECTED"/>
189869      <ItemData ItemOID="QEVAL" Value=""/>
189870    </ItemGroupData>
189871    <ItemGroupData ItemGroupOID="[redacted]SUPPLBUR" TransactionType="Insert">
189872      <ItemData ItemOID="STUDYID" Value="[redacted]"/>
189873      <ItemData ItemOID="RDOMAIN" Value="LB"/>
189874      <ItemData ItemOID="USUBJID" Value="4005"/>
189875      <ItemData ItemOID="IDVAR" Value="LBSEQ"/>
189876      <ItemData ItemOID="IDVARVAL" Value="53"/>
189877      <ItemData ItemOID="QNAM" Value="REASEV"/>
189878      <ItemData ItemOID="QLABEL" Value="Reason for Event"/>
189879      <ItemData ItemOID="QVAL" Value="Former measurement: Reserve subject"/>
189880      <ItemData ItemOID="QORIG" Value="COLLECTED"/>
189881      <ItemData ItemOID="QEVAL" Value=""/>
189882    </ItemGroupData>
```

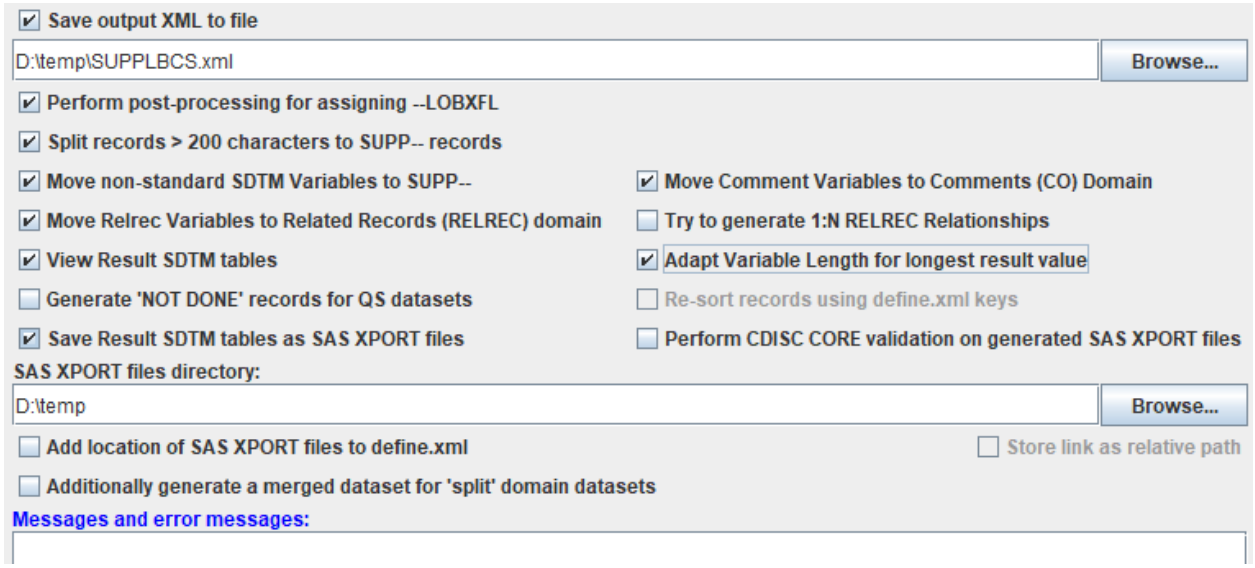
each record having an identifier XXX:yyyy where XXX is the StudyID (hidden here) and yyy is the dataset identifier.
However, we also needed to generate a separate SUPPLBCS dataset ("CS" standing for "Clinically Significant"), as we have a field like "(Clinically significant) XXX lab value abnormalities?" which is only asked once per visit, and where XXX can e.g. be "glucose", "coagulation", "hematology". As these are not directly related to a single measurement value, we choose to generate the SUPPLBCS dataset with the identifying variable IDVAR=VISITNUM[4].

We generated this SUPPLBCS dataset separately in SDTM-ETL, using the menu "Insert - Domain specific SUPPQUAL", and then used QNAM as the variable to iterate over, i.e. a record is generated for each entry in the ODM dataset for which there is a mapping defined in QNAM:

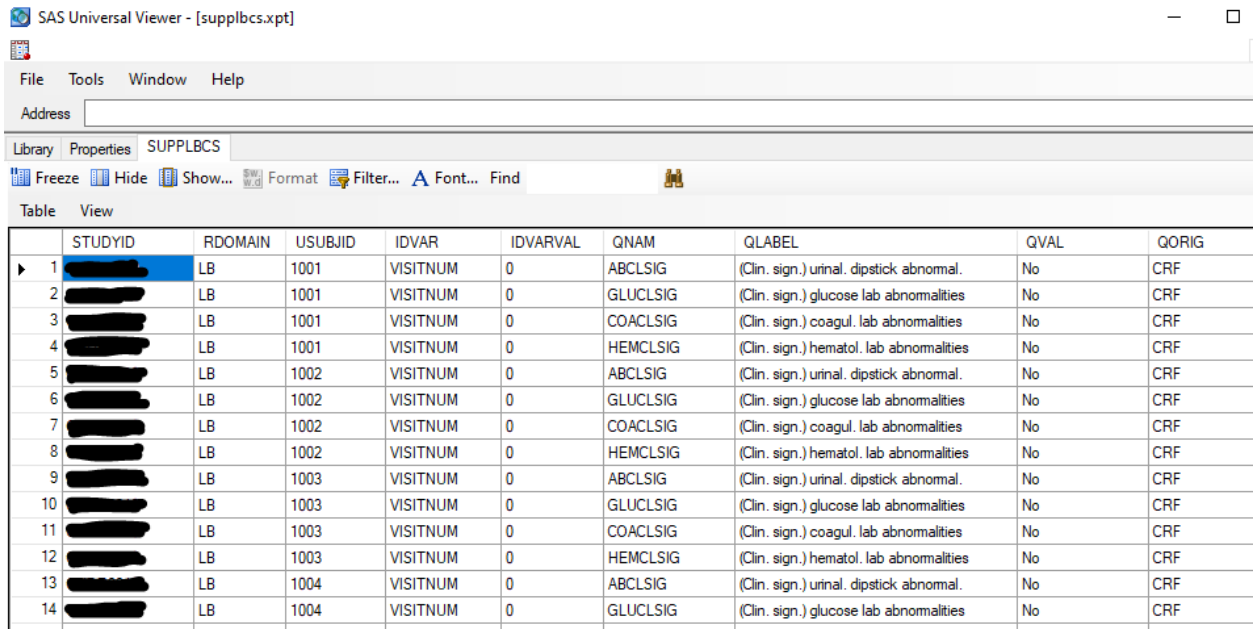
RELREC	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	RELTYPE	RELID	
SUPPQUAL	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	SUPPQUAL.QNAM	SUPPQUAL.QLABEL	SU
RELSUB	STUDYID	USUBJID	RELSUB.POOLID	RELSUB.RSUB...	RELSUB.SREL			
OI	STUDYID	DOMAIN	OI.NHOID	OI.OISEQ	OI.OIPARMCD	OI.OIPARM	OI.OIVAL	
[redacted]SUPPLBCS	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	SUPPQUAL.QNAM	SUPPQUAL.QLABEL	SU

Also here, when we generate the XPT datasets, we also want to save a copy in XML format:

[4] Another solution would have been to add a variable LBGRPID in each of the LB instances, and reference that.



and the generated XPT (supplbcs.xpt) file looks like:



	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	QNAM	QLABEL	QVAL	QORIG
1	[redacted]	LB	1001	VISITNUM	0	ABCLSIG	(Clin. sign.) urinal. dipstick abnormal.	No	CRF
2	[redacted]	LB	1001	VISITNUM	0	GLUCLSIG	(Clin. sign.) glucose lab abnormalities	No	CRF
3	[redacted]	LB	1001	VISITNUM	0	COACLSIG	(Clin. sign.) coagul. lab abnormalities	No	CRF
4	[redacted]	LB	1001	VISITNUM	0	HEMCLSIG	(Clin. sign.) hematol. lab abnormalities	No	CRF
5	[redacted]	LB	1002	VISITNUM	0	ABCLSIG	(Clin. sign.) urinal. dipstick abnormal.	No	CRF
6	[redacted]	LB	1002	VISITNUM	0	GLUCLSIG	(Clin. sign.) glucose lab abnormalities	No	CRF
7	[redacted]	LB	1002	VISITNUM	0	COACLSIG	(Clin. sign.) coagul. lab abnormalities	No	CRF
8	[redacted]	LB	1002	VISITNUM	0	HEMCLSIG	(Clin. sign.) hematol. lab abnormalities	No	CRF
9	[redacted]	LB	1003	VISITNUM	0	ABCLSIG	(Clin. sign.) urinal. dipstick abnormal.	No	CRF
10	[redacted]	LB	1003	VISITNUM	0	GLUCLSIG	(Clin. sign.) glucose lab abnormalities	No	CRF
11	[redacted]	LB	1003	VISITNUM	0	COACLSIG	(Clin. sign.) coagul. lab abnormalities	No	CRF
12	[redacted]	LB	1003	VISITNUM	0	HEMCLSIG	(Clin. sign.) hematol. lab abnormalities	No	CRF
13	[redacted]	LB	1004	VISITNUM	0	ABCLSIG	(Clin. sign.) urinal. dipstick abnormal.	No	CRF
14	[redacted]	LB	1004	VISITNUM	0	GLUCLSIG	(Clin. sign.) glucose lab abnormalities	No	CRF

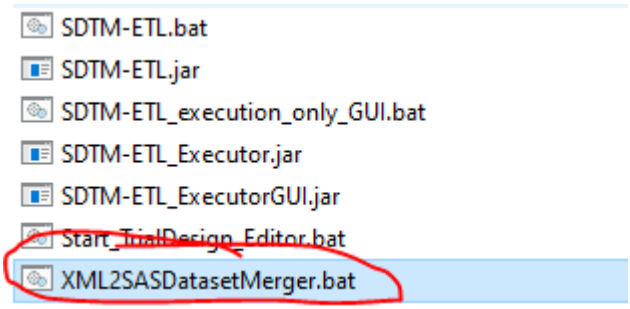
The corresponding XML file SUPPLBCS.xml has e.g.:

```
<ItemGroupData ItemGroupOID="[redacted]SUPPLBCS" TransactionType="Insert">
  <ItemData ItemOID="STUDYID" Value="[redacted]"/>
  <ItemData ItemOID="RDOMAIN" Value="LB"/>
  <ItemData ItemOID="USUBJID" Value="1001"/>
  <ItemData ItemOID="IDVAR" Value="VISITNUM"/>
  <ItemData ItemOID="IDVARVAL" Value="0"/>
  <ItemData ItemOID="SUPPQUAL.QNAM" Value="ABCLSIG"/>
  <ItemData ItemOID="SUPPQUAL.QLABEL" Value="(Clin. sign.) urinal. dipstick abnormal."/>
  <ItemData ItemOID="SUPPQUAL.QVAL" Value="No"/>
  <ItemData ItemOID="SUPPQUAL.QORIG" Value="CRF"/>
  <ItemData ItemOID="SUPPQUAL.QEVAL" IsNull="Yes"/>
</ItemGroupData>
<ItemGroupData ItemGroupOID="[redacted]SUPPLBCS" TransactionType="Insert">
  <ItemData ItemOID="STUDYID" Value="[redacted]"/>
  <ItemData ItemOID="RDOMAIN" Value="LB"/>
  <ItemData ItemOID="USUBJID" Value="1001"/>
  <ItemData ItemOID="IDVAR" Value="VISITNUM"/>
  <ItemData ItemOID="IDVARVAL" Value="0"/>
  <ItemData ItemOID="SUPPQUAL.QNAM" Value="GLUCLSIG"/>
  <ItemData ItemOID="SUPPQUAL.QLABEL" Value="(Clin. sign.) glucose lab abnormalities"/>
  <ItemData ItemOID="SUPPQUAL.QVAL" Value="No"/>
  <ItemData ItemOID="SUPPQUAL.QORIG" Value="CRF"/>
  <ItemData ItemOID="SUPPQUAL.QEVAL" IsNull="Yes"/>
</ItemGroupData>
```

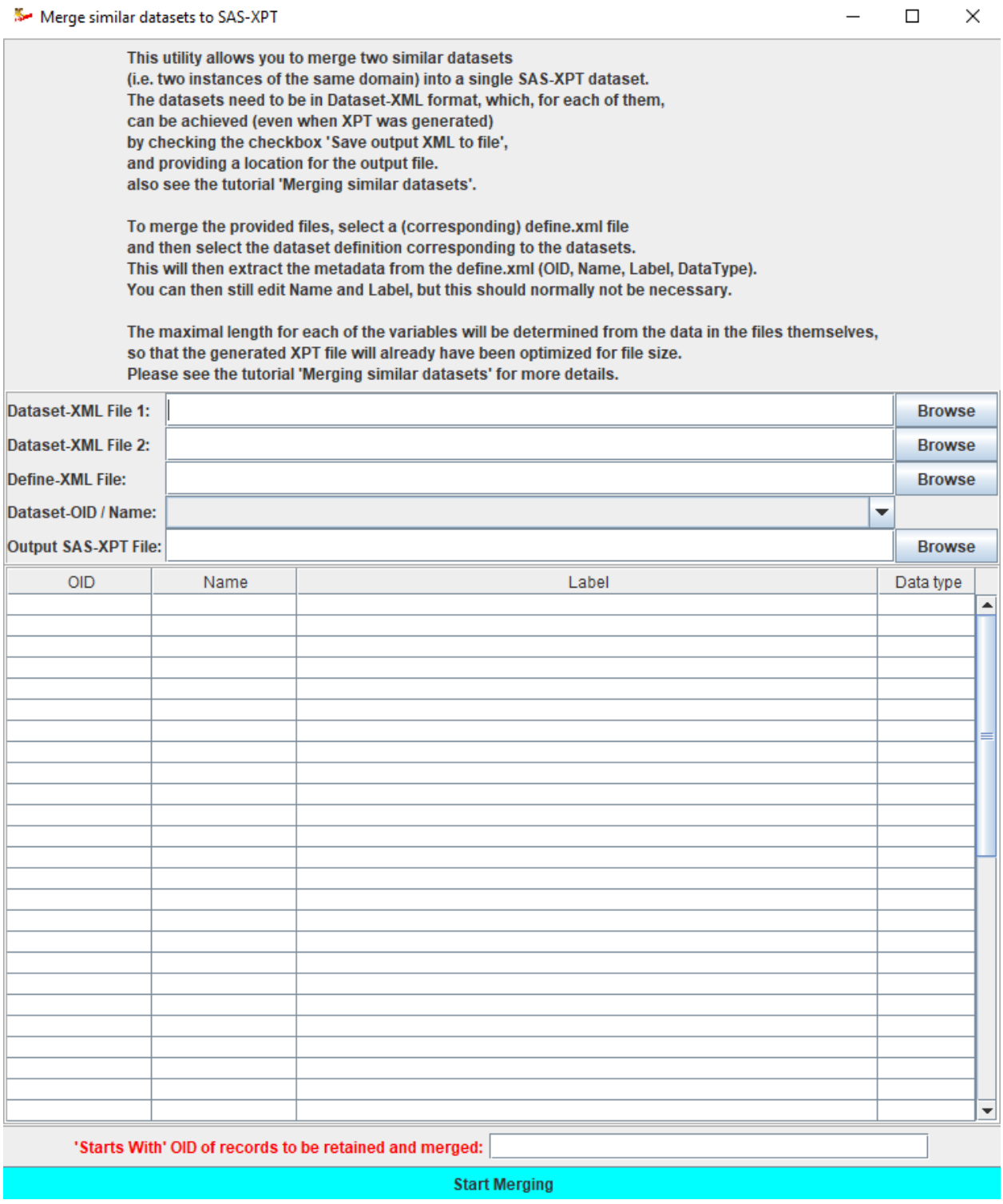
Suppose now we want to generate a single SUPPLB-XPT file that contains as well the "REASEV" Reason for Event (REASEV) as the "clinical significance of xxx abnormalities". This means that we need to merge the SUPPLB records from both files.

We will not try to merge the XPT datasets directly (this would require an over-expensive SAS license), but we will use our new program "XML2SASDatasetMerger" which comes with the new SDTM-ETL v.4.3 distribution.

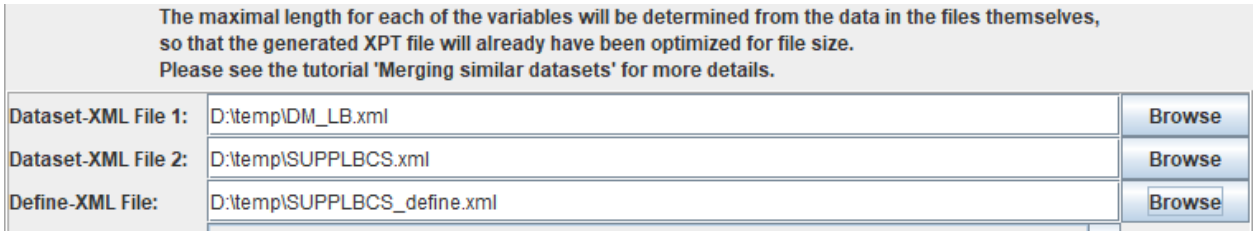
It can be started from the distribution folder by double-clicking (when using Windows) "XML2SASDatasetMerger.bat":



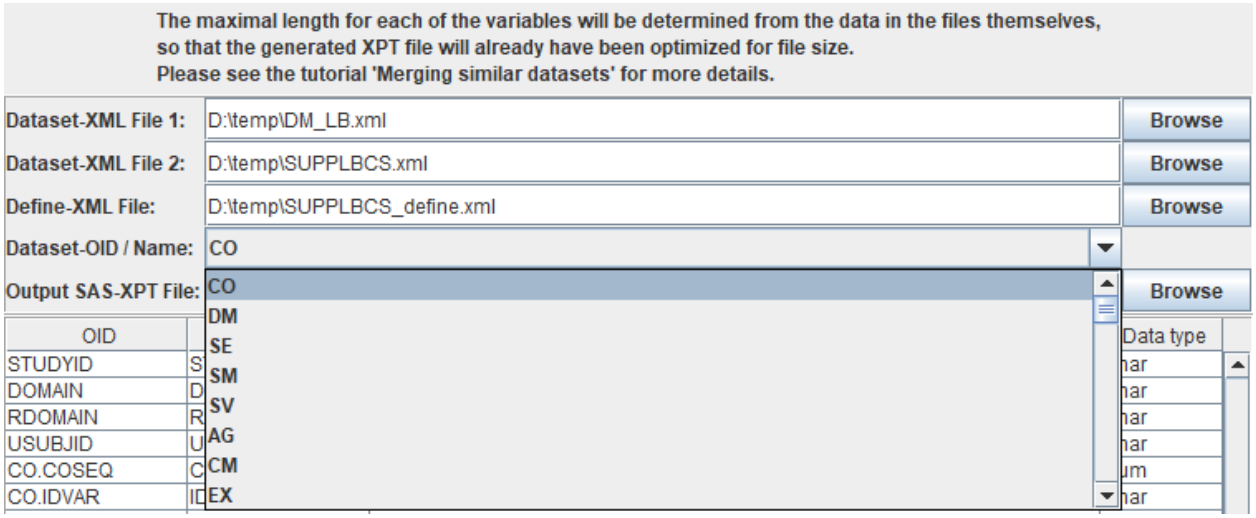
When executed, a new window is opened:



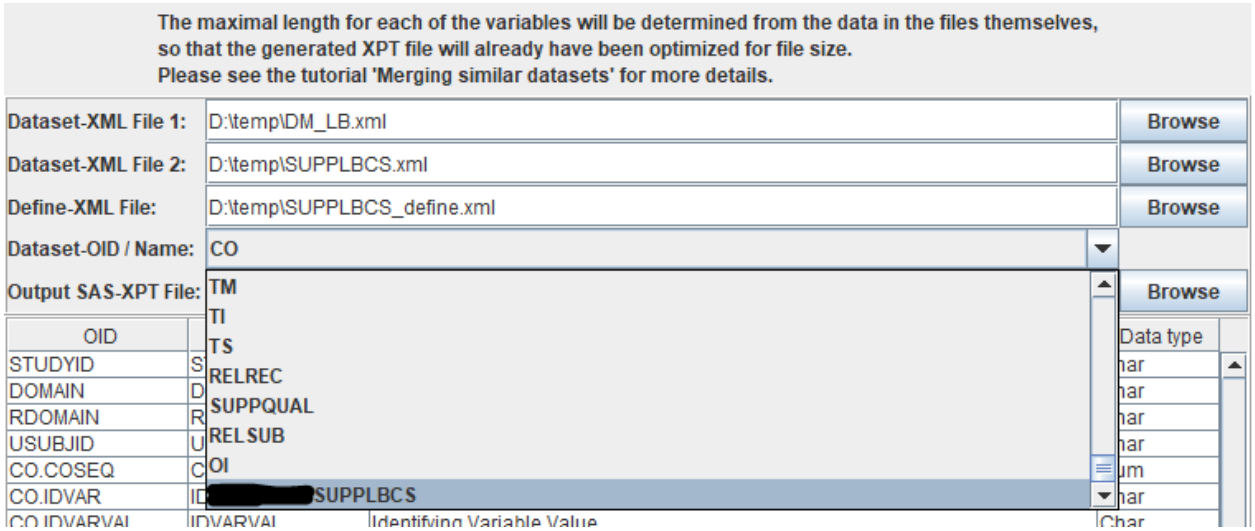
We can already fill the location of the two XML files, and use the define.xml file (needed in order to find variable names labels that need to go into the XPT dataset):



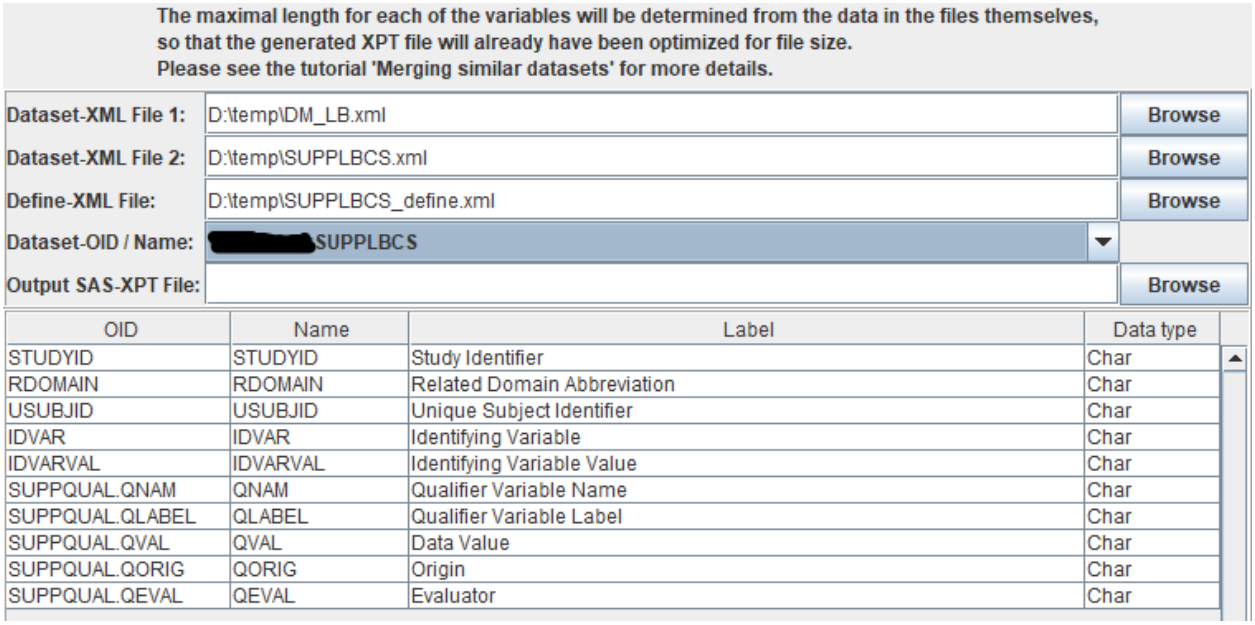
As soon as we add the location of the define.xml file, the combobox "Dataset OID / Name" presents a list with all the dataset definitions found in that define.xml:



We scroll down, and look for SUPPLBxx, and select it:

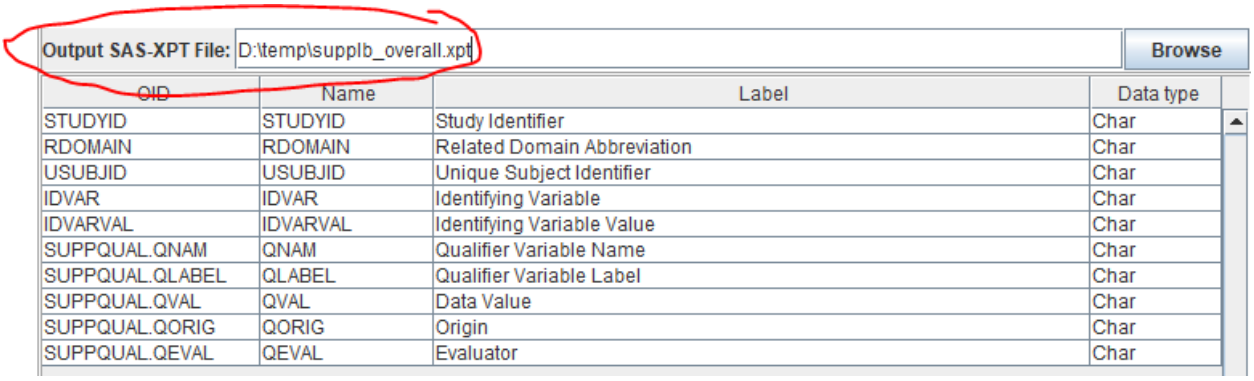


with "SUPPLBCS" prefixed by the StudyID (hidden here).
Immediately, the table lower down updates to:



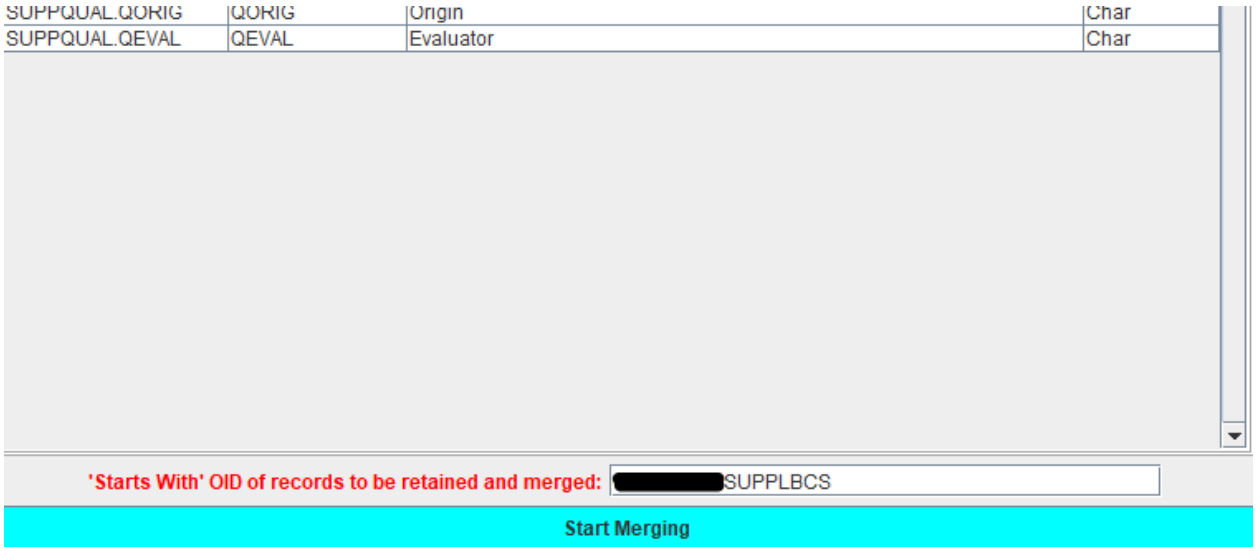
showing all the variables and their properties. This table will then be used to assign the "labels" in the XPT dataset.

We then also add the name and location of the XPT dataset we want to produce, e.g.:



The next step is a very important, as it allows to steer the selection of the records that will go into the output XPT dataset.

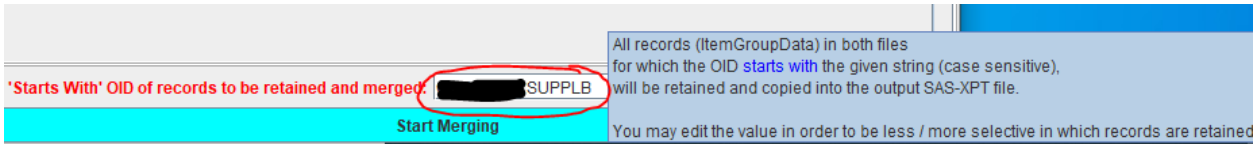
In the lower field we find:



with XXX:SUPPLBCS just being a first proposal ...
and XXX being the StudyID (hidden here).

As we have seen before, all our "REASEV" records in the Dataset-XML file "DM_LB.xml" have the identifier (OID) either being "XXX:SUPPLBBL", "XXX:SUPPLBBR", or "XXX:SUPPLBUR",
and the one for the "Significance" records in dataset "SUPPLBCS.xml" have the identifier "XXX:SUPPLBCS".

The "Starts With" field now allows us to select which records go into the XPT dataset that we want to generate. If we leave it being "XXX:SUPPLBCS", we will get a SUPPLB dataset that only has "Clinical Significant" records. So, we change it into:



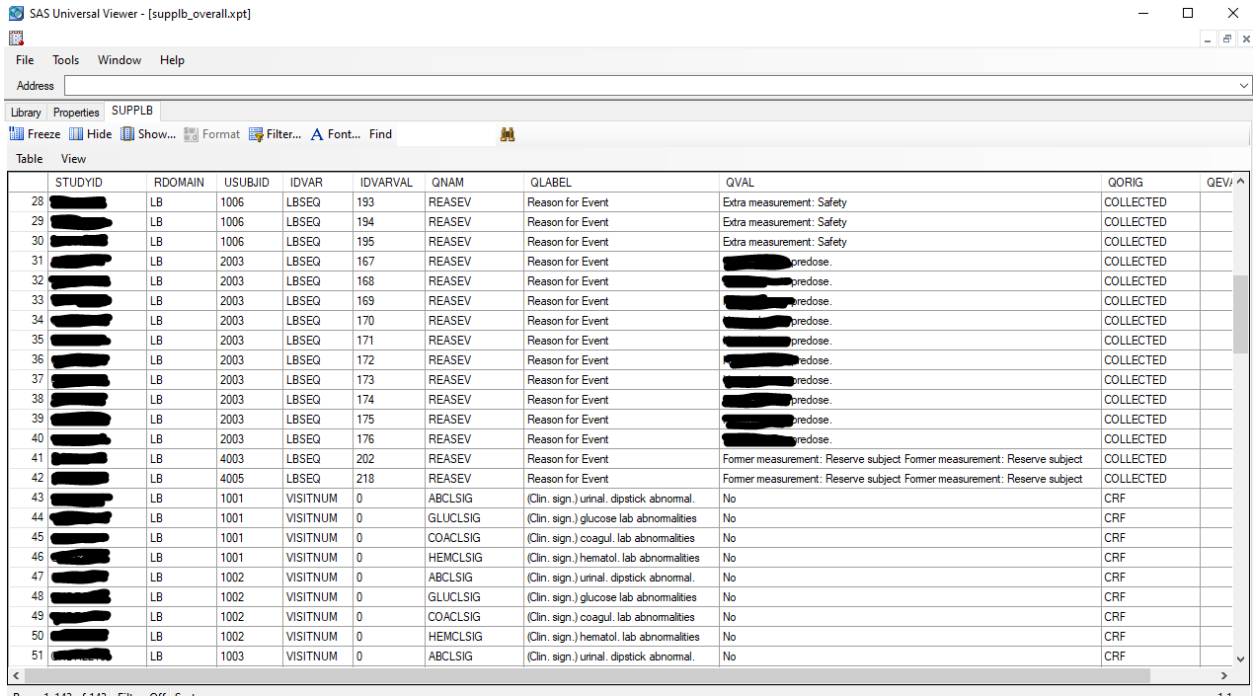
stating that all records for which the identifier "starts with" "XXX:SUPPLB" will be selected, which means all "REASEV" records in file "DM_LB.xml" as well as all "Significance" records in the file "SUPPLBCS.xml".

When then clicking "Start Merging", the software starts the following process:
- it first analyzes the length for each value of each variable from the XML files, in order to determine the maximal value length, that SAS-XPT requires[5].
- it then selects the records from each dataset and combines them into a single XML file (if you want to see it, it is named "temp_merge.xml" and is in the "temp" folder of your SDTM-ETL distribution folder).
- it then takes this combined XML dataset, and with the given labels and determined maximal lengths for the variables, and generates the SAS-XPT dataset and writes it in the output file.

The result in this case e.g. is:

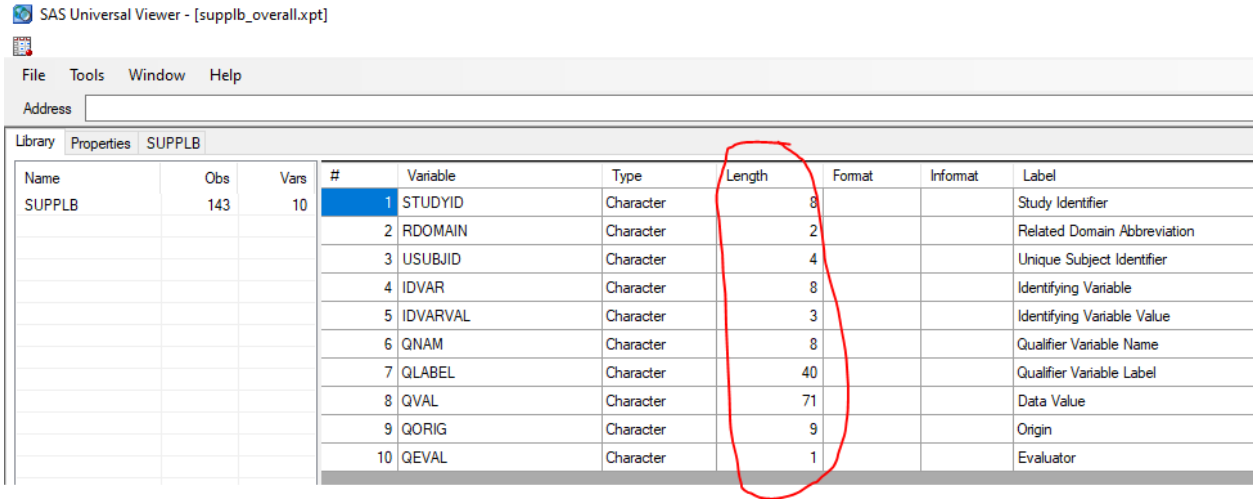
---

[5] "Maximal length"of the variable value would be completely irrelevant when working with modern data formats like XML, JSON, YAML, ...



	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	QNAM	QLABEL	QVAL	QORIG	QEVA
28	[redacted]	LB	1006	LBSEQ	193	REASEV	Reason for Event	Extra measurement: Safety	COLLECTED	
29	[redacted]	LB	1006	LBSEQ	194	REASEV	Reason for Event	Extra measurement: Safety	COLLECTED	
30	[redacted]	LB	1006	LBSEQ	195	REASEV	Reason for Event	Extra measurement: Safety	COLLECTED	
31	[redacted]	LB	2003	LBSEQ	167	REASEV	Reason for Event	[redacted]predose.	COLLECTED	
32	[redacted]	LB	2003	LBSEQ	168	REASEV	Reason for Event	[redacted]predose.	COLLECTED	
33	[redacted]	LB	2003	LBSEQ	169	REASEV	Reason for Event	[redacted]predose.	COLLECTED	
34	[redacted]	LB	2003	LBSEQ	170	REASEV	Reason for Event	[redacted]predose.	COLLECTED	
35	[redacted]	LB	2003	LBSEQ	171	REASEV	Reason for Event	[redacted]predose.	COLLECTED	
36	[redacted]	LB	2003	LBSEQ	172	REASEV	Reason for Event	[redacted]predose.	COLLECTED	
37	[redacted]	LB	2003	LBSEQ	173	REASEV	Reason for Event	[redacted]predose.	COLLECTED	
38	[redacted]	LB	2003	LBSEQ	174	REASEV	Reason for Event	[redacted]predose.	COLLECTED	
39	[redacted]	LB	2003	LBSEQ	175	REASEV	Reason for Event	[redacted]predose.	COLLECTED	
40	[redacted]	LB	2003	LBSEQ	176	REASEV	Reason for Event	[redacted]predose.	COLLECTED	
41	[redacted]	LB	4003	LBSEQ	202	REASEV	Reason for Event	Former measurement: Reserve subject Former measurement: Reserve subject	COLLECTED	
42	[redacted]	LB	4005	LBSEQ	218	REASEV	Reason for Event	Former measurement: Reserve subject Former measurement: Reserve subject	COLLECTED	
43	[redacted]	LB	1001	VISITNUM	0	ABCLSIG	(Clin. sign.) urinal. dipstick abnormal.	No	CRF	
44	[redacted]	LB	1001	VISITNUM	0	GLUCLSIG	(Clin. sign.) glucose lab abnormalities	No	CRF	
45	[redacted]	LB	1001	VISITNUM	0	COACLSIG	(Clin. sign.) coagul. lab abnormalities	No	CRF	
46	[redacted]	LB	1001	VISITNUM	0	HEMCLSIG	(Clin. sign.) hematol. lab abnormalities	No	CRF	
47	[redacted]	LB	1002	VISITNUM	0	ABCLSIG	(Clin. sign.) urinal. dipstick abnormal.	No	CRF	
48	[redacted]	LB	1002	VISITNUM	0	GLUCLSIG	(Clin. sign.) glucose lab abnormalities	No	CRF	
49	[redacted]	LB	1002	VISITNUM	0	COACLSIG	(Clin. sign.) coagul. lab abnormalities	No	CRF	
50	[redacted]	LB	1002	VISITNUM	0	HEMCLSIG	(Clin. sign.) hematol. lab abnormalities	No	CRF	
51	[redacted]	LB	1003	VISITNUM	0	ABCLSIG	(Clin. sign.) urinal. dipstick abnormal.	No	CRF	

And as one can see from the properties, the combined SUPPLB file (supplb_overall.xpt) has been optimized for "minimizing" the file size:



Name	Obs	Vars
SUPPLB	143	10

#	Variable	Type	Length	Format	Informat	Label
1	STUDYID	Character	8			Study Identifier
2	RDOMAIN	Character	2			Related Domain Abbreviation
3	USUBJID	Character	4			Unique Subject Identifier
4	IDVAR	Character	8			Identifying Variable
5	IDVARVAL	Character	3			Identifying Variable Value
6	QNAM	Character	8			Qualifier Variable Name
7	QLABEL	Character	40			Qualifier Variable Label
8	QVAL	Character	71			Data Value
9	QORIG	Character	9			Origin
10	QEVAL	Character	1			Evaluator

We have demonstrated the XML2SASDatasetMerger software for the more complicated case of merging two different types of SUPPLB datasets. The same approach can however be used for merging e.g. different instances of CO (Comments) that have been generated separately for different domains in different datasets.
The only current limitation is that the datasets to be merged have the same (number of) variables. For example, it will usually not allow to e.g. merge an LB dataset with an MB dataset.

Conclusions

The by FDA and other regulatory authorities mandated (but outdated) SAS-XPT format makes life unnecessary complicated when generating CDISC SDTM and SEND datasets.
Essentially, "splitting" and "merging" datasets should never have to be done, and if so anyway, it should be easy to accomplish. It is the SAS-XPT format that makes it complicated: when using XML (e.g. Dataset-XML) or JSON (e.g. Dataset-JSON), this is much easier to accomplish.

This tutorial demonstrated two ways of merging datasets for the case of XPT as the output format. For the simple case that several instances of the same domain are present in the same "working" define.xml, one can set the "*Additionally generate a merged dataset for 'split' domain datasets"* checkbox.
In case datasets to be merged have been created separately, one can use the new utility tool (as of SDTM-ETL v.4.3) named "XML2SASDatasetMerger" that comes as a separate program.

We expect that FDA will give green light for Dataset-JSON submissions by the end of this year. As soon as that is clear, we will also provide functionalities for merging Dataset-JSON files.
Developing these will however be much more straightforward than for SAS-XPT.