SDTM-ETL 4.3: Summary of New Features

Author: Jozef Aerts, XML4Pharma

Last update: **2023-09-13**

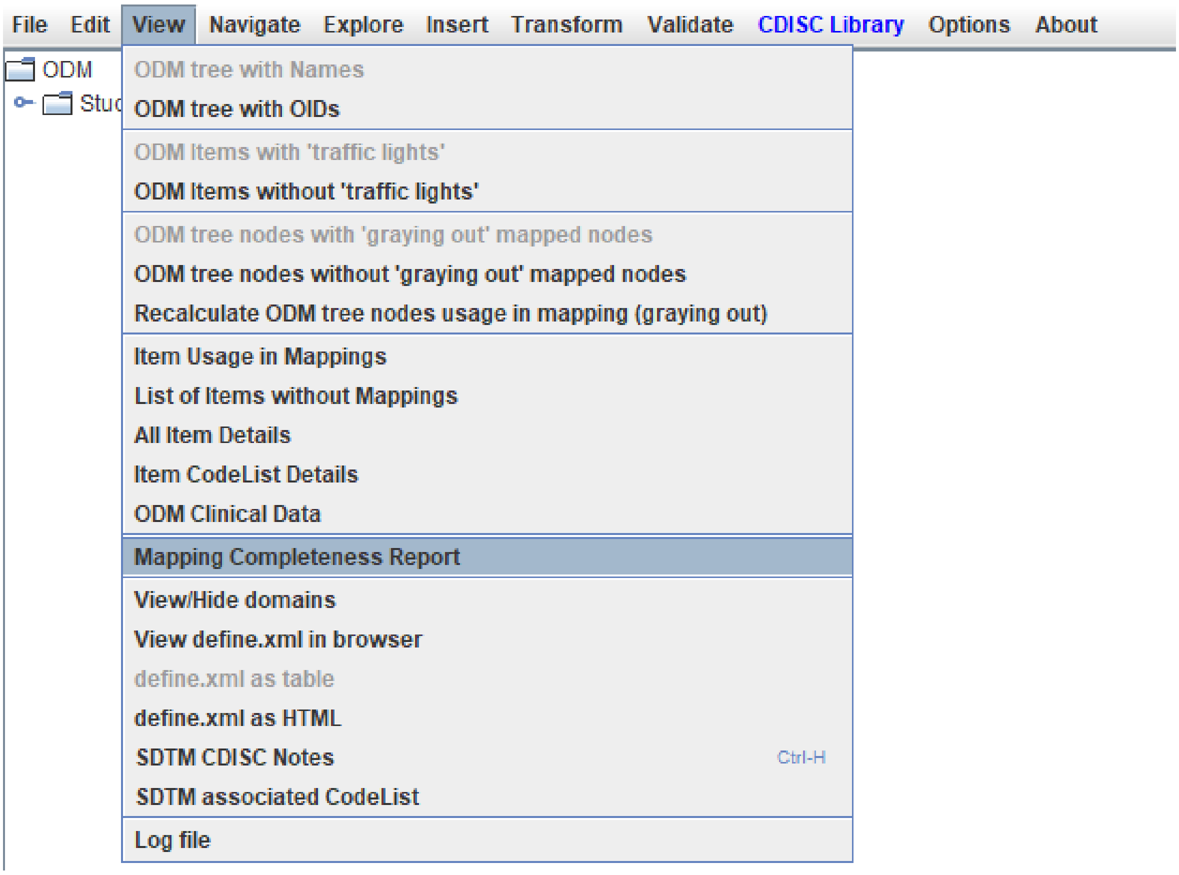
Summary

This document contains a summary of the most important new features of SDTM-ETL 4.3 and bug fixes.
There are many minor improvements and new features that are not described in this document, but that can be found in other manuals / tutorials of SDTM-ETL 4.3.

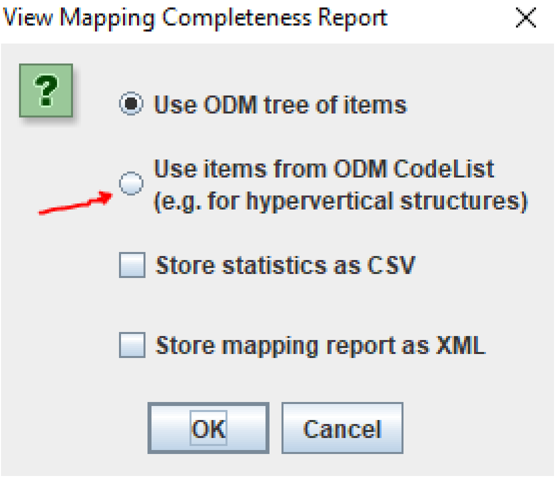
Table of Contents

Mapping Completeness report for hypervertical ODM structures

SDTM-ETL 4.2 introduced a number of new features for the case of "hypervertical" ODM structures, i.e. where the ItemGroups contain items for respectively "parameter name", "parameter description" and "parameter value" (EAV - Entity-Attribute-Value system).
As in such structures, the list of tests performed (or questions asked) is not in individual ItemDefs anymore, but in a codelists referenced by the "parameter name" ItemDef, another approach is needed when one wants to obtain which tests and questions have been used in mappings, this in order to find out whether the mappings is "complete". The menu used for obtaining such a "completeness report" for this is "View - Mapping Completeness Report":



When used, and new in SDTM-ETL 4.3 is, that a new wizard is displayed, asking the user whether to use the (classic way) of using the ODM tree of the items, or (new) to use the items from a codelists, which is usually the case when working with hypervertical structures.

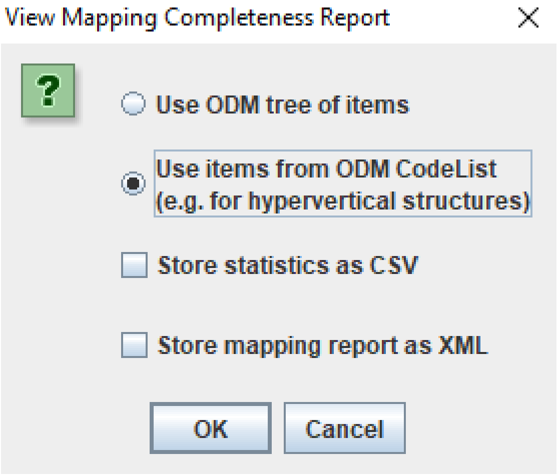

View Mapping Completeness Report

● Use ODM tree of items

○ Use items from ODM CodeList (e.g. for hypervertical structures)

□ Store statistics as CSV

□ Store mapping report as XML


View Mapping Completeness Report

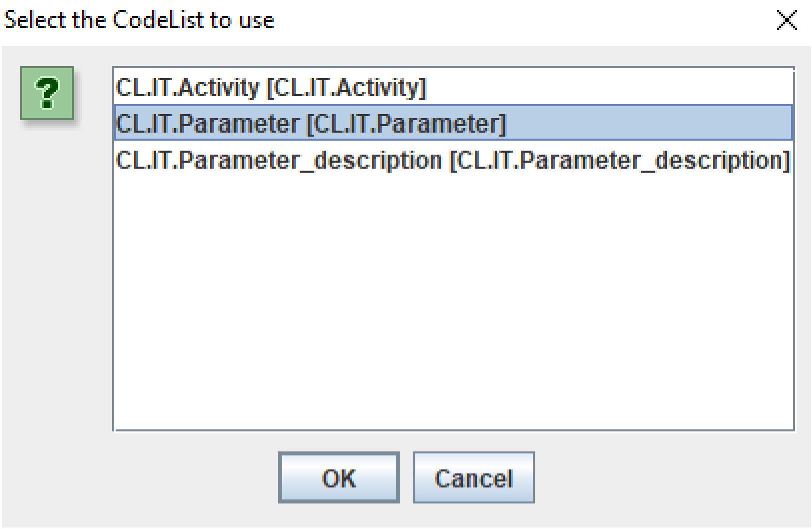
○ Use ODM tree of items

● Use items from ODM CodeList (e.g. for hypervertical structures)

□ Store statistics as CSV

□ Store mapping report as XML

When "Use Item from ODM CodeList" is selected, the user is asked to select the appropriate codelist, e.g.:

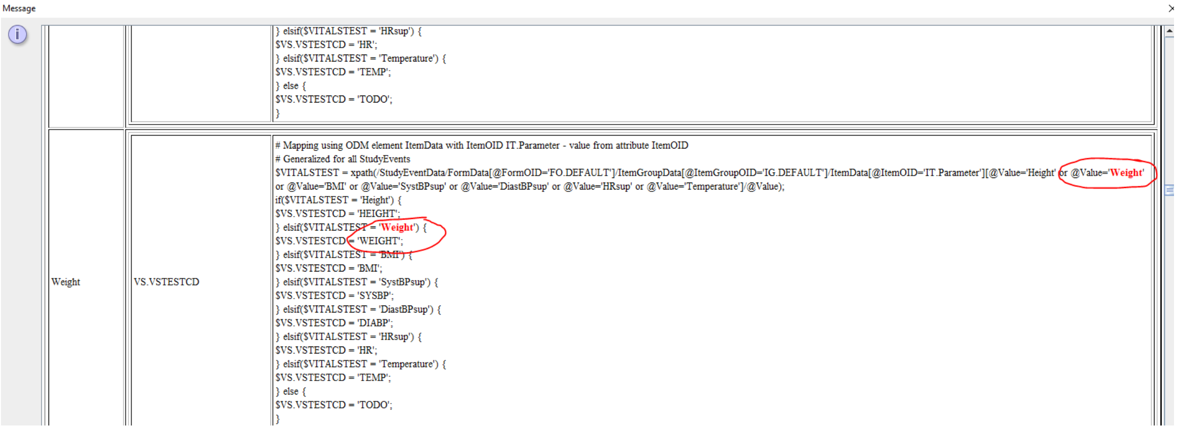


CL.IT.Activity [CL.IT.Activity]
CL.IT.Parameter [CL.IT.Parameter]
CL.IT.Parameter_description [CL.IT.Parameter_description]

In our case, the codelist with OID "CL.IT.Parameter" contains the list of all tests and questions, e.g.:

```
<CodeListItem CodedValue="BMI">
    <Decode>
        <TranslatedText xml:lang="en">BMI</TranslatedText>
    </Decode>
</CodeListItem>
<CodeListItem CodedValue="Height">
    <Decode>
        <TranslatedText xml:lang="en">Height</TranslatedText>
    </Decode>
</CodeListItem>
<CodeListItem CodedValue="Weight">
    <Decode>
        <TranslatedText xml:lang="en">Weight</TranslatedText>
    </Decode>
</CodeListItem>
<CodeListItem CodedValue="ECG">
    <Decode>
        <TranslatedText xml:lang="en">ECG</TranslatedText>
    </Decode>
</CodeListItem>
```

After the appropriate codelist is selected, and "OK" is clicked, the "mapping completeness report" is assembled (which can take a bit of time), and displayed to the user. For example:



Message

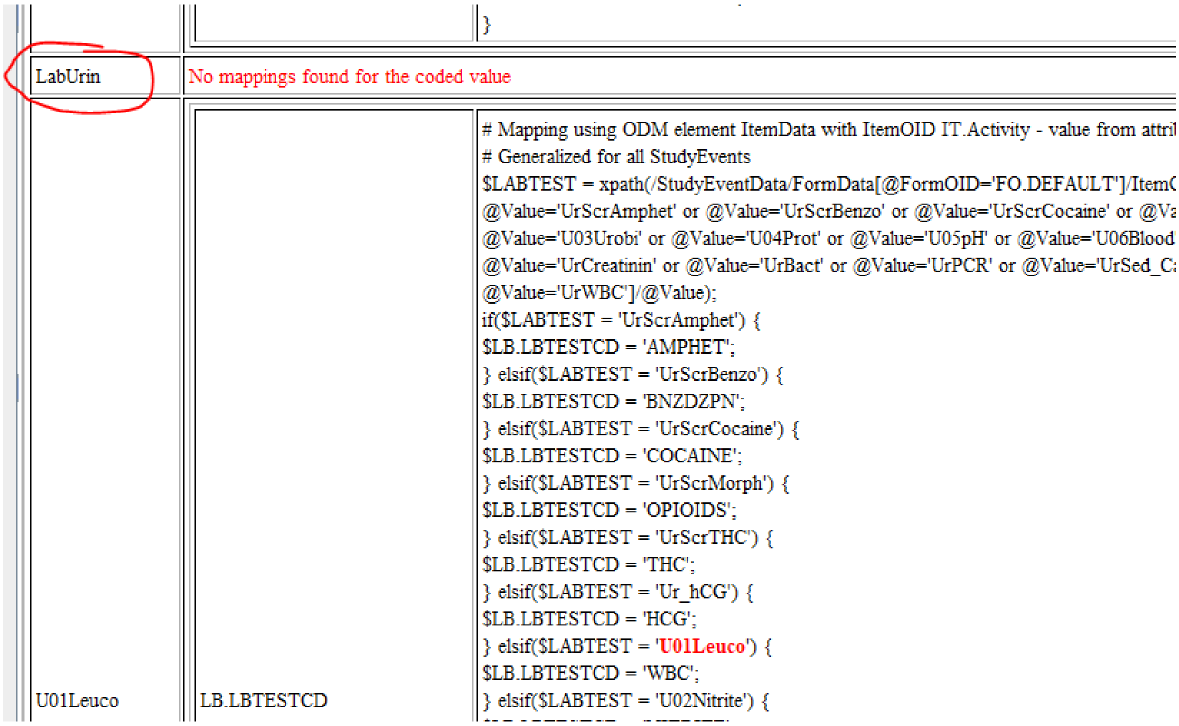
```
} elsif($VITALSTEST = 'HRsup') {
$VS.VSTESTCD = 'HR';
} elsif($VITALSTEST = 'Temperature') {
$VS.VSTESTCD = 'TEMP';
} else {
$VS.VSTESTCD = 'TODO';
}
```

Weight

VS.VSTESTCD

```
# Mapping using ODM element ItemData with ItemOID IT.Parameter - value from attribute ItemOID
# Generalized for all StudyEvents
$VITALSTEST = xpath(/StudyEventData/FormData[@FormOID='FO.DEFAULT']/ItemGroupData[@ItemGroupOID='IG.DEFAULT']/ItemData[@ItemOID='IT.Parameter'][@Value='Height' or @Value='Weight'
or @Value='BMI' or @Value='SystBPsup' or @Value='DiastBPsup' or @Value='HRsup' or @Value='Temperature']/@Value);
if($VITALSTEST = 'Height') {
$VS.VSTESTCD = 'HEIGHT';
} elsif($VITALSTEST = 'Weight') {
$VS.VSTESTCD = 'WEIGHT';
} elsif($VITALSTEST = 'BMI') {
$VS.VSTESTCD = 'BMI';
} elsif($VITALSTEST = 'SystBPsup') {
$VS.VSTESTCD = 'SYSBP';
} elsif($VITALSTEST = 'DiastBPsup') {
$VS.VSTESTCD = 'DIABP';
} elsif($VITALSTEST = 'HRsup') {
$VS.VSTESTCD = 'HR';
} elsif($VITALSTEST = 'Temperature') {
$VS.VSTESTCD = 'TEMP';
} else {
$VS.VSTESTCD = 'TODO';
}
```

showing that (and how) "Weight" from the source was mapped to SDTM.
In case no mapping was found (from the XPath expressions in the mapping scripts), this is also reported. For example:



```
}
```

LabUrin

No mappings found for the coded value

U01Leuco

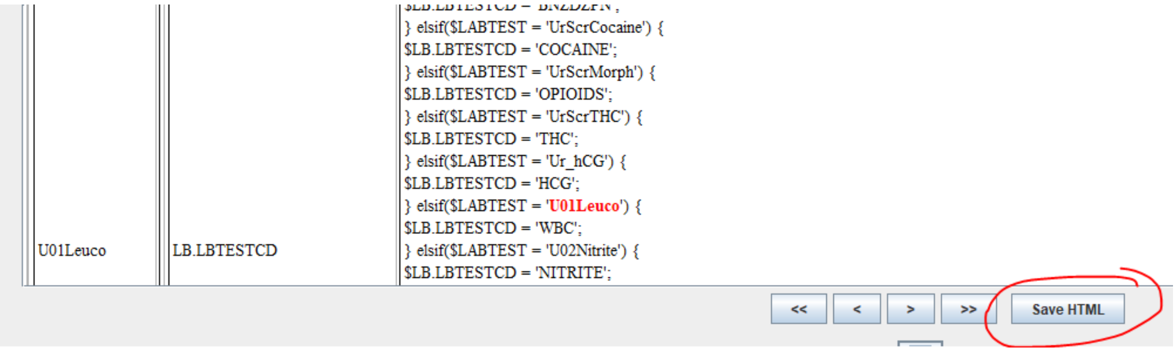
LB.LBTESTCD

```
# Mapping using ODM element ItemData with ItemOID IT.Activity - value from attri
# Generalized for all StudyEvents
$LABTEST = xpath(/StudyEventData/FormData[@FormOID='FO.DEFAULT']/ItemC
@Value='UrScrAmphet' or @Value='UrScrBenzo' or @Value='UrScrCocaine' or @Va
@Value='U03Urobi' or @Value='U04Prot' or @Value='U05pH' or @Value='U06Blood
@Value='UrCreatinin' or @Value='UrBact' or @Value='UrPCR' or @Value='UrSed_C
@Value='UrWBC']/@Value);
if($LABTEST = 'UrScrAmphet') {
$LB.LBTESTCD = 'AMPHET';
} elsif($LABTEST = 'UrScrBenzo') {
$LB.LBTESTCD = 'BNZDZPN';
} elsif($LABTEST = 'UrScrCocaine') {
$LB.LBTESTCD = 'COCAINE';
} elsif($LABTEST = 'UrScrMorph') {
$LB.LBTESTCD = 'OPIOIDS';
} elsif($LABTEST = 'UrScrTHC') {
$LB.LBTESTCD = 'THC';
} elsif($LABTEST = 'Ur_hCG') {
$LB.LBTESTCD = 'HCG';
} elsif($LABTEST = 'U01Leuco') {
$LB.LBTESTCD = 'WBC';
} elsif($LABTEST = 'U02Nitrite') {
```

IMPORTANT!
- "No mappings found" does not necessarily mean that the item was never used. It means that it was not used in a direct way in an XPath expression (i.e. in a query to the ODM).
When an item is reported with "No mappings found", the user should carefully check the mappings, and also check the SDTM results.
There may be good reasons why an item from the ODM was not used, such as e.g. "cleaning aids" for which a typical example is "Did any adverse events occur".
- When an item from the ODM is reported to be used in a mapping, this does not necessarily mean that a result for it will appear in the SDTM, or that the mapping is correct. For example, if one maps "Weight" to the "LB" (Laboratory) domain, this obviously is incorrect.
After all, SDTM is about "categorization" of data, which always is ... arbitrary[1].

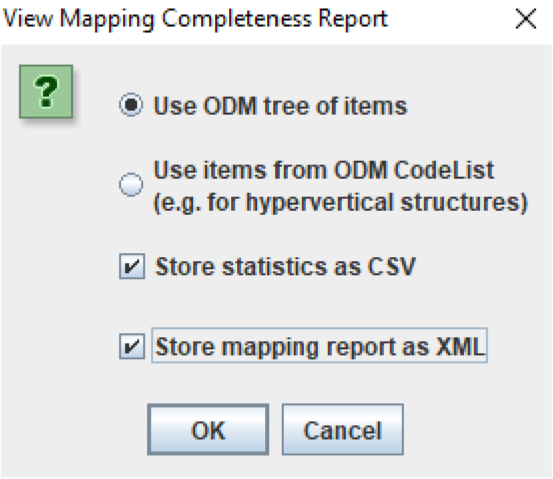
[1] An example is "Oxygen Saturation". If it is measured from blood retrieved from the subject, it will go into LB. When it is read from a pulse oxymeter, it goes into ... VS.

The generated "Mapping Completeness Report" can then be saved to file as HTML (e.g. for discussion with colleagues) by using the button "Save HTML":



U01Leuco	LB.LBTESTCD	$LB.LBTESTCD = 'BNZDZPN';
		} elsif($LABTEST = 'UrScrCocaine') {
		$LB.LBTESTCD = 'COCAINE';
		} elsif($LABTEST = 'UrScrMorph') {
		$LB.LBTESTCD = 'OPIOIDS';
		} elsif($LABTEST = 'UrScrTHC') {
		$LB.LBTESTCD = 'THC';
		} elsif($LABTEST = 'Ur_hCG') {
		$LB.LBTESTCD = 'HCG';
		} elsif($LABTEST = 'U01Leuco') {
		$LB.LBTESTCD = 'WBC';
		} elsif($LABTEST = 'U02Nitrite') {
		$LB.LBTESTCD = 'NITRITE';

Also new are the checkboxes "Store statistics as CSV" and "Store mapping report as XML" in the prior dialog:



View Mapping Completeness Report

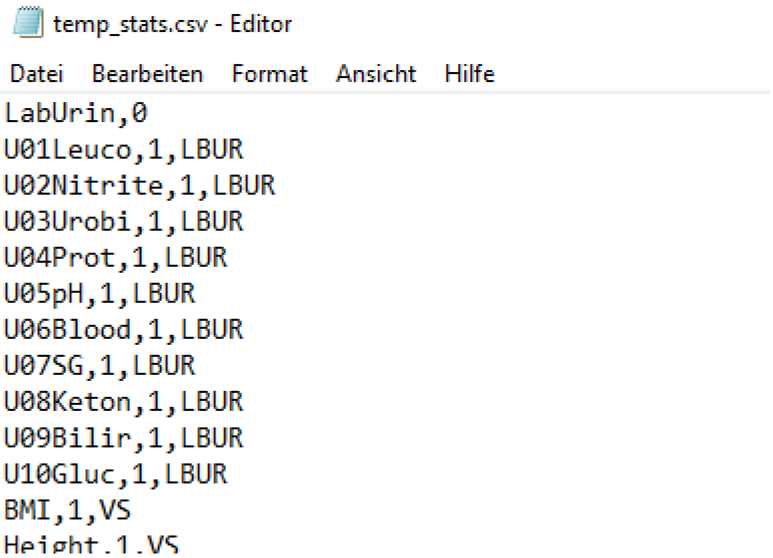
● Use ODM tree of items

○ Use items from ODM CodeList (e.g. for hypervertical structures)

☑ Store statistics as CSV

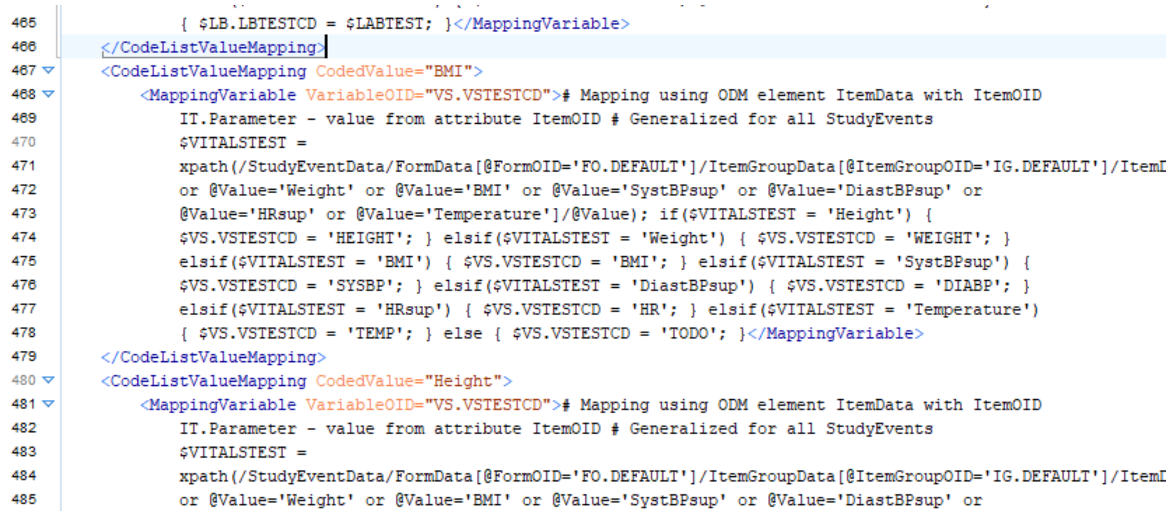
☑ Store mapping report as XML

The former will generate a CSV file that can e.g. be used in spreadsheets like Excel. It lists each of the items with the number of mappings found for it, and to which dataset/domain the item was mapped to. For example:



```
LabUrin,0
U01Leuco,1,LBUR
U02Nitrite,1,LBUR
U03Urobi,1,LBUR
U04Prot,1,LBUR
U05pH,1,LBUR
U06Blood,1,LBUR
U07SG,1,LBUR
U08Keton,1,LBUR
U09Bilir,1,LBUR
U10Gluc,1,LBUR
BMI,1,VS
Height,1,VS
```

The "Mapping Completeness Report" in XML format can be used in other (more modern applications). The format is shown below:

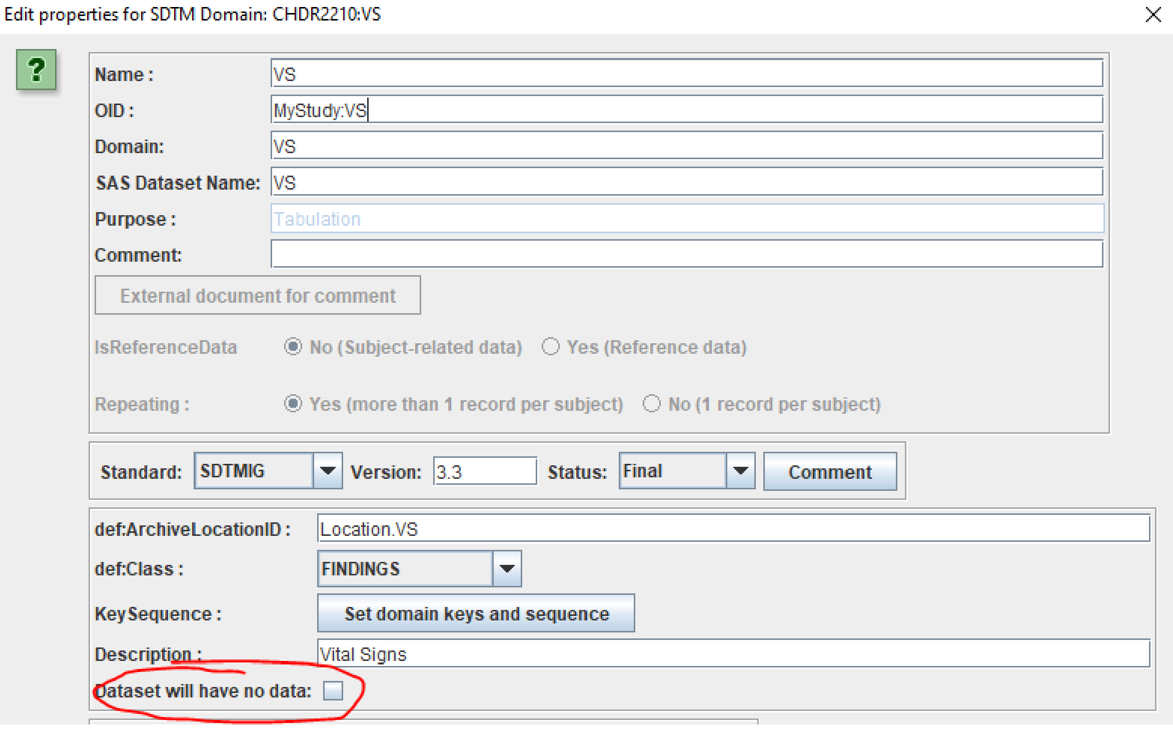


```
        { $LB.LBTESTCD = $LABTEST; }</MappingVariable>
</CodeListValueMapping>
<CodeListValueMapping CodedValue="BMI">
    <MappingVariable VariableOID="VS.VSTESTCD"># Mapping using ODM element ItemData with ItemOID
        IT.Parameter - value from attribute ItemOID # Generalized for all StudyEvents
        $VITALSTEST =
        xpath(/StudyEventData/FormData[@FormOID='FO.DEFAULT']/ItemGroupData[@ItemGroupOID='IG.DEFAULT']/ItemI
        or @Value='Weight' or @Value='BMI' or @Value='SystBPsup' or @Value='DiastBPsup' or
        @Value='HRsup' or @Value='Temperature']/@Value); if($VITALSTEST = 'Height') {
        $VS.VSTESTCD = 'HEIGHT'; } elsif($VITALSTEST = 'Weight') { $VS.VSTESTCD = 'WEIGHT'; }
        elsif($VITALSTEST = 'BMI') { $VS.VSTESTCD = 'BMI'; } elsif($VITALSTEST = 'SystBPsup') {
        $VS.VSTESTCD = 'SYSBP'; } elsif($VITALSTEST = 'DiastBPsup') { $VS.VSTESTCD = 'DIABP'; }
        elsif($VITALSTEST = 'HRsup') { $VS.VSTESTCD = 'HR'; } elsif($VITALSTEST = 'Temperature')
        { $VS.VSTESTCD = 'TEMP'; } else { $VS.VSTESTCD = 'TODO'; }</MappingVariable>
</CodeListValueMapping>
<CodeListValueMapping CodedValue="Height">
    <MappingVariable VariableOID="VS.VSTESTCD"># Mapping using ODM element ItemData with ItemOID
        IT.Parameter - value from attribute ItemOID # Generalized for all StudyEvents
        $VITALSTEST =
        xpath(/StudyEventData/FormData[@FormOID='FO.DEFAULT']/ItemGroupData[@ItemGroupOID='IG.DEFAULT']/ItemI
        or @Value='Weight' or @Value='BMI' or @Value='SystBPsup' or @Value='DiastBPsup' or
```

If received well by the customers, the "Mapping Completeness Report" in XML format will further be optimized and enhanced. We do however realize many companies still use Excel for such tasks.

Support for "HasNoData" in Define-XML 2.1

CDISC Define-XML 2.1 has a new feature for indicating that a dataset has no data, i.e. is empty. This property can now be set (at least when Define-XML version 2.1 is used) using the menu "Edit - SDTM Domain Properties" or "Edit - SEND Domain Properties" in the case of CDISC-SEND, or by a double-click of the first cell (containing the dataset information) in the SDTM table. When doing so, in the case of Define-XML 2.1, one will notice an additional checkbox in the dialog:



Edit properties for SDTM Domain: CHDR2210:VS

Name : VS

OID : MyStudy:VS

Domain: VS

SAS Dataset Name: VS

Purpose : Tabulation

Comment:

IsReferenceData ● No (Subject-related data) ○ Yes (Reference data)

Repeating : ● Yes (more than 1 record per subject) ○ No (1 record per subject)

Standard: SDTMIG Version: 3.3 Status: Final

def:ArchiveLocationID : Location.VS

def:Class : FINDINGS

KeySequence :

Description : Vital Signs

Dataset will have no data: □

When checked, the underlying "ItemGroupDef" for that dataset definition will have like:

<ItemGroupDef OID="MyStudy:VS" Name="VS" Domain="VS" **def:HasNoData="Yes"** ...

IMPORTANT: currently, setting "Dataset will have no data" to "true" will not prevent to add mappings (which will generate data in the dataset) for that dataset, nor to delete the existing mappings for that dataset. It remains the responsibility of the user to, when the checkbox was checked, to take care that the dataset really has no data.

Multiple instances of the same domain

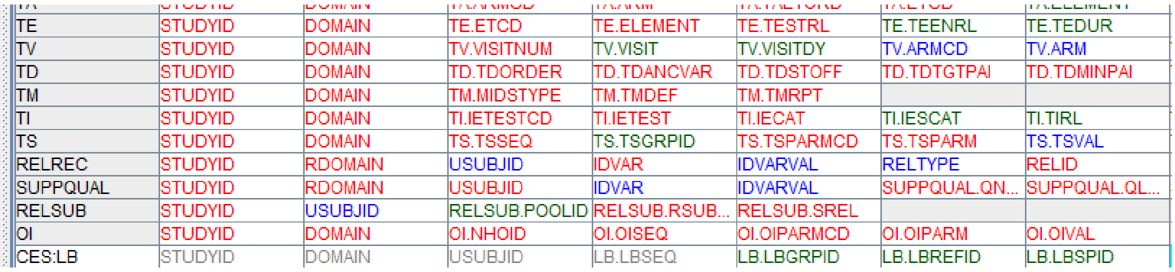
In some cases, users will want to have multiple instances of the same (SDTM) domain. This is already required for QS (Questionnaires) when there is more than one questionnaire used in the study. In such cases, the requirement is that there is one individual instance of QS per single questionnaire. For example, in the "SDTM Metadata Submission Guidelines v.2.0", one will find a QSPH (Patient Health Questionnaire-9) dataset and a QSSL dataset (Satisfaction with Life Survey). CDISC often speaks about "splitted datasets", but this designation is essentially wrong, as no splitting of whatsoever was done: one just generates two or more datasets based on the same SDTM domain.
Having different instances of the same domain can however also have advantages for other domains than QS (where it is required), for example, to keep file size at an acceptable level. We also often encourage customers to develop different instances for e.g. LB (Laboratory), as that not only makes mapping considerably easier, but also makes review of the datasets much more "reviewer-friendly", the reviewer not having to filter or sort a dataset with millions of rows. For LB, one can e.g. have different instances based on the category (LBCAT), like "urinalysis", "hematology", "chemistry". This would e.g. mean that one then generates the datasets such as LBUR, LBHM, LBCH, ... Remark that dataset names (except for SUPPxx

datasets) are limited to 4 characters by the SDTMIG[2].

When having multiple instances of the same domain, one will sometimes want to keep the properties of the different instances identical, and sometimes not.
For example, for QS, it doesn't make sense to align the properties (variable lengths, associated codelists, etc.) aligned between the different instances.
In the case of LB for example, one may want to merge (maybe on request of the regulatory authority) the different instances into a single (probably huge) LB dataset later. This may become complicated when the properties of the different instances differ, for example if the number of variables between the different instances differ, like when one there are timepoints (LBTPT, LBTPTNUM, ...) defined for one category of lab data, but none for another category of lab data. Also the "length" for the same variable can differ between different instances of the same domain, making merging a bit more complicated in the case of having to use the outdated SAS Transport 5 format[3]. If only the "lengths" differ between variables, SDTM-ETL 4.3 will however take the largest value for the "length" when also generating a "super", merged dataset (and the corresponding SUPPxx dataset) in XPT format, when the checkbox "Additionally generate a merged dataset for 'split' domain datasets" was checked (see next section)

New features have been added to SDTM-ETL v.4.3 to allow the user to align, or to omit aligning properties between different instances of the same domain.

Assume that we already have set up a study-specific LB dataset.

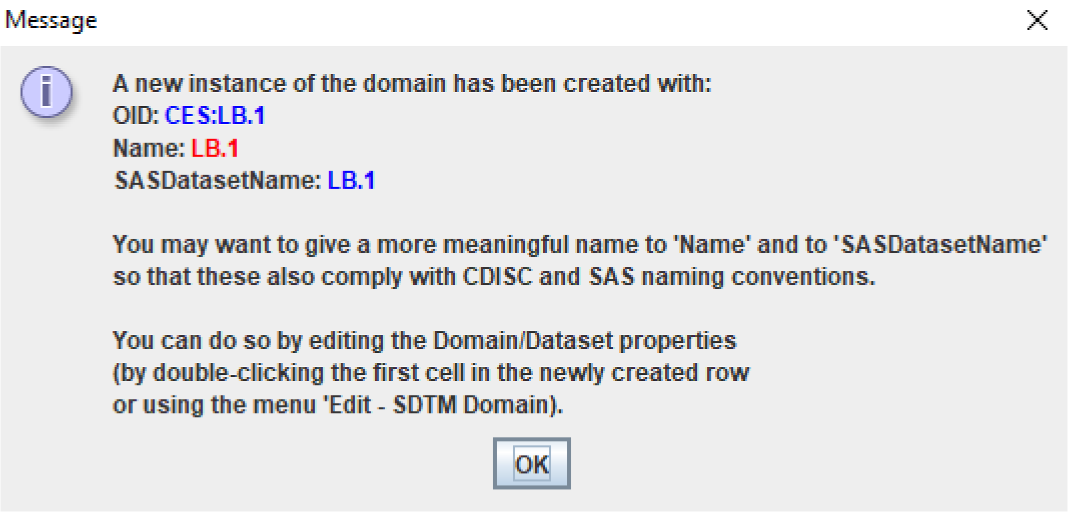


TA	STUDYID	DOMAIN	TA.ARMCD	TA.ARM	TA.TAETORD	TA.ETCD	TA.ELEMENT
TE	STUDYID	DOMAIN	TE.ETCD	TE.ELEMENT	TE.TESTRL	TE.TEENRL	TE.TEDUR
TV	STUDYID	DOMAIN	TV.VISITNUM	TV.VISIT	TV.VISITDY	TV.ARMCD	TV.ARM
TD	STUDYID	DOMAIN	TD.TDORDER	TD.TDANCVAR	TD.TDSTOFF	TD.TDTGTPAI	TD.TDMINPAI
TM	STUDYID	DOMAIN	TM.MIDSTYPE	TM.TMDEF	TM.TMRPT		
TI	STUDYID	DOMAIN	TI.IETESTCD	TI.IETEST	TI.IECAT	TI.IESCAT	TI.TIRL
TS	STUDYID	DOMAIN	TS.TSSEQ	TS.TSGRPID	TS.TSPARMCD	TS.TSPARM	TS.TSVAL
RELREC	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	RELTYPE	RELID
SUPPQUAL	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	SUPPQUAL.QN...	SUPPQUAL.QL...
RELSUB	STUDYID	USUBJID	RELSUB.POOLID	RELSUB.RSUB...	RELSUB.SREL		
OI	STUDYID	DOMAIN	OI.NHOID	OI.OISEQ	OI.OIPARMCD	OI.OIPARM	OI.OIVAL
CES:LB	STUDYID	DOMAIN	USUBJID	LB.LBSEQ	LB.LBGRPID	LB.LBREFID	LB.LBSPID

and we now want to add an additional one.
This can be done by either drag-and-drop from the template, or drag-and-drop CES:LB itself.
The result will then be a dialog showing up:

[2] This is once again due to the use of the outdated SAS Transport 5 format (see the TS-140 specification). It allows for up to 8 characters for dataset names. Very probably the 4-character rule is meant to avoid that the corresponding "supplemental qualifier" dataset get a name that has more than 8 characters. For example, when allowing LBCHEM, the corresponding "supplemental qualifier" dataset would have to be named SUPPLBCHEM (by "naming convention"), which exceeds 8 characters. Modern IT looks different ...

[3] Merging would be a "no brainer" in the case we would be allowed to use modern transport formats like CDISC Dataset-JSON or Dataset-XML.

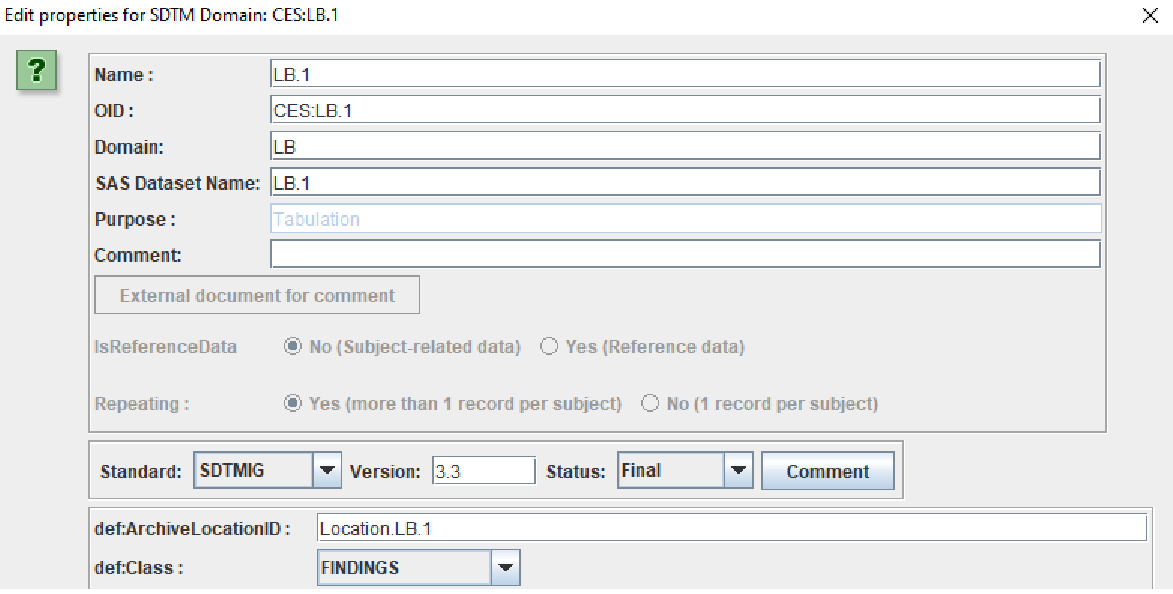

Message

A new instance of the domain has been created with:
OID: CES:LB.1
Name: LB.1
SASDatasetName: LB.1

You may want to give a more meaningful name to 'Name' and to 'SASDatasetName' so that these also comply with CDISC and SAS naming conventions.

You can do so by editing the Domain/Dataset properties
(by double-clicking the first cell in the newly created row
or using the menu 'Edit - SDTM Domain).

recommending to rename the dataset to something more usual.
After clicking "OK", we can then change the properties of "LB.1" by double click of the first cell, or by selecting the "LB.1" row and then using the menu "Edit - SDTM Domain Properties". This leads to:


Edit properties for SDTM Domain: CES:LB.1

Name : LB.1
OID : CES:LB.1
Domain: LB
SAS Dataset Name: LB.1
Purpose : Tabulation
Comment:

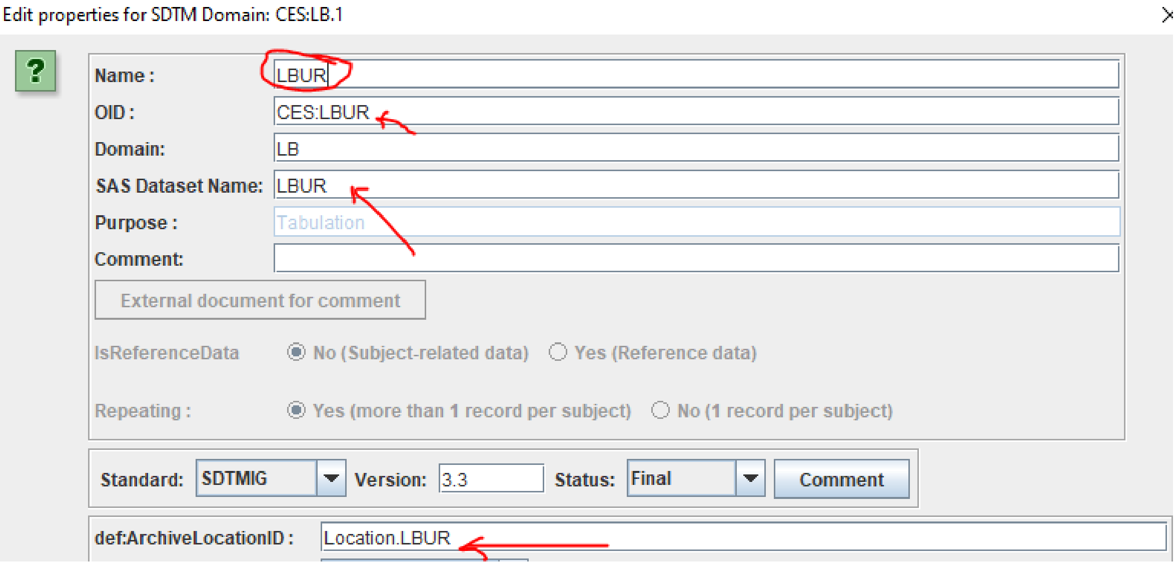
IsReferenceData ● No (Subject-related data) ○ Yes (Reference data)

Repeating : ● Yes (more than 1 record per subject) ○ No (1 record per subject)

Standard: SDTMIG Version: 3.3 Status: Final

def:ArchiveLocationID : Location.LB.1
def:Class : FINDINGS

New in v.4.3 of SDTM-ETL is that one edits the "Name", the values of "OID", "SAS Datatset Name" and "def:ArchiveLocation" will automatically be synchronized, meaning that when one types in the "Name" field, the values in the latter fields will automatically be adapted. For example:



Edit properties for SDTM Domain: CES:LB.1

Name : LBUR

OID : CES:LBUR

Domain: LB

SAS Dataset Name: LBUR

Purpose : Tabulation

Comment:

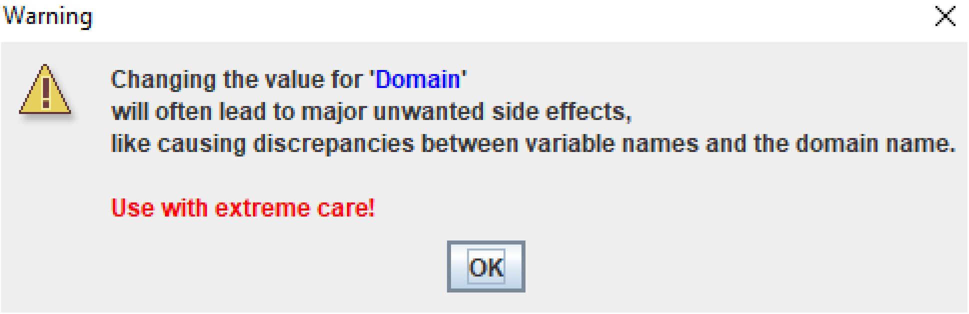
IsReferenceData ● No (Subject-related data) ○ Yes (Reference data)

Repeating : ● Yes (more than 1 record per subject) ○ No (1 record per subject)

Standard: SDTMIG Version: 3.3 Status: Final

def:ArchiveLocationID : Location.LBUR

One can then still individually edit the field "OID", "SAS Dataset Name" and "def:ArchiveLocationID", but this usually will not make much sense.
Remark that the field "Domain" remains unchanged, and when one tries to edit it, one will get the warning:

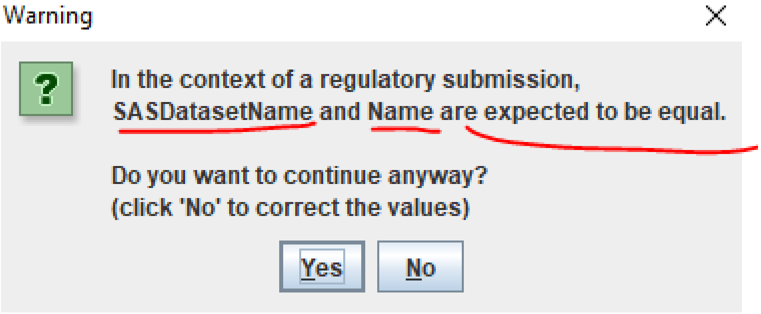


Warning

Changing the value for 'Domain'
will often lead to major unwanted side effects,
like causing discrepancies between variable names and the domain name.

Use with extreme care!

Also, when trying to edit the "SAS Dataset Name" (another relict of SAS Transport 5), a warning will be shown when clicking the "OK" button:



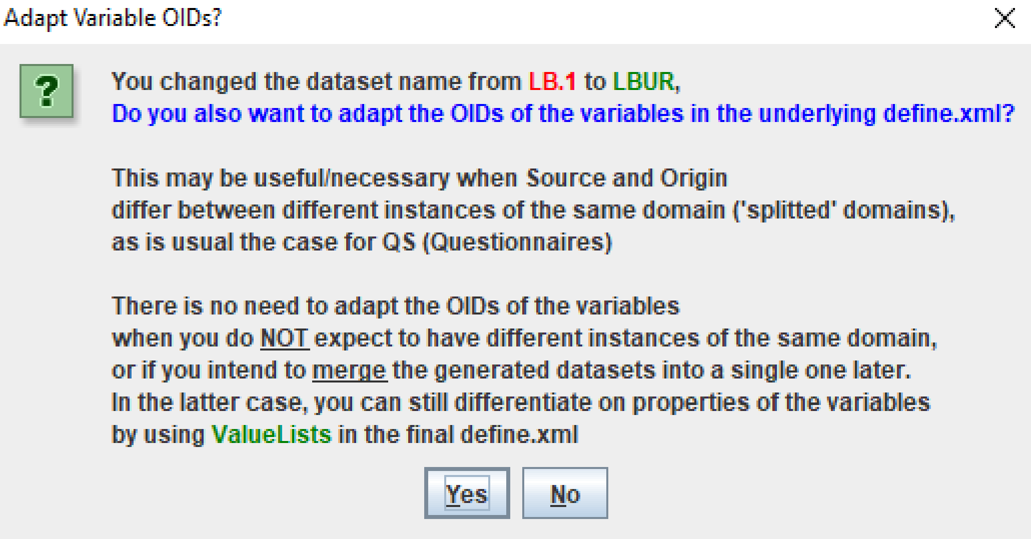
Warning

In the context of a regulatory submission,
SASDatasetName and Name are expected to be equal.

Do you want to continue anyway?
(click 'No' to correct the values)

with the opportunity to still adapt the value by clicking "No".

When then finally "saving" the changed properties (one can also adapt the dataset "label" somewhat to make it clearer what the dataset is about), a new dialog is shown:



Adapt Variable OIDs?

You changed the dataset name from LB.1 to LBUR,
Do you also want to adapt the OIDs of the variables in the underlying define.xml?

This may be useful/necessary when Source and Origin
differ between different instances of the same domain ('splitted' domains),
as is usual the case for QS (Questionnaires)

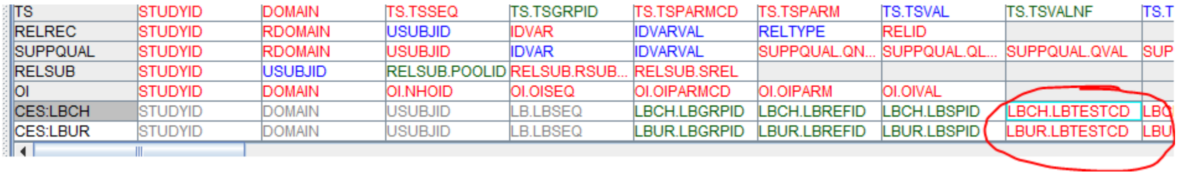
There is no need to adapt the OIDs of the variables
when you do NOT expect to have different instances of the same domain,
or if you intend to merge the generated datasets into a single one later.
In the latter case, you can still differentiate on properties of the variables
by using ValueLists in the final define.xml

If one selects "Yes", meaning that also the OIDs of the variables will be adapted (e.g. the OID for LBTESTCD, which is "LB.LBTESTCD" will then be changed into "LBUR.LBTESTCD". This then means that the properties for LBTESTCD, such as length, associated codelist, ...) will be different between the different instances of the LB domain.

When selecting "No" however, the OIDs of the variables of all instances of the LB domain will be identical, i.e. one aligns the properties of the variables between the different instances. This can make sense when one later wants to merge all the LB instances into a single LB (which may cause a huge LB dataset however). It however also means that in the final define.xml, one will want or need to define a lot of ValueLists to differentiate between the different categories of data.

Having dataset and variable properties differing between different instances of the same domain always makes sense in the case of QS (as reviewers will not want to merge these anyway), and depends on what wants to do in the case of e.g. LB.

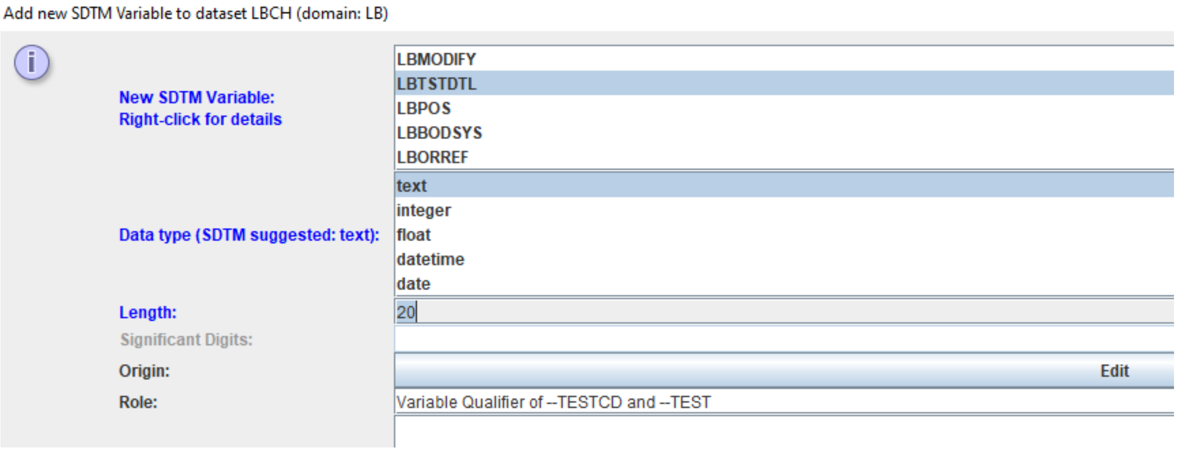
In our example, one can then still also rename the original "LB" dataset, into e.g. LBHE (for "hematology", e.g. leading to:



TS	STUDYID	DOMAIN	TS.TSSEQ	TS.TSGRPID	TS.TSPARMCD	TS.TSPARM	TS.TSVAL	TS.TSVALNF	TS.T
RELREC	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	RELTYPE	RELID		
SUPPQUAL	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	SUPPQUAL.QN...	SUPPQUAL.QL...	SUPPQUAL.QVAL	SUP
RELSUB	STUDYID	USUBJID	RELSUB.POOLID	RELSUB.RSUB...	RELSUB.SREL				
OI	STUDYID	DOMAIN	OI.NHOID	OI.OISEQ	OI.OIPARMCD	OI.OIPARM	OI.OIVAL		
CES:LBCH	STUDYID	DOMAIN	USUBJID	LB.LBSEQ	LBCH.LBGRPID	LBCH.LBREFID	LBCH.LBSPID	LBCH.LBTESTCD	LBC
CES:LBUR	STUDYID	DOMAIN	USUBJID	LB.LBSEQ	LBUR.LBGRPID	LBUR.LBREFID	LBUR.LBSPID	LBUR.LBTESTCD	LBU

Notice that for example, for LBTESTCD, the OIDs have been changed into "LBCH.LBTESTCD" and "LBUR.LBTESTCD" respectively, meaning that although the variable name is the same (it still is "LBTESTCD"), the properties can differ, depending on whether the variable is used in LBCH (Chemistry) or LBUR (Urinalysis). For example, the contents of the associated codelist for LBTESTCD can then differ for "Chemistry" and for "Urinalysis" (which completely makes sense). Both codelists will probably be a subset of the CDISC "LBTESTCD" codelist (NCI-code C65047), maybe extended with codes for which there is no CDISC code.

Similarly, when one wants to add an additional "standard" variable to one of the instances, using the menu "Insert - New SDTM Variable", e.g. LBTSTDTL ("Test Detail") for LBCH:



Add new SDTM Variable to dataset LBCH (domain: LB)

New SDTM Variable:
Right-click for details

LBMODIFY
LBTSTDTL
LBPOS
LBBODSYS
LBORREF

Data type (SDTM suggested: text):

text
integer
float
datetime
date

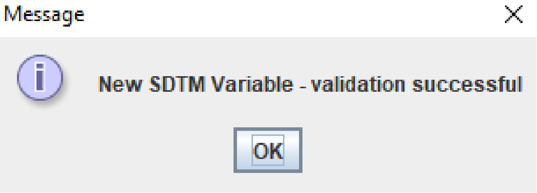
Length: 20

Significant Digits:

Origin:

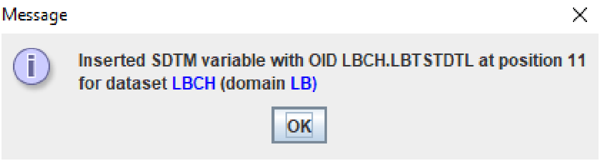
Role: Variable Qualifier of --TESTCD and --TEST

and, after filling the necessary field (like maximal length), and clicking OK, the classic dialogs will be showed:



Message

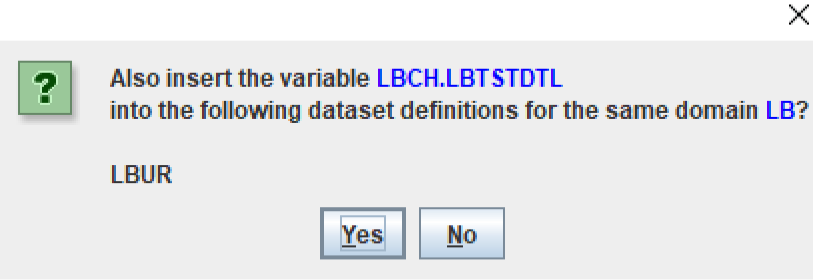
New SDTM Variable - validation successful



Message

Inserted SDTM variable with OID LBCH.LBTSTDTL at position 11 for dataset LBCH (domain LB)

followed by a new dialog:

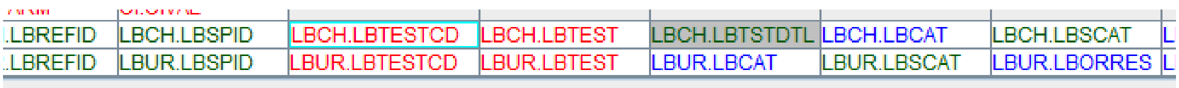


Also insert the variable LBCH.LBTSTDTL into the following dataset definitions for the same domain LB?

LBUR

asking whether LBTSTDTL also needs to be added to the "other" instance of LB (i.e. LBUR) or not. If one clicks "Yes", both instances of LB will then have an LBTSTDTL variable (although with possible different properties). When clicking "No", LBTSTDTL is only inserted into the LBCH ("Chemistry") instance, and not into LBUR ("Urinalysis").
This is just fine, and probably very useful when LBTSTDTL does not make sense in LBUR (only unnecessary blowing up the XPT file size with a column that has no data), but it may make things complicated when later wanting to merge LBCH and LBUR into a single dataset.

If "No" is clicked, we will find:



.LBREFID	LBCH.LBSPID	LBCH.LBTESTCD	LBCH.LBTEST	LBCH.LBTSTDTL	LBCH.LBCAT	LBCH.LBSCAT	L
.LBREFID	LBUR.LBSPID	LBUR.LBTESTCD	LBUR.LBTEST	LBUR.LBCAT	LBUR.LBSCAT	LBUR.LBORRES	L

i.e. variable LBTSTDTL has only been added to the LBCH instance.

This, and other features for working with different instances of the same domain, is explained in a separate tutorial "Best practices for Questionnaires and other 'split' domains".

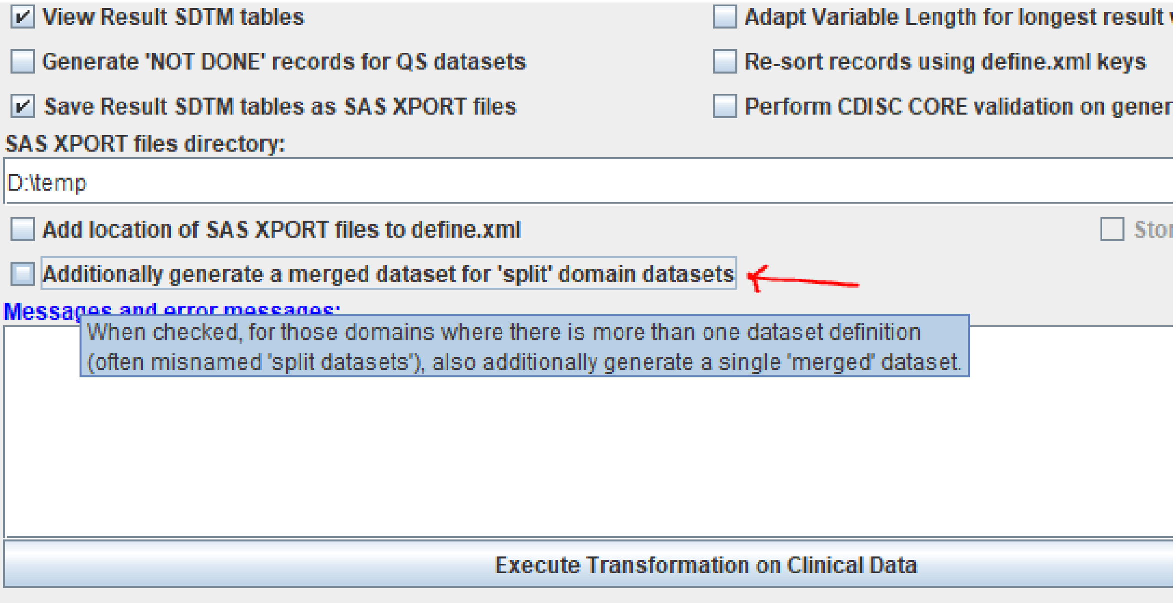
Merging different dataset instances for the same domain when needing to use SAS-XPT format

The SAS Transport (5 or 8) format ("XPT") is not only completely outdated (it was developed to allow exchange between IBM mainframes and VAX workstations - do you still have one of these?), it is also extremely inefficient. Essentially, the records in XPT files are nothing else than sort of "punch cards" each having the same length.
The reason is that for each variable in the dataset, the "length" has to be set to at least the maximum length of an individual value for that length (with a maximum of 200). Suppose for example, that for a specific findings dataset, all the --ORRES variables have either the value "Y" or "N", but there is only one --ORRES value "the quick brown fox jumps over the lazy dog" (43 characters), than all "Y" and "N" fields in the file are padded with 42 blanks. Otherwise said, for the --ORRES column only, the data file will consist of over 95% of blanks. This also means that when using a modern format like Dataset-JSON, the file size could be about 50 times smaller.
This extreme inefficiency is also the reason FDA states "*the allotted length for each column containing character (text) data should be set to the maximum length of the variable used in the individual dataset*" thus reducing further "damage"[4].

Very often, it makes sense to have different instances of the same domain, especially when there is a lot of data for it. Usually, this applies to the LB domain. Unless there is only a small amount of lab data, generating different instances of LB, usually based on the "category" (which is then set in LBCAT) completely makes sense. This not only makes the mapping considerably more easy, but also makes review of the lab data by the regulatory authorities considerably easier ("reviewer-friendly"). For example, reviewing a dataset with several millions of records for different categories of lab data is extremely tedious.

SDTM-ETL encourages developing different instances of the same domain by nature. If done in a clever way, e.g. by basing the different instances on the different categories of the (e.g. lab) data, "reviewer-friendly" SDTM datasets are produced.
There can be situations however, that a reviewer still wants to obtain a "merged" or "overall" single dataset, although this usually doesn't make sense.
As of version 4.2, SDTM-ETL enables to additionally generate a "merged" dataset when it detects that there are different instances of the same domain to be generated. This feature can be switched on by checking the checkbox "Additionally generate a merged dataset for 'split' domain datasets":

[4] It is a bit strange that FDA first mandates an extremely inefficient format, which by nature, leads to very large file sizes, and then forces sponsors to "optimize" the files for variable length. This is like mandating people to drive an extremely polluting car (e.g. Diesel driven), and then say they should drive slower to reduce the pollution.


☑ View Result SDTM tables

☐ Adapt Variable Length for longest result

☐ Generate 'NOT DONE' records for QS datasets

☐ Re-sort records using define.xml keys

☑ Save Result SDTM tables as SAS XPORT files

☐ Perform CDISC CORE validation on gener

SAS XPORT files directory:

D:\temp

☐ Add location of SAS XPORT files to define.xml

☐ Stor

☐ Additionally generate a merged dataset for 'split' domain datasets

Messages and error messages:

When checked, for those domains where there is more than one dataset definition (often misnamed 'split datasets'), also additionally generate a single 'merged' dataset.

The checkbox will however only appear in the case of SAS-XPT as the output format. We will later also introduce it for Dataset-JSON, depending on the outcome of the Dataset-JSON-FDA pilot.
In v.4.2, "merging" different instances of the same domain only worked when all variables within the domain, over the different instances had the same properties (i.e. shared the same Define-XML "ItemDefs"). As of v.4.3, it is however possible to have different properties for the same SDTM variable, depending on the domain instance (see before). For example, in one instance, the "length" of the --ORRES variable may have been set to 10, whereas it is set to 20 in another instance. For merging with SAS-XPT, this may lead to problems, as XPT is a "fixed-field-length" format. Therefore, the algorithm for merging has been altered.
When merging different instances of the same domain, the software will now first look up what the maximum length set for each variable is in each of the instances, and use that length for the "merged" dataset. In our example, this would mean that it is found that for the different instances of the --ORRES variable, it is found that the maximum length set is 20, and that length will be used for the --ORRES field in the merged dataset.
This of course increases the inefficiency in the merged dataset (increasing the number of "blank" bytes), but that is inherent to SAS Transport - so blame the FDA ...

Furthermore, in version 4.3, the generation of "merged" datasets has been extended to also generate a "merged" dataset when also different instances of SUPPxx are being generated. For example, when we have a LBUR (urinalysis) and a LBCH (blood chemistry) instance of LB, and these have one or more non-standard variables that will then be "banned" into SUPPLBUR and SUPPLBCH, and the checkbox "Additionally generate a merged dataset for 'split' domain datasets" is checked, also a single SUPPLB SAS-XPT dataset will be generated, combining the records from SUPPLBUR and SUPPLBCH.

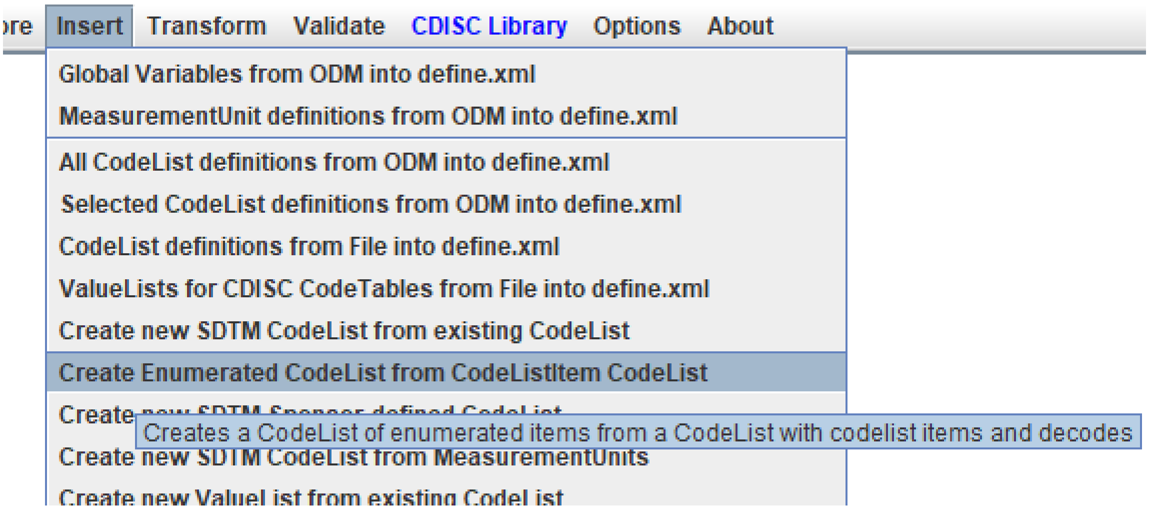
New is also that the distribution comes with a separate software, named "XML2SASDatasetMerger", allowing to merge different datasets for the same domain, into a single SAS-XPT dataset.

All these features are explained in the tutorial "Merging Datasets", which can be found at the SDTM-ETL website.

Generate "Enumerated" CodeList from "CodeListItem" CodeList

On request of one of our customers, we also added the feature to generate a codelist containing only "enumerated" items, i.e. a simple list of items, starting from a codelist that has as well coded values as "decoded", i.e. containing the meaning of the code in one or more languages.

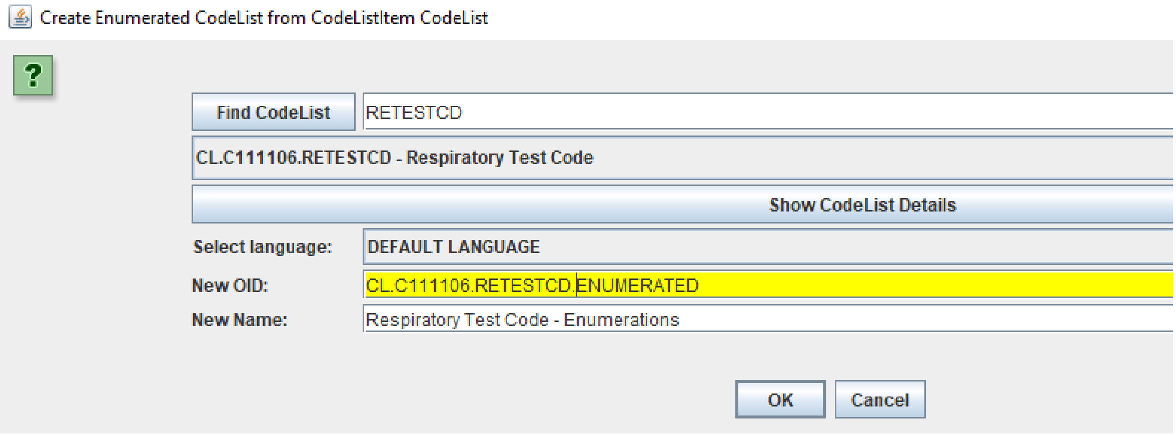
When using the feature using the menu "Edit - Create Enumerated CodeList from CodeListItem CodeList"


ore
Insert
Transform
Validate
CDISC Library
Options
About
Global Variables from ODM into define.xml
MeasurementUnit definitions from ODM into define.xml
All CodeList definitions from ODM into define.xml
Selected CodeList definitions from ODM into define.xml
CodeList definitions from File into define.xml
ValueLists for CDISC CodeTables from File into define.xml
Create new SDTM CodeList from existing CodeList
Create Enumerated CodeList from CodeListItem CodeList
Create new SDTM Sponsor-defined CodeList
Create new SDTM CodeList from MeasurementUnits
Create new ValueList from existing CodeList

Creates a CodeList of enumerated items from a CodeList with codelist items and decodes

only the "decoded" values for the selected language (which can be the "default" language") is retained, including the NCI codes (when present).

By default, the new codelist has the OID of the source codelist, extended with ".ENUMERATED", and also the codelist name is extended with "- Enumerations". For example, when using this feature on the codelist "CL.C111106.RETESTCD" with name "Respiratory Test Code", the OID of the new codelist will be "CL.C111106.RETESTCD.ENUMERATED" and the new name will be "Respiratory Test Code - Enumerations":



RETESTCD

CL.C111106.RETESTCD - Respiratory Test Code

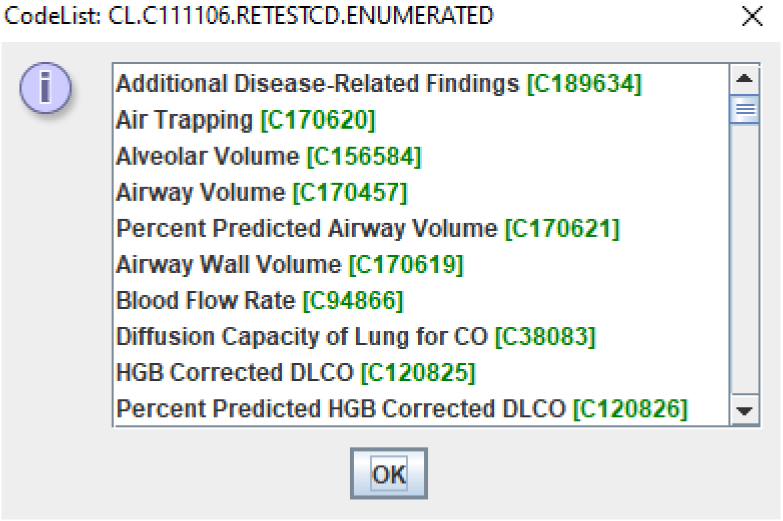
Select language: DEFAULT LANGUAGE

New OID: CL.C111106.RETESTCD.ENUMERATED

New Name: Respiratory Test Code - Enumerations

Of course one can then still change the values for the new OID and new name.

The newly created codelist then contains:



CodeList: CL.C111106.RETESTCD.ENUMERATED

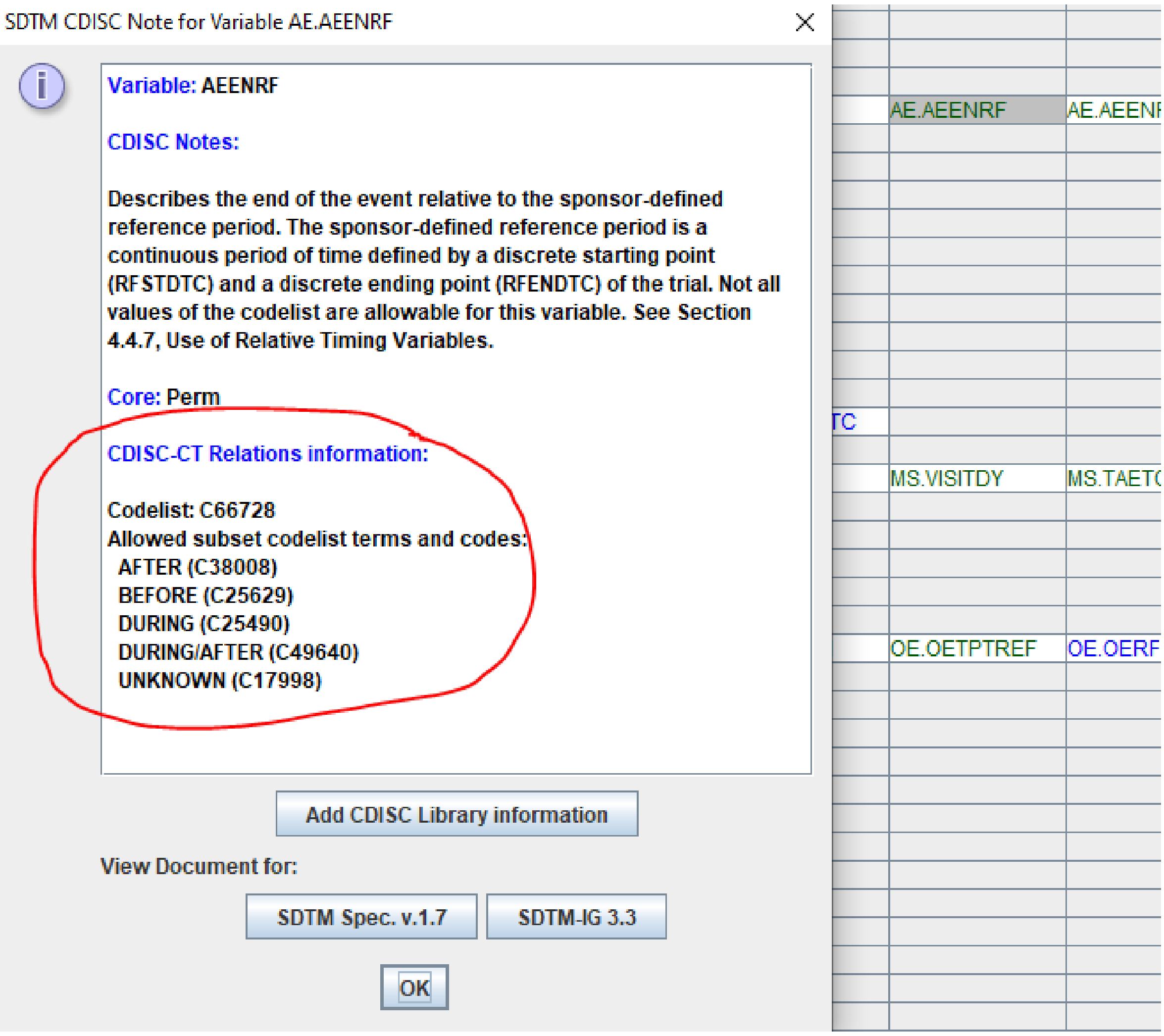
Additional Disease-Related Findings [C189634]
Air Trapping [C170620]
Alveolar Volume [C156584]
Airway Volume [C170457]
Percent Predicted Airway Volume [C170621]
Airway Wall Volume [C170619]
Blood Flow Rate [C94866]
Diffusion Capacity of Lung for CO [C38083]
HGB Corrected DLCO [C120825]
Percent Predicted HGB Corrected DLCO [C120826]

This feature can e.g. be used when one has a sponsor-defined codelist for tests (--TESTCD codelist) with the meaning of the test codes included, or when one has extended an existing CDISC codelist considerably, and also wants to generate the corresponding codelist for the --TEST variable.

Implementation of "CT Relationships"

Unfortunately, the CDISC-CT team has always refused to publish subsets of existing codelists for specific use cases. For example, all SDTM flag variables (ending with "FL" in the variable name) must, according to the SDTM-IGs, obey the C66742 "NY" codelist, having 4 allowed values, but at the same time, only "Y" is allowed in the case of -FL variables according to the SDTM-IG. So it would make sense to also have a "Yes-only" subset codelist. The CT team however still refuses to publish such a codelist. The only way to find out whether such a subset codelist is applicable for a specific SDTM variable, is to look into the PDF SDTM-IG, inspecting the "CDISC Notes" or "Assumptions". There was no electronic form of this information.
This is where the "CT Relationships" come into play. It, among other things, provides information about which subset codelists are applicable for which variables under which conditions. The newest version of it has been published as part of the "CDISC Wiki", and is available in electronic form in Excel, JSON, and YAML. We decided to use the JSON implementation.

When a template or study-specific define.xml is loaded, the JSON file is read and internally stored. When then asking for the "CDISC Notes" for a specific SDTM variable (for SDTM-IG 3.3 or 3.4), using either CTRL-H or the menu "View - SDTM CDISC Notes", the information from the "CT Relationships" is looked up, and added to the information dialog. For example, for AEENRF (End Relative to Reference Period):


SDTM CDISC Note for Variable AE.AEENRF

Variable: AEENRF

CDISC Notes:

Describes the end of the event relative to the sponsor-defined reference period. The sponsor-defined reference period is a continuous period of time defined by a discrete starting point (RFSTDTC) and a discrete ending point (RFENDTC) of the trial. Not all values of the codelist are allowable for this variable. See Section 4.4.7, Use of Relative Timing Variables.

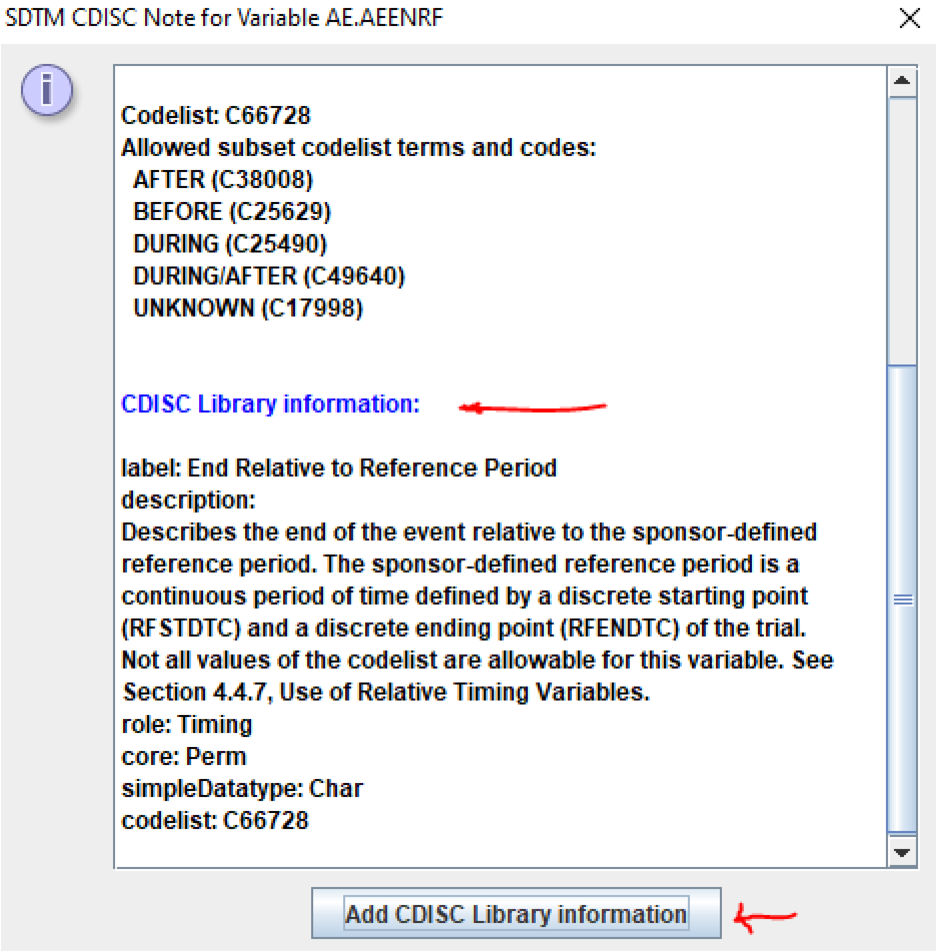
Core: Perm

CDISC-CT Relations information:

Codelist: C66728
Allowed subset codelist terms and codes:
AFTER (C38008)
BEFORE (C25629)
DURING (C25490)
DURING/AFTER (C49640)
UNKNOWN (C17998)

showing that only 5 values of codelist C66728 (otherwise containing 8 values) are allowed.

Also remind that when one has a CDISC Library account and API key (and has added it into the "properties.dat" file, one can also have the CDISC Library information added, by clicking the "Add CDISC Library information" button. For example:



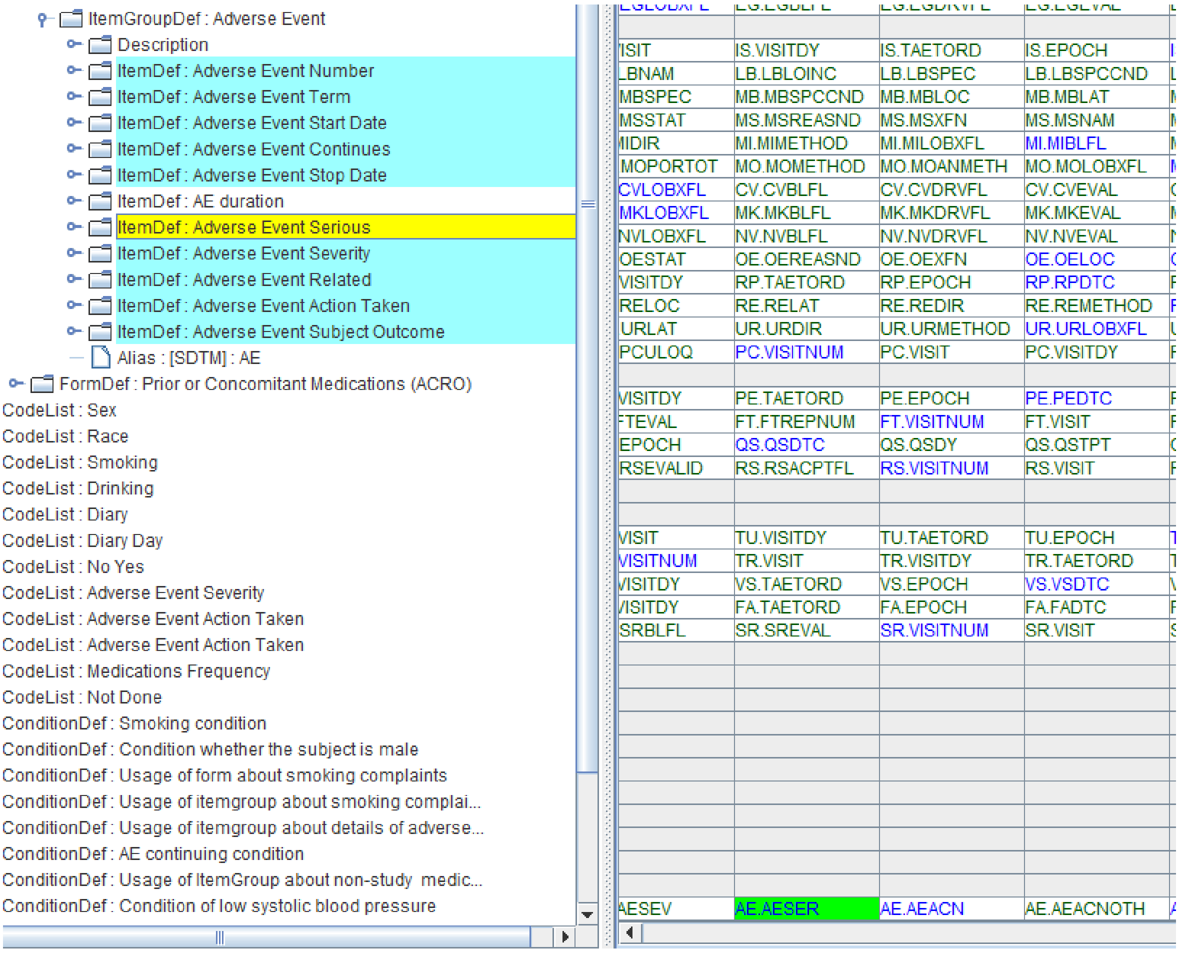
SDTM CDISC Note for Variable AE.AEENRF

Codelist: C66728
Allowed subset codelist terms and codes:
  AFTER (C38008)
  BEFORE (C25629)
  DURING (C25490)
  DURING/AFTER (C49640)
  UNKNOWN (C17998)

CDISC Library information:

label: End Relative to Reference Period
description:
Describes the end of the event relative to the sponsor-defined reference period. The sponsor-defined reference period is a continuous period of time defined by a discrete starting point (RFSTDTC) and a discrete ending point (RFENDTC) of the trial. Not all values of the codelist are allowable for this variable. See Section 4.4.7, Use of Relative Timing Variables.
role: Timing
core: Perm
simpleDatatype: Char
codelist: C66728

When now trying to map such a variable for which the "CT Relationships" apply, either by drag-and-drop, or by direct editing the mapping script for the variable, the system will compare the currently assigned codelist with the one from the "CT Relationships", and if these differ, ask the user to replace the currently assigned codelist by the one provided by the "CT Relationships" (which usually is a subset of the current codelist). For example for AESER when doing drag-and-drop from the ODM item "Adverse Event Serious" to the cell "AESER":

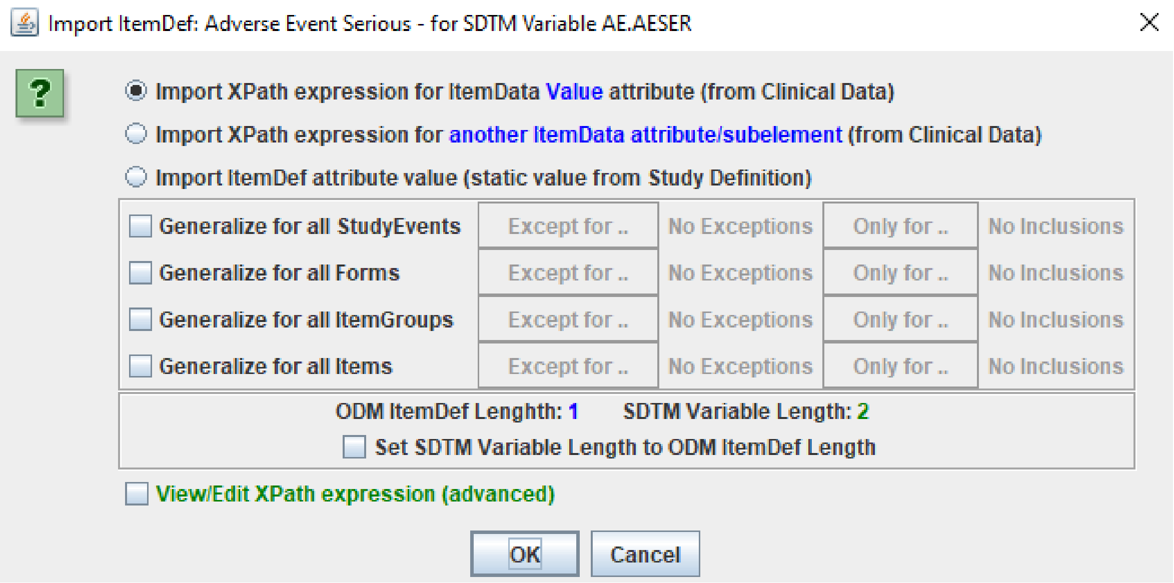


- ItemGroupDef : Adverse Event
  - Description
  - ItemDef : Adverse Event Number
  - ItemDef : Adverse Event Term
  - ItemDef : Adverse Event Start Date
  - ItemDef : Adverse Event Continues
  - ItemDef : Adverse Event Stop Date
  - ItemDef : AE duration
  - ItemDef : Adverse Event Serious
  - ItemDef : Adverse Event Severity
  - ItemDef : Adverse Event Related
  - ItemDef : Adverse Event Action Taken
  - ItemDef : Adverse Event Subject Outcome
  - Alias : [SDTM] : AE
- FormDef : Prior or Concomitant Medications (ACRO)

CodeList : Sex
CodeList : Race
CodeList : Smoking
CodeList : Drinking
CodeList : Diary
CodeList : Diary Day
CodeList : No Yes
CodeList : Adverse Event Severity
CodeList : Adverse Event Action Taken
CodeList : Adverse Event Action Taken
CodeList : Medications Frequency
CodeList : Not Done
ConditionDef : Smoking condition
ConditionDef : Condition whether the subject is male
ConditionDef : Usage of form about smoking complaints
ConditionDef : Usage of itemgroup about smoking complai...
ConditionDef : Usage of itemgroup about details of adverse...
ConditionDef : AE continuing condition
ConditionDef : Usage of ItemGroup about non-study medic...
ConditionDef : Condition of low systolic blood pressure

LGLOBXFL	LG.LGBLFL	LG.LGDRVFL	LG.LGEVAL	L
ISIT	IS.VISITDY	IS.TAETORD	IS.EPOCH	I
LBNAM	LB.LBLOINC	LB.LBSPEC	LB.LBSPCCND	L
MBSPEC	MB.MBSPCCND	MB.MBLOC	MB.MBLAT	M
MSSTAT	MS.MSREASND	MS.MSXFN	MS.MSNAM	M
MIDIR	MI.MIMETHOD	MI.MILOBXFL	MI.MIBLFL	M
MOPORTOT	MO.MOMETHOD	MO.MOANMETH	MO.MOLOBXFL	M
CVLOBXFL	CV.CVBLFL	CV.CVDRVFL	CV.CVEVAL	C
MKLOBXFL	MK.MKBLFL	MK.MKDRVFL	MK.MKEVAL	M
NVLOBXFL	NV.NVBLFL	NV.NVDRVFL	NV.NVEVAL	N
OESTAT	OE.OEREASND	OE.OEXFN	OE.OELOC	O
VISITDY	RP.TAETORD	RP.EPOCH	RP.RPDTC	F
RELOC	RE.RELAT	RE.REDIR	RE.REMETHOD	F
URLAT	UR.URDIR	UR.URMETHOD	UR.URLOBXFL	U
PCULOQ	PC.VISITNUM	PC.VISIT	PC.VISITDY	F
VISITDY	PE.TAETORD	PE.EPOCH	PE.PEDTC	F
FTEVAL	FT.FTREPNUM	FT.VISITNUM	FT.VISIT	F
EPOCH	QS.QSDTC	QS.QSDY	QS.QSTPT	Q
RSEVALID	RS.RSACPTFL	RS.VISITNUM	RS.VISIT	F
VISIT	TU.VISITDY	TU.TAETORD	TU.EPOCH	T
VISITNUM	TR.VISIT	TR.VISITDY	TR.TAETORD	T
VISITDY	VS.TAETORD	VS.EPOCH	VS.VSDTC	V
VISITDY	FA.TAETORD	FA.EPOCH	FA.FADTC	F
SRBLFL	SR.SREVAL	SR.VISITNUM	SR.VISIT	S
AESEV	AE.AESER	AE.AEACN	AE.AEACNOTH	A

After drag-and-drop:



Import ItemDef: Adverse Event Serious - for SDTM Variable AE.AESER

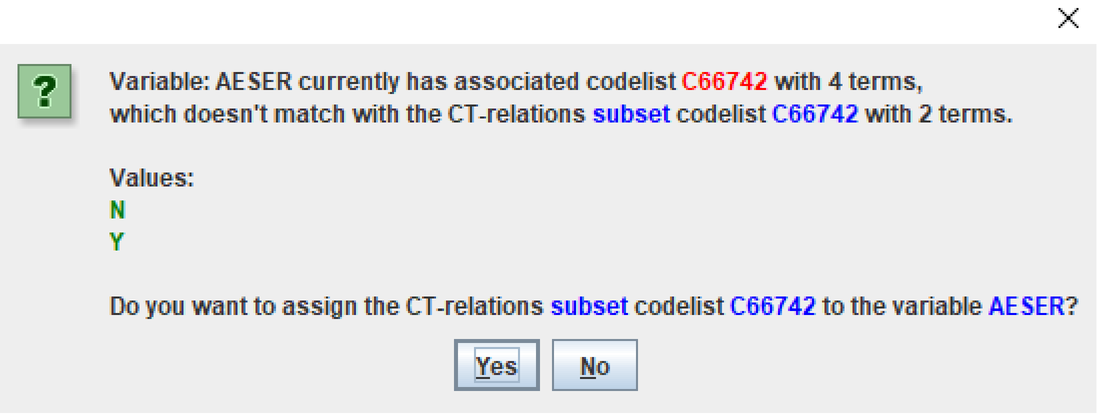
● Import XPath expression for ItemData Value attribute (from Clinical Data)
○ Import XPath expression for another ItemData attribute/subelement (from Clinical Data)
○ Import ItemDef attribute value (static value from Study Definition)

□ Generalize for all StudyEvents	Except for ..	No Exceptions	Only for ..	No Inclusions
□ Generalize for all Forms	Except for ..	No Exceptions	Only for ..	No Inclusions
□ Generalize for all ItemGroups	Except for ..	No Exceptions	Only for ..	No Inclusions
□ Generalize for all Items	Except for ..	No Exceptions	Only for ..	No Inclusions

ODM ItemDef Lenghth: 1   SDTM Variable Length: 2
□ Set SDTM Variable Length to ODM ItemDef Length

□ View/Edit XPath expression (advanced)

followed by:

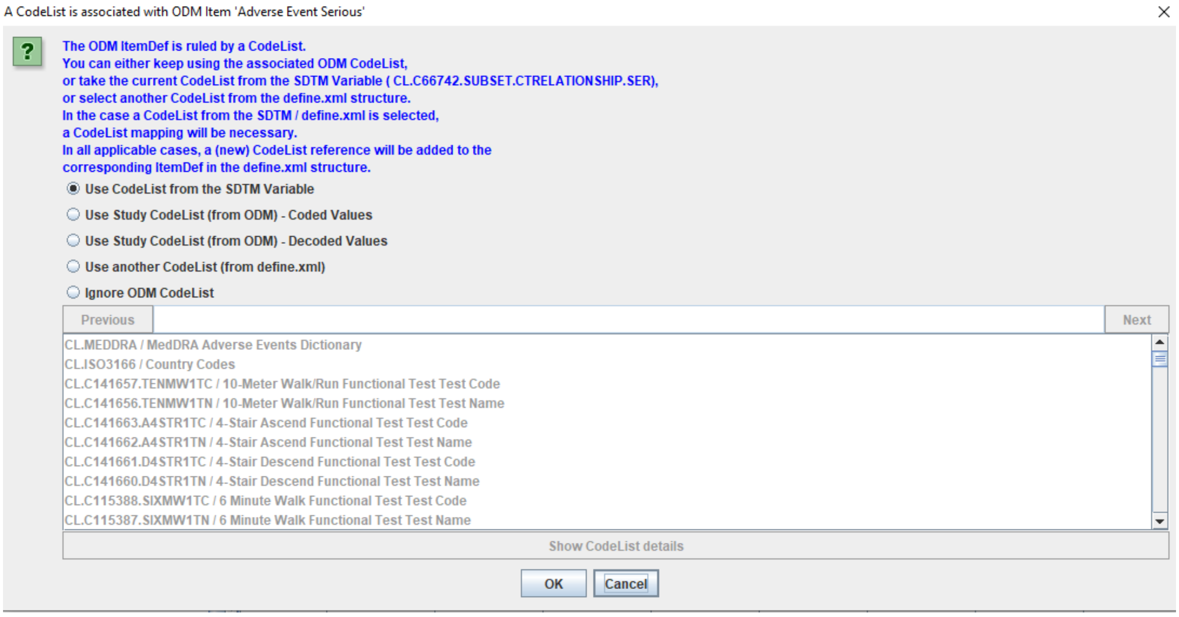


Variable: AESER currently has associated codelist C66742 with 4 terms,
which doesn't match with the CT-relations subset codelist C66742 with 2 terms.

Values:
N
Y

Do you want to assign the CT-relations subset codelist C66742 to the variable AESER?

and when clicking "Yes", the subset codelist is generated and applied to AESER:



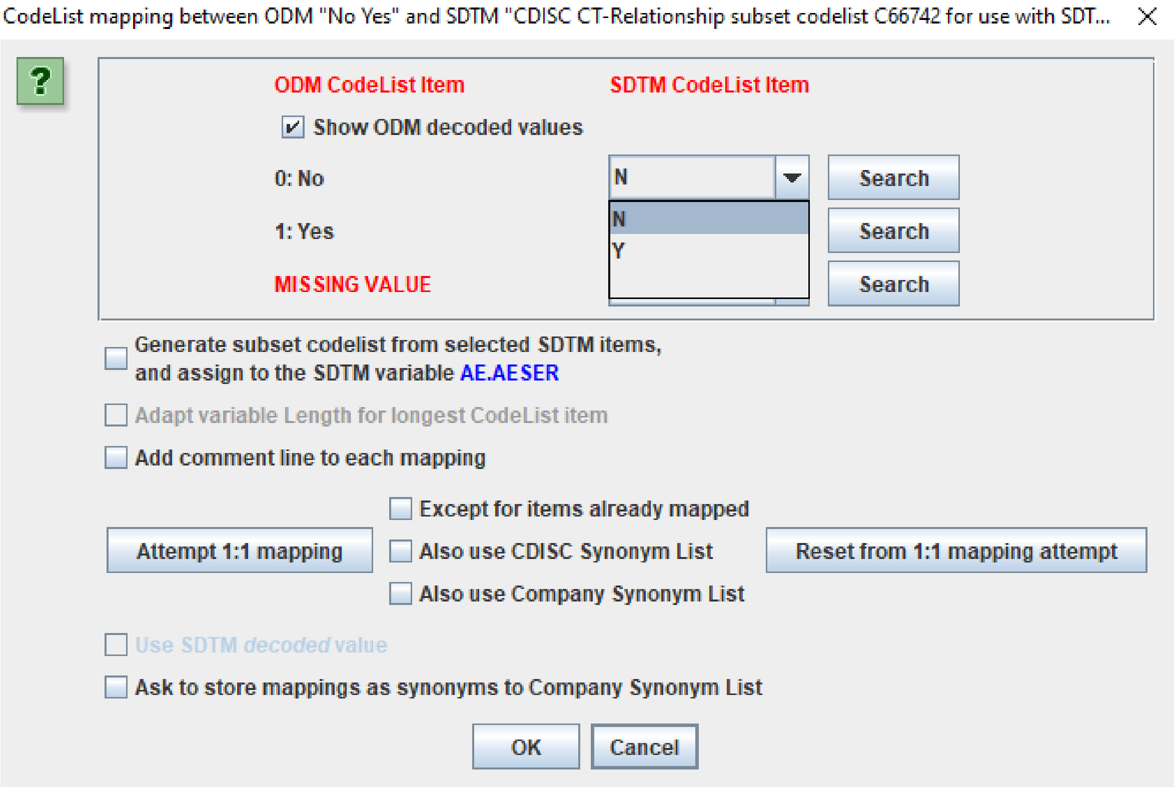
A CodeList is associated with ODM Item 'Adverse Event Serious'

The ODM ItemDef is ruled by a CodeList.
You can either keep using the associated ODM CodeList,
or take the current CodeList from the SDTM Variable (CL.C66742.SUBSET.CTRELATIONSHIP.SER),
or select another CodeList from the define.xml structure.
In the case a CodeList from the SDTM / define.xml is selected,
a CodeList mapping will be necessary.
In all applicable cases, a (new) CodeList reference will be added to the
corresponding ItemDef in the define.xml structure.

● Use CodeList from the SDTM Variable
○ Use Study CodeList (from ODM) - Coded Values
○ Use Study CodeList (from ODM) - Decoded Values
○ Use another CodeList (from define.xml)
○ Ignore ODM CodeList

CL.MEDDRA / MedDRA Adverse Events Dictionary
CL.ISO3166 / Country Codes
CL.C141657.TENMW1TC / 10-Meter Walk/Run Functional Test Test Code
CL.C141656.TENMW1TN / 10-Meter Walk/Run Functional Test Test Name
CL.C141663.A4STR1TC / 4-Stair Ascend Functional Test Test Code
CL.C141662.A4STR1TN / 4-Stair Ascend Functional Test Test Name
CL.C141661.D4STR1TC / 4-Stair Descend Functional Test Test Code
CL.C141660.D4STR1TN / 4-Stair Descend Functional Test Test Name
CL.C115388.SIXMW1TC / 6 Minute Walk Functional Test Test Code
CL.C115387.SIXMW1TN / 6 Minute Walk Functional Test Test Name

followed by the mapping wizard:



CodeList mapping between ODM "No Yes" and SDTM "CDISC CT-Relationship subset codelist C66742 for use with SDT...

ODM CodeList Item	SDTM CodeList Item
0: No	N
1: Yes	
MISSING VALUE	

☑ Show ODM decoded values

☐ Generate subset codelist from selected SDTM items, and assign to the SDTM variable AE.AESER

☐ Adapt variable Length for longest CodeList item

☐ Add comment line to each mapping

☐ Except for items already mapped

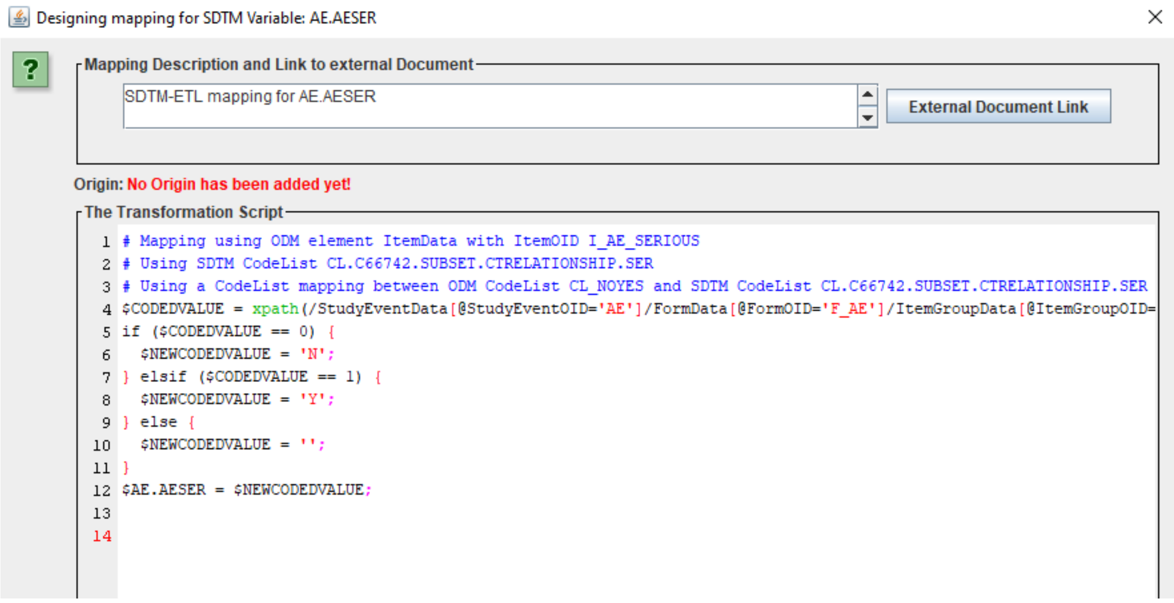
☐ Also use CDISC Synonym List

☐ Also use Company Synonym List

☐ Use SDTM decoded value

☐ Ask to store mappings as synonyms to Company Synonym List

and the generated mapping script:



Designing mapping for SDTM Variable: AE.AESER

Mapping Description and Link to external Document

SDTM-ETL mapping for AE.AESER

Origin: No Origin has been added yet!

The Transformation Script

```
# Mapping using ODM element ItemData with ItemOID I_AE_SERIOUS
# Using SDTM CodeList CL.C66742.SUBSET.CTRELATIONSHIP.SER
# Using a CodeList mapping between ODM CodeList CL_NOYES and SDTM CodeList CL.C66742.SUBSET.CTRELATIONSHIP.SER
$CODEDVALUE = xpath(/StudyEventData[@StudyEventOID='AE']/FormData[@FormOID='F_AE']/ItemGroupData[@ItemGroupOID=
if ($CODEDVALUE == 0) {
  $NEWCODEDVALUE = 'N';
} elsif ($CODEDVALUE == 1) {
  $NEWCODEDVALUE = 'Y';
} else {
  $NEWCODEDVALUE = '';
}
$AE.AESER = $NEWCODEDVALUE;


```

The originally assigned codelist (provided by the SDTMIG table) has 4 terms:
"N", "Y", "NA" and "U"
but the "CDISC Notes" in the SDTMIG only allow "N" and "Y", having been implemented in the "CT Relationships" in electronic format.

Remind that the "CT Relationships" only apply to SDTMIG versions 3.3, and not to SEND or earlier SDTMIG versions.

<u>Current limitations:</u>

In some cases, e.g. for EGSTRESC, more than one codelist is proposed. In such a case, currently only the last one encountered in the "CT Relationships" file is presented to the user. In a next version, in such a case, a dialog will be presented asking the user which one to assign, or to not assign a codelist at all, with the suggestion that a ValueList may be more suitable.

Extended automated --LOBXFL generation

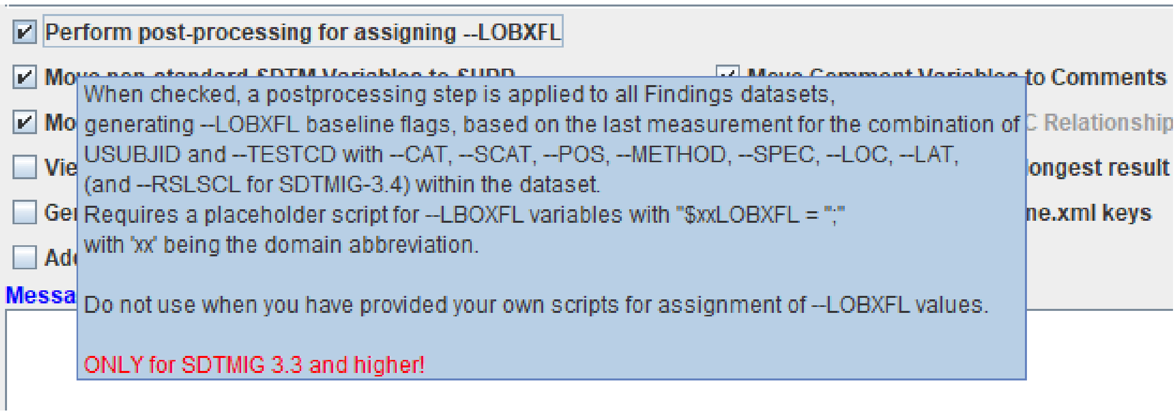
For some time now, for SDTM, FDA requires to have a --LOBXFL variable "Last Observation before First Exposure" for each "Findings" dataset, with the value being "Y" if the record is the last observation for that specific test before the first drug or treatment exposure. FDA nor CDISC however state how "specific test" should be defined.
Some information can however be deduced from the FDA validation rule SD1445 (FDAB026)[5].
Before version 4.3, we implemented the automated calculation of --LOBXFL as a postprocessing step on basis of USUBJID and --TESTCD, i.e. for each unique value of --TESTCD within the records for a single subject, a maximum of one record is assigned to be the "last observation before first exposure" with --LOBXFL=Y.

This may however be not sufficient.
For example, suppose an LB dataset is containing records for both glucose in urine as glucose in blood. These have to be regarded as two separate tests, so in such a case, within each subject, we need to assign a "last observation before first exposure" flag for each separately, meaning that we may have one flag for the combination of LBTESTCD=GLUC with LBSPEC=URINE and one for the combination of LBTESTCD=GLUC with LBSPEC=BLOOD.
Also, for e.g. two blood pressure measurements, one in sitting, and one in lying position, LBTESTCD for both will be SYSBP and/or DIABP, meaning that we may have up to 4 "last observation before first exposure" flags (one for each combination).

As of version 4.3, the automated postprocessing --LOBXFL calculation uses the variables USUBJID, --TESTCD, --CAT, --SCAT, --POS, --METHOD, --SPEC, --LOC and --LAT, extended with --RSLSCL (Result Scale) for the case of SDTMIG-3.4.
The latter is necessary for e.g. the case that there are 2 tests of "glucose in urine", one being quantitative (i.e. a concentration) and one being ordinal (with values of e.g. +1, +2, ...).

This is now also depicted in the tooltip on the checkbox in the "Execute Transformation" window:

[5] See e.g. https://www.fda.gov/media/103587/download



☑ Perform post-processing for assigning --LOBXFL

☑ Mo

☑ Mo

□ Vie

□ Ge

□ Ad

Messa

☑ Move Comment Variables to Comments

C Relationship

ongest result

ne.xml keys

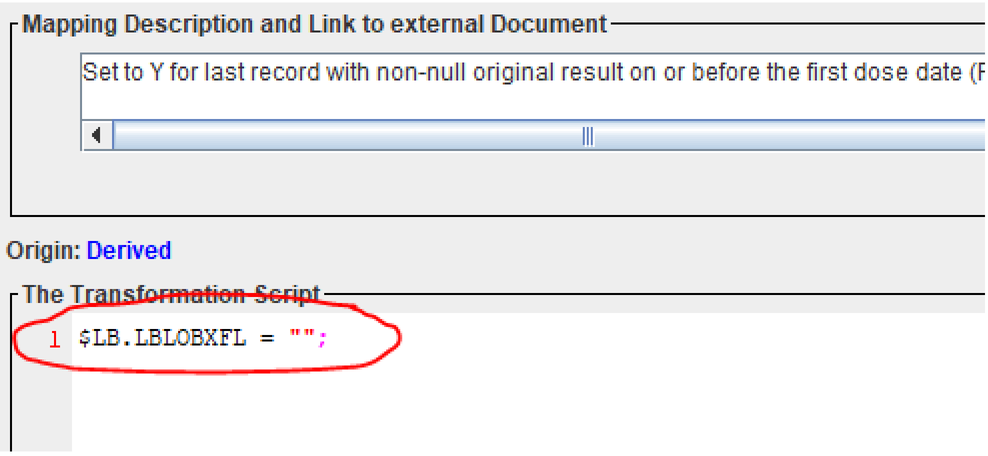
When checked, a postprocessing step is applied to all Findings datasets, generating --LOBXFL baseline flags, based on the last measurement for the combination of USUBJID and --TESTCD with --CAT, --SCAT, --POS, --METHOD, --SPEC, --LOC, --LAT, (and --RSLSCL for SDTMIG-3.4) within the dataset.
Requires a placeholder script for --LBOXFL variables with "$xxLOBXFL = ";"
with 'xx' being the domain abbreviation.

Do not use when you have provided your own scripts for assignment of --LOBXFL values.

ONLY for SDTMIG 3.3 and higher!

The current extension will further help to improve the assignment of --LOBXFL values in an automated way.

Two important remarks:
- in order to have the --LOBXFL values automatically calculated in the postprocessing step, you need to have the --LOBXFL column present, and have a "placeholder" script like:



Mapping Description and Link to external Document

Set to Y for last record with non-null original result on or before the first dose date (F

Origin: Derived

The Transformation Script

```
$LB.LBLOBXFL = "";
```

- Do not use this option when you have provided your own mapping script(s) for --LOBXFL, e.g. based on time points (--TPT or --TPTNUM). For example, if there is a time point "1 minute for first dose", and there is always a collected value for it, you may decide to assign the --LOBXFL flag on basis of this, by having a script that does so for the --LOBXFL variable.

For SDTM-ETL v.4.4 we are further planning that the use can also select to have the --LOBXFL assignment being based on the LOINC code for the test uniqueness.
After all, the LOINC code is the only real identifier for test uniqueness, something that CDISC unfortunately does not want to acknowledge ("not-invented-here" syndrome).

Updated Dataset-JSON generation

CDISC has published the specification for a new, modern transport format (aimed to replace the outdated SAS Transport 5 format), named Dataset-JSON.
A pilot has been set up with the FDA, together with the Phuse organization, who has set up several working groups, for which already very many people have volunteered.
This shows that there is also a strong drive from the industry to finally get rid of SAS

Transport format for regulatory submissions.

We therefore expect that Dataset-JSON will be accepted in 2024 by FDA, and that other regulatory authorities such as PMDA and NMPA will soon follow[6].

We already introduced the generation of SDTM/SEND datasets in Dataset-JSON format in version SDTM-ETL v.4.1. In the mean time however, the specification of Dataset-JSON has slightly changed, so we updated the SDTM-ETL software correspondingly.

Remark that also CDISC's validation (open source) CORE software is currently being updated for Dataset-JSON, which will help FDA and other regulatory authorities to work with as well Dataset-JSON as CORE, essentially making their currently used validation software (which is very buggy and produces many "false positives" superfluous.

Remark that the use of CORE has already been implemented in SDTM-ETL as of version 4.2.

A separate tutorial covering CORE validation is available on the SDTM-ETL website.

CDISC CORE Validation

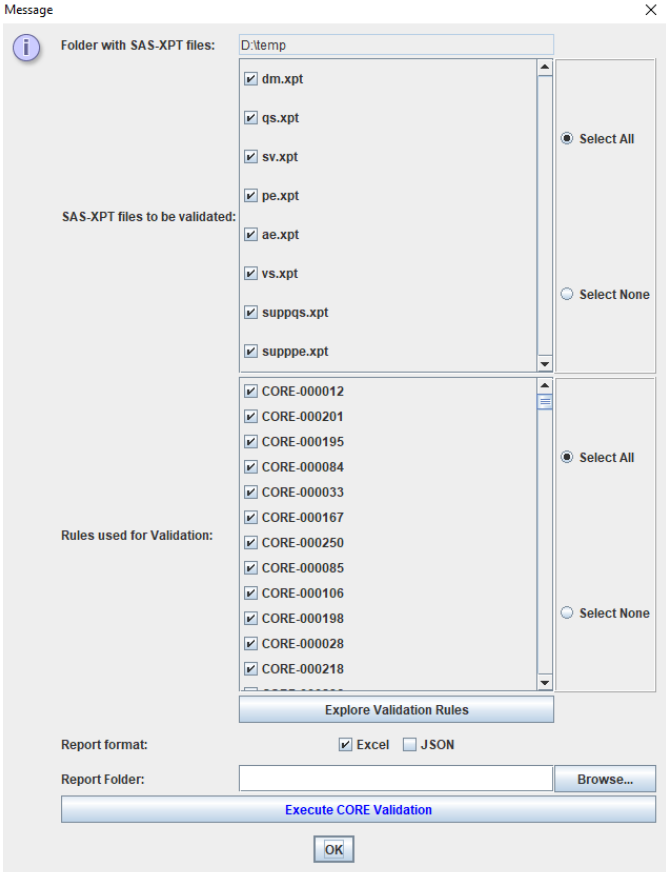
CDISC CORE is a revolution in the area of validation of CDISC datasets for submissions of datasets to the regulatory authorities (and beyond that).

As one of the first software vendors (if not the first), we have implemented CORE in our software, enabling to use CORE from within SDTM-ETL v.4.2. The implementation allows to select on datasets generated, as well as on rules to be executed during the validation process.

As CORE is still evolving, we have implemented it so that the CORE engine can easily be exchanged for a newer version (which we will make available regularly) without the need of an SDTM-ETL software update.

A separate tutorial covering CORE validation is available on the SDTM-ETL website.

[6] Especially for PMDA (Japan) and NMPA (China), getting rid of SAS Transport format, and moving to a modern JSON format would be a huge step forward, as SAS Transport does not support Asian characters. JSON however uses Unicode by default, fully supporting all characters of any written language in the world.



Message

Folder with SAS-XPT files: D:\temp

SAS-XPT files to be validated:

- ☑ dm.xpt
- ☑ qs.xpt
- ☑ sv.xpt
- ☑ pe.xpt
- ☑ ae.xpt
- ☑ vs.xpt
- ☑ suppqs.xpt
- ☑ suppe.xpt

● Select All ○ Select None

Rules used for Validation:

- ☑ CORE-000012
- ☑ CORE-000201
- ☑ CORE-000195
- ☑ CORE-000084
- ☑ CORE-000033
- ☑ CORE-000167
- ☑ CORE-000250
- ☑ CORE-000085
- ☑ CORE-000106
- ☑ CORE-000198
- ☑ CORE-000028
- ☑ CORE-000218

● Select All ○ Select None

Report format: ☑ Excel □ JSON

Report Folder:

Define.xml validation - RESTful Web Services

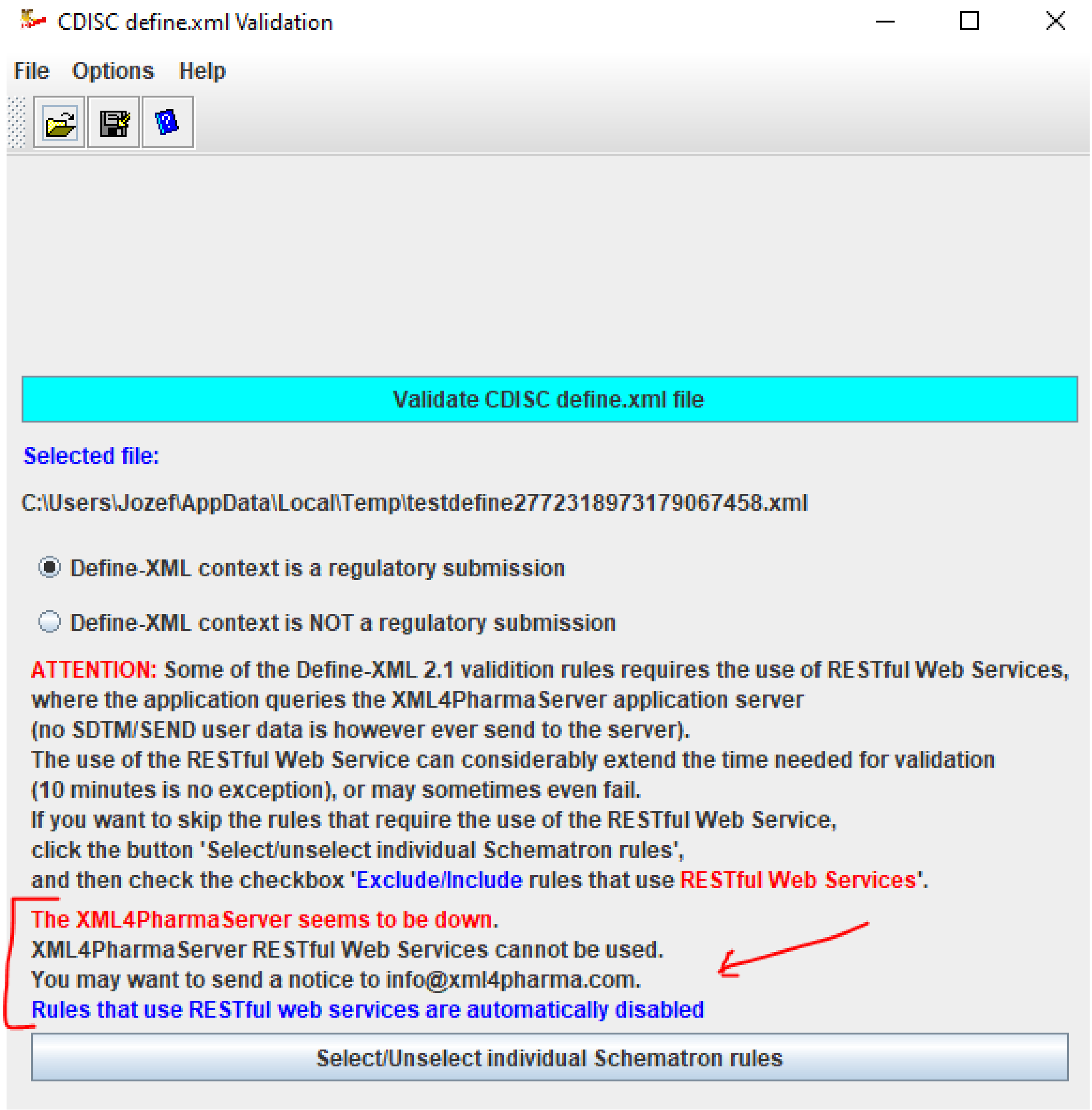
The define.xml validation (menu "Validate - Validate define.xml" uses Schematron to implement the validation rules published by the CDISC Define-XML team.
For Define-XML 2.1, some of these rules check against the SDTM or SEND standard, for example the requirement that when a variable is "required", the "Mandatory" attribute in the define.xml is set to "Yes". Such rules require the use of a RESTful Web Service (RWS) that was developed for this purpose. See http://xml4pharmaserver.com/WebServices/ for details about the RWS used.

The use of the RWS for some of the validation rules can be relative slow, so that validating the full define.xml can take up to 10 minutes (e.g. when the internet connection is slow).

When there is no internet connection, or the RWS server is down (which can happen), there was no way until now to detect this before starting the validation, and the validation takes even more time, and is of course incomplete.

In SDTM-ETL v.4.3, we added a feature that, before starting the validation, checks whether the RWS server can be reached and is working correctly.

If it isn't, additional information about this is provided in the dialog from where one can start the validation:



Selected file:

C:\Users\Jozef\AppData\Local\Temp\testdefine2772318973179067458.xml

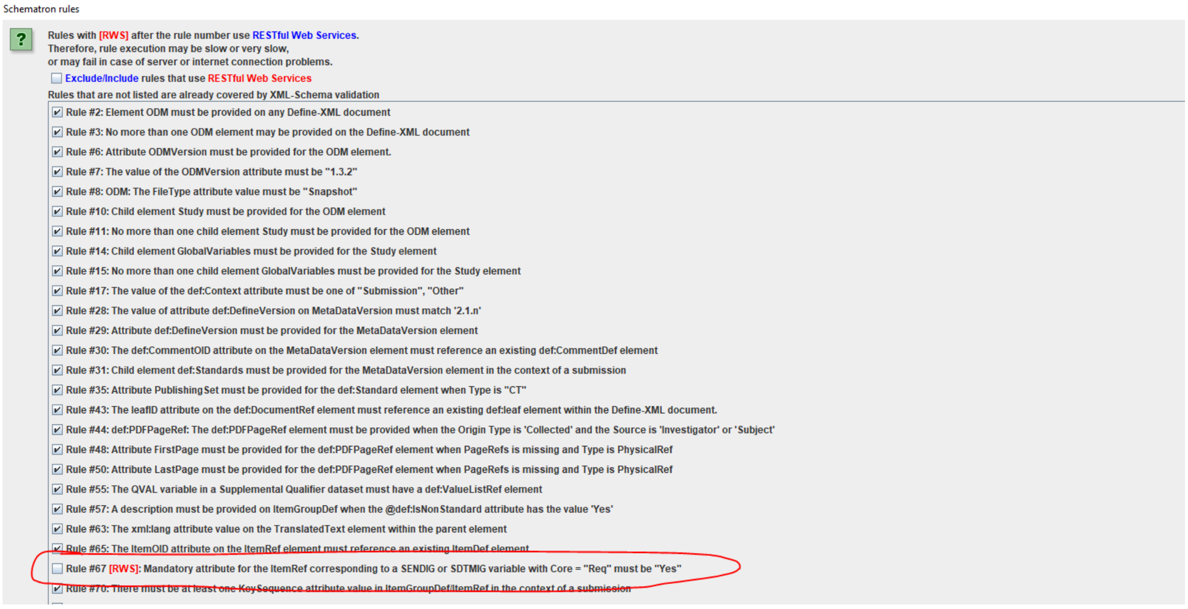
● Define-XML context is a regulatory submission

○ Define-XML context is NOT a regulatory submission

ATTENTION: Some of the Define-XML 2.1 validition rules requires the use of RESTful Web Services,
where the application queries the XML4PharmaServer application server
(no SDTM/SEND user data is however ever send to the server).
The use of the RESTful Web Service can considerably extend the time needed for validation
(10 minutes is no exception), or may sometimes even fail.
If you want to skip the rules that require the use of the RESTful Web Service,
click the button 'Select/unselect individual Schematron rules',
and then check the checkbox 'Exclude/Include rules that use RESTful Web Services'.

The XML4PharmaServer seems to be down.
XML4PharmaServer RESTful Web Services cannot be used.
You may want to send a notice to info@xml4pharma.com.
Rules that use RESTful web services are automatically disabled

and automatically, all rules that require the use of the RWS are disabled. One can easily see this by clicking the "Select/Unselect individual Schematron rules":



Schematron rules

Rules with [RWS] after the rule number use RESTful Web Services.
Therefore, rule execution may be slow or very slow,
or may fail in case of server or internet connection problems.

□ Exclude/Include rules that use RESTful Web Services

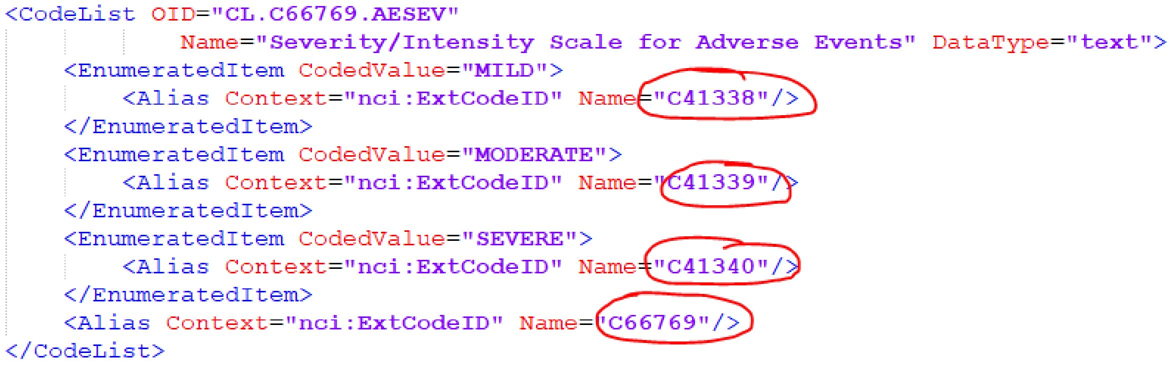
Rules that are not listed are already covered by XML-Schema validation

- ☑ Rule #2: Element ODM must be provided on any Define-XML document
- ☑ Rule #3: No more than one ODM element may be provided on the Define-XML document
- ☑ Rule #6: Attribute ODMVersion must be provided for the ODM element.
- ☑ Rule #7: The value of the ODMVersion attribute must be "1.3.2"
- ☑ Rule #8: ODM: The FileType attribute value must be "Snapshot"
- ☑ Rule #10: Child element Study must be provided for the ODM element
- ☑ Rule #11: No more than one child element Study must be provided for the ODM element
- ☑ Rule #14: Child element GlobalVariables must be provided for the Study element
- ☑ Rule #15: No more than one child element GlobalVariables must be provided for the Study element
- ☑ Rule #17: The value of the def:Context attribute must be one of "Submission", "Other"
- ☑ Rule #28: The value of attribute def:DefineVersion on MetaDataVersion must match '2.1.n'
- ☑ Rule #29: Attribute def:DefineVersion must be provided for the MetaDataVersion element
- ☑ Rule #30: The def:CommentOID attribute on the MetaDataVersion element must reference an existing def:CommentDef element
- ☑ Rule #31: Child element def:Standards must be provided for the MetaDataVersion element in the context of a submission
- ☑ Rule #35: Attribute PublishingSet must be provided for the def:Standard element when Type is "CT"
- ☑ Rule #43: The leafID attribute on the def:DocumentRef element must reference an existing def:leaf element within the Define-XML document.
- ☑ Rule #44: def:PDFPageRef: The def:PDFPageRef element must be provided when the Origin Type is 'Collected' and the Source is 'Investigator' or 'Subject'
- ☑ Rule #48: Attribute FirstPage must be provided for the def:PDFPageRef element when PageRefs is missing and Type is PhysicalRef
- ☑ Rule #50: Attribute LastPage must be provided for the def:PDFPageRef element when PageRefs is missing and Type is PhysicalRef
- ☑ Rule #55: The QVAL variable in a Supplemental Qualifier dataset must have a def:ValueListRef element
- ☑ Rule #57: A description must be provided on ItemGroupDef when the @def:IsNonStandard attribute has the value 'Yes'
- ☑ Rule #63: The xml:lang attribute value on the TranslatedText element within the parent element
- ☑ Rule #65: The ItemOID attribute on the ItemRef element must reference an existing ItemDef element
- □ Rule #67 [RWS]: Mandatory attribute for the ItemRef corresponding to a SENDIG or SDTMIG variable with Core = "Req" must be "Yes"
- ☑ Rule #70: There must be at least one KeySequence attribute value in ItemGroupDef/ItemRef in the context of a submission

As one sees, rule 67 has been disabled, as it uses the RWS, which at that time, is not available as the server at that moment was down.

Cleaning up define.xml: non-CDISC-CT aliases

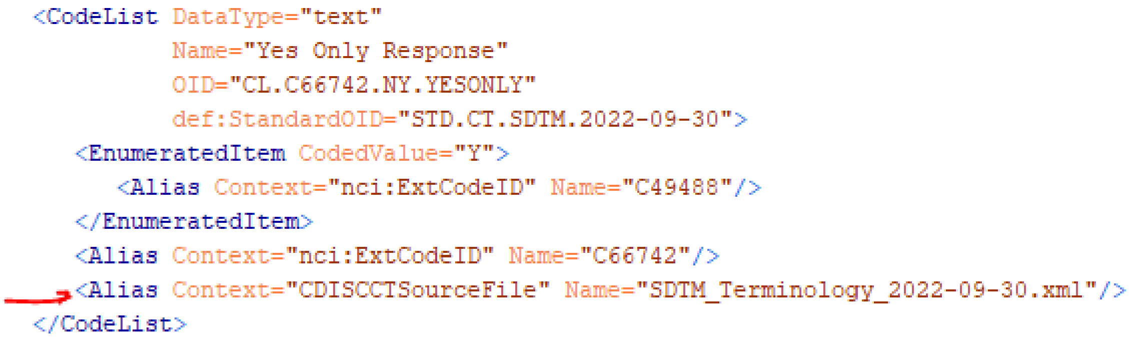
In SDTM-ETL, it is possible to add different "Alias" elements for SDTM variables. The best known one in the "Alias" providing the CDISC-NCI code for codelists and codelists items. For example:



```
<CodeList OID="CL.C66769.AESEV"
            Name="Severity/Intensity Scale for Adverse Events" DataType="text">
    <EnumeratedItem CodedValue="MILD">
        <Alias Context="nci:ExtCodeID" Name="C41338"/>
    </EnumeratedItem>
    <EnumeratedItem CodedValue="MODERATE">
        <Alias Context="nci:ExtCodeID" Name="C41339"/>
    </EnumeratedItem>
    <EnumeratedItem CodedValue="SEVERE">
        <Alias Context="nci:ExtCodeID" Name="C41340"/>
    </EnumeratedItem>
    <Alias Context="nci:ExtCodeID" Name="C66769"/>
</CodeList>
```

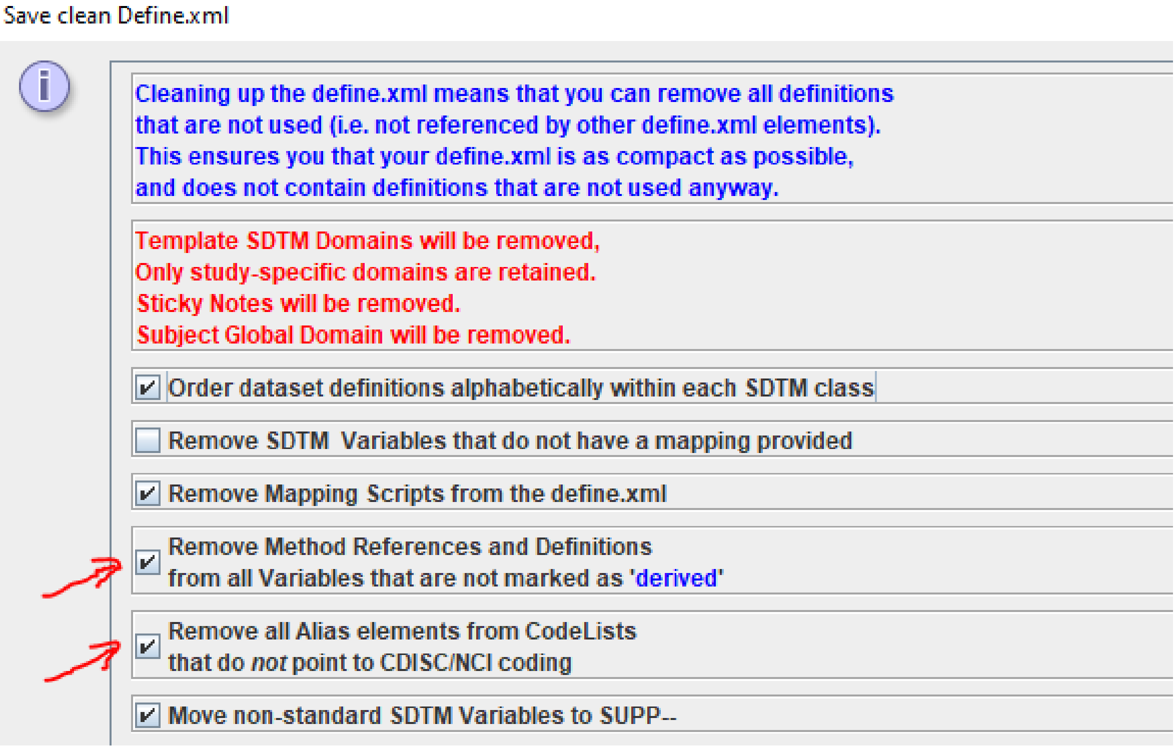
providing the CDISC-NCI codes for as well the codelist itself (C66769) as well as for the individual items. More and more, CDISC is using this CDISC-NCI code as the unique identifier. In some of the codelists, the "CodedValue" is now the NCI code.
That it is a CDISC-NCI code is indicated by the value of the "Context" attribute, which must be "nci:ExtCodeID" in the case of a CDISC codelist.

During the mapping however, other "Alias" elements may have been added, sometimes even automatically. For example:



```
<CodeList DataType="text"
          Name="Yes Only Response"
          OID="CL.C66742.NY.YESONLY"
          def:StandardOID="STD.CT.SDTM.2022-09-30">
   <EnumeratedItem CodedValue="Y">
      <Alias Context="nci:ExtCodeID" Name="C49488"/>
   </EnumeratedItem>
   <Alias Context="nci:ExtCodeID" Name="C66742"/>
   <Alias Context="CDISCCTSourceFile" Name="SDTM_Terminology_2022-09-30.xml"/>
</CodeList>
```

Often, when "cleaning up" the define.xml using the menu "File - Save cleaned define.xml", one will often want to only retain the "Alias" elements that point to CDISC controlled terminology. This is now achieved by a new checkbox in the wizard:



Save clean Define.xml

Cleaning up the define.xml means that you can remove all definitions that are not used (i.e. not referenced by other define.xml elements). This ensures you that your define.xml is as compact as possible, and does not contain definitions that are not used anyway.

Template SDTM Domains will be removed,
Only study-specific domains are retained.
Sticky Notes will be removed.
Subject Global Domain will be removed.

☑ Order dataset definitions alphabetically within each SDTM class

☐ Remove SDTM Variables that do not have a mapping provided

☑ Remove Mapping Scripts from the define.xml

☑ Remove Method References and Definitions from all Variables that are not marked as 'derived'

☑ Remove all Alias elements from CodeLists that do *not* point to CDISC/NCI coding

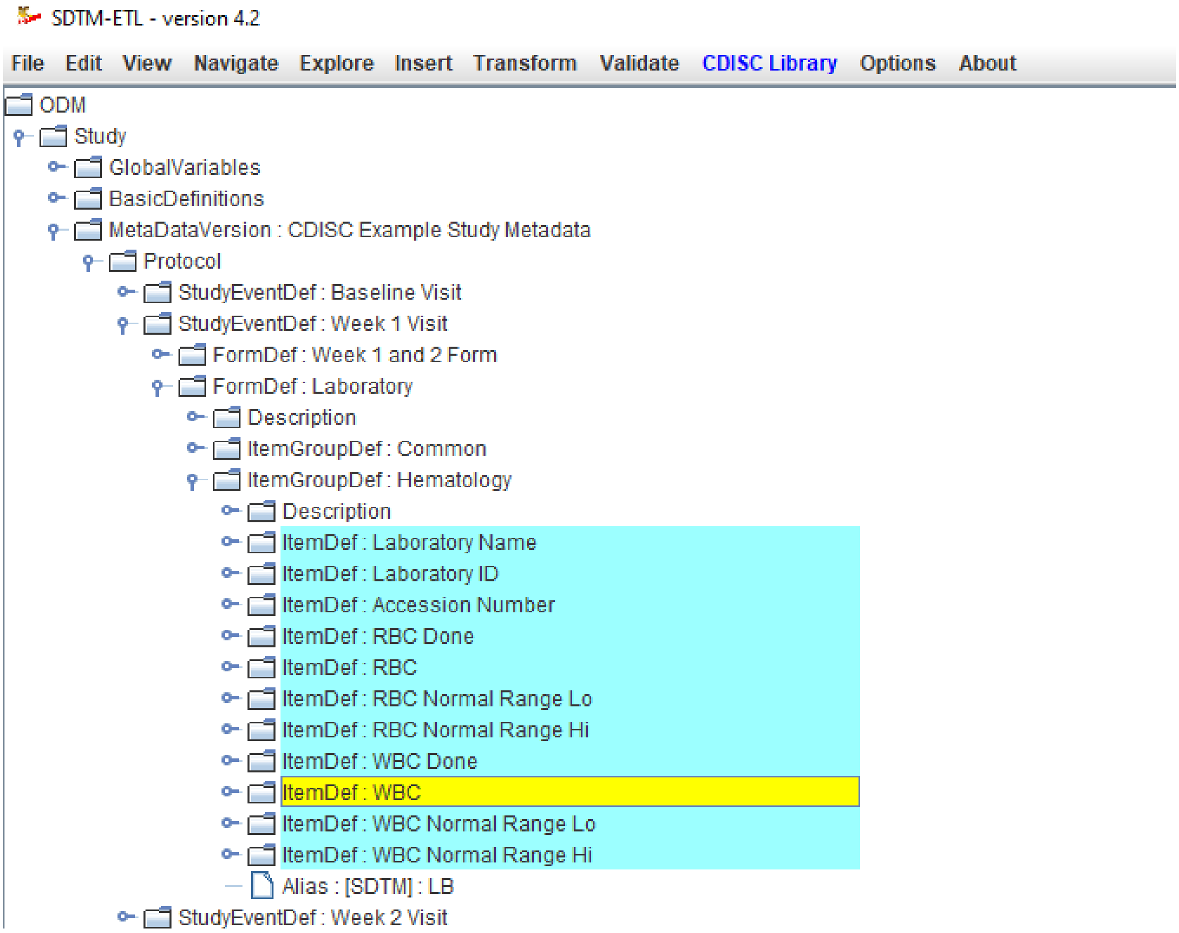
☑ Move non-standard SDTM Variables to SUPP--

Also new is the checkbox "Remove Method References and Definitions from all Variables that are not marked as "derived". It takes care that all references to, and the method definitions are removed for all SDTM/SEND variables for which the "Origin" is not "derived", so only retaining the ones for "derived" variables.

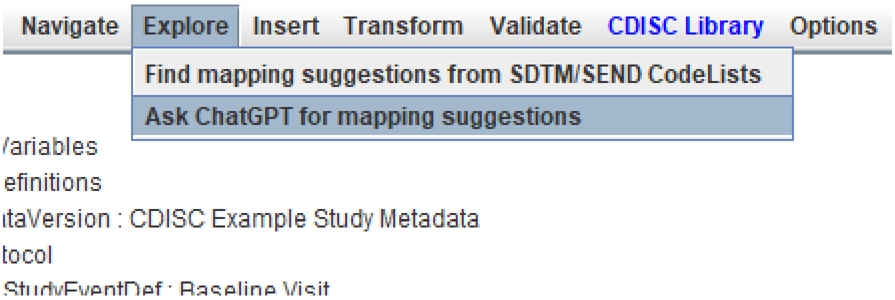
ChatGPT and word similarity use for mapping suggestions

ChatGPT (based on artificial intelligence - AI) has, for many of us, become part of our daily life. Especially for SDTM beginners, it can provide reasonable hints for mappings. Therefore, we have build an interface to ChatGPT into SDTM-ETL v.4.2.
Its use however requires that the user has obtain a ChatGPT API key, which needs to be added to the "properties.dat" file (see further on).

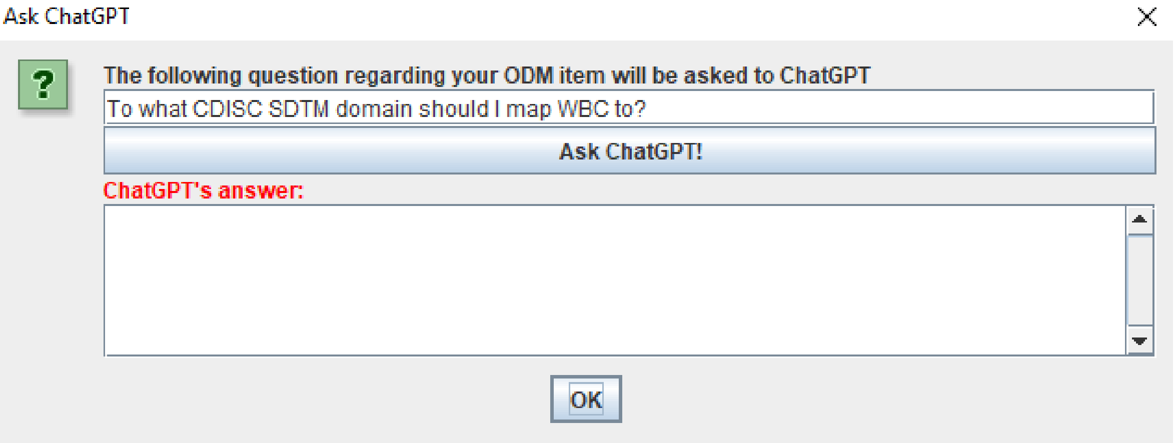
In order to use ChatGPT for obtaining a mapping hint, first select an item in the ODM tree, for example "WBC":


SDTM-ETL - version 4.2
File
Edit
View
Navigate
Explore
Insert
Transform
Validate
CDISC Library
Options
About
ODM
Study
GlobalVariables
BasicDefinitions
MetaDataVersion : CDISC Example Study Metadata
Protocol
StudyEventDef : Baseline Visit
StudyEventDef : Week 1 Visit
FormDef : Week 1 and 2 Form
FormDef : Laboratory
Description
ItemGroupDef : Common
ItemGroupDef : Hematology
Description
ItemDef : Laboratory Name
ItemDef : Laboratory ID
ItemDef : Accession Number
ItemDef : RBC Done
ItemDef : RBC
ItemDef : RBC Normal Range Lo
ItemDef : RBC Normal Range Hi
ItemDef : WBC Done
ItemDef : WBC
ItemDef : WBC Normal Range Lo
ItemDef : WBC Normal Range Hi
Alias : [SDTM] : LB
StudyEventDef : Week 2 Visit

Then use the (new) menu "Explore - Ask ChatGPT for mapping suggestions",


Navigate
Explore
Insert
Transform
Validate
CDISC Library
Options
Find mapping suggestions from SDTM/SEND CodeLists
Ask ChatGPT for mapping suggestions
/ariables
efinitions
ıtaVersion : CDISC Example Study Metadata
tocol
StudyEventDef : Baseline Visit

leading to a pre-filled dialog:

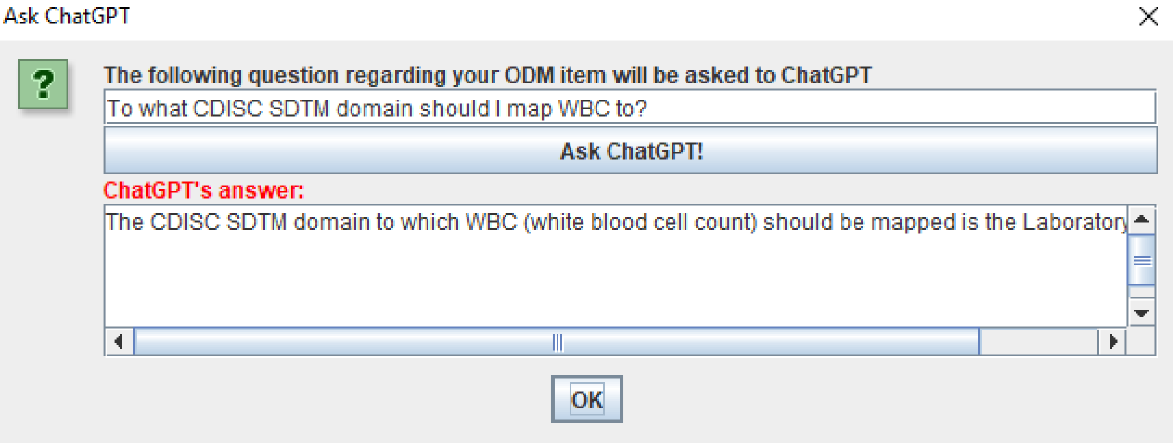


The following question regarding your ODM item will be asked to ChatGPT

To what CDISC SDTM domain should I map WBC to?

ChatGPT's answer:

One can of course than still change the wording of the question.
Clicking "Ask ChatGPT" then leads (after a few seconds) to:



The following question regarding your ODM item will be asked to ChatGPT

To what CDISC SDTM domain should I map WBC to?

ChatGPT's answer:

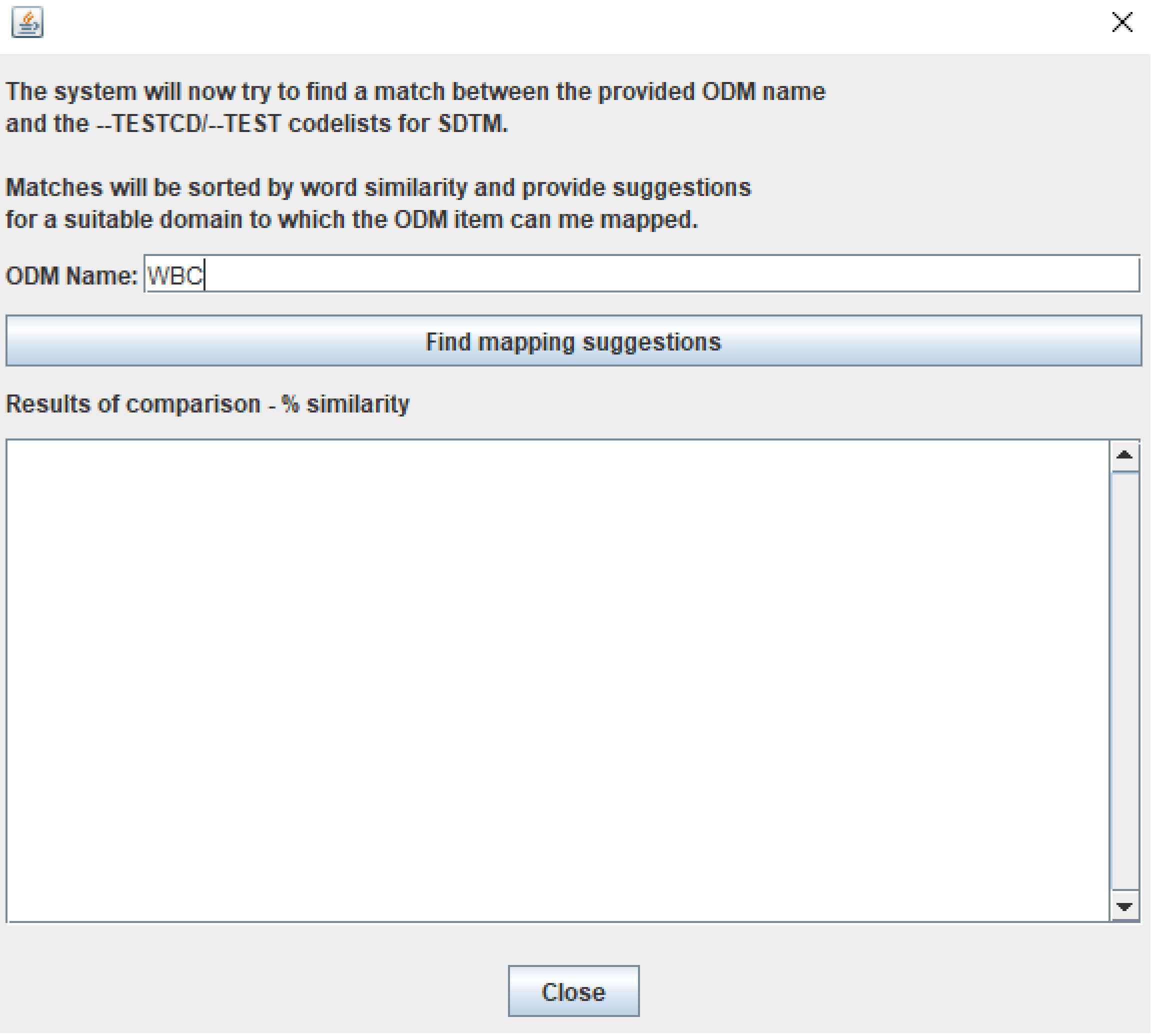
The CDISC SDTM domain to which WBC (white blood cell count) should be mapped is the Laboratory

with ChatGPT's answer: "*The CDISC SDTM domain to which WBC (white blood cell count) should be mapped is the Laboratory (LB) domain.*"

However, when one changes the question into: "To what CDISC SDTM domain and SDTM variable should I map WBC to?", ChatGPT's answer is not entirely correct: "*The CDISC SDTM domain for WBC (white blood cell count) is LBS (Laboratory Test Results). The SDTM variable for WBC is LBSTRES (Laboratory Test Result).*", as there is no SDTM variable "LBSTRES". The answer should be "LBORRES", and some information could be provided about the use of LBSTRESN and LBSTRESC.

We expect however that ChatGPT will rapidly become better, also for clinical research and for mapping to SDTM and SEND, as there is already so much knowledge available in articles, forum discussions and blogs.

Another possibility to obtain a mapping suggestion is based on word similarity between the ODM item name (and the question) and the SDTM coded value and decode in CDISC codelists. To use it, select an item in the ODM tree (such as "WBC") and then use the menu "Explore - Find mapping suggestions from SDTM/SEND codelists". This leads to the following dialog:



The system will now try to find a match between the provided ODM name
and the --TESTCD/--TEST codelists for SDTM.

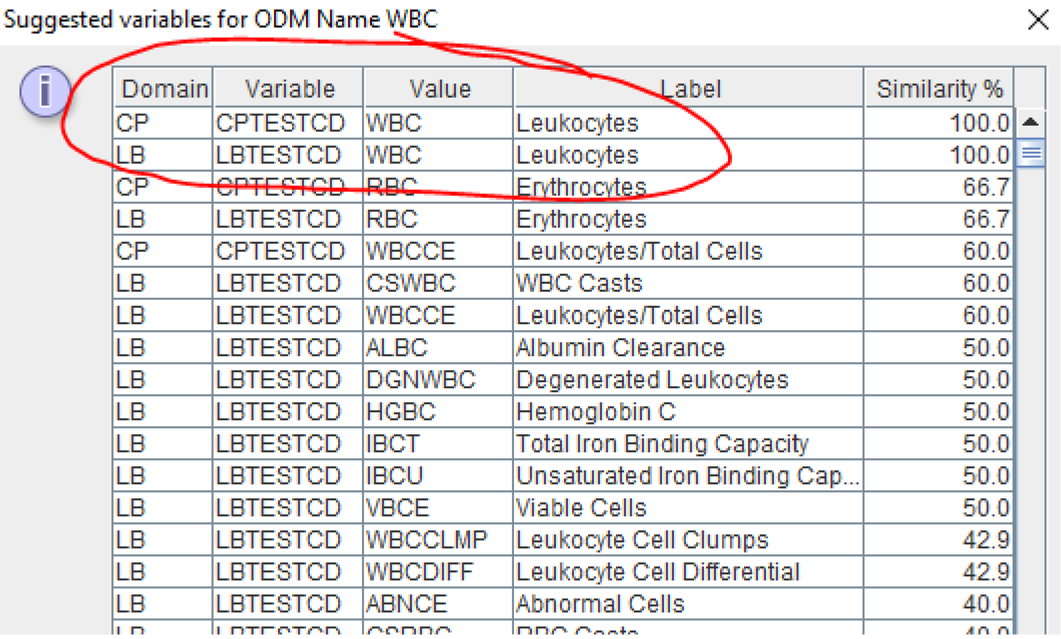
Matches will be sorted by word similarity and provide suggestions
for a suitable domain to which the ODM item can me mapped.

ODM Name: WBC

Results of comparison - % similarity

Clicking "Find mapping suggestions" then starts a process, the system going to all the CDISC codelists, and then sorting according to word similarity. This may take a few minutes, so maybe time to go for a cup of tea or coffee.

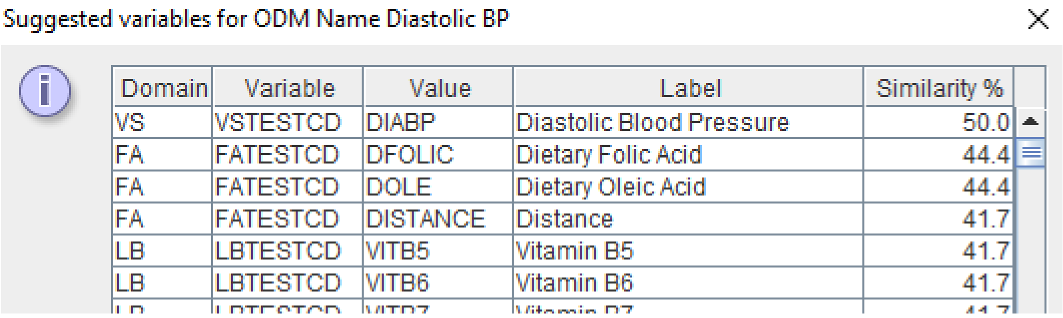
When finished, we obtain:



Suggested variables for ODM Name WBC

Domain	Variable	Value	Label	Similarity %
CP	CPTESTCD	WBC	Leukocytes	100.0
LB	LBTESTCD	WBC	Leukocytes	100.0
CP	CPTESTCD	RBC	Erythrocytes	66.7
LB	LBTESTCD	RBC	Erythrocytes	66.7
CP	CPTESTCD	WBCCE	Leukocytes/Total Cells	60.0
LB	LBTESTCD	CSWBC	WBC Casts	60.0
LB	LBTESTCD	WBCCE	Leukocytes/Total Cells	60.0
LB	LBTESTCD	ALBC	Albumin Clearance	50.0
LB	LBTESTCD	DGNWBC	Degenerated Leukocytes	50.0
LB	LBTESTCD	HGBC	Hemoglobin C	50.0
LB	LBTESTCD	IBCT	Total Iron Binding Capacity	50.0
LB	LBTESTCD	IBCU	Unsaturated Iron Binding Cap...	50.0
LB	LBTESTCD	VBCE	Viable Cells	50.0
LB	LBTESTCD	WBCCLMP	Leukocyte Cell Clumps	42.9
LB	LBTESTCD	WBCDIFF	Leukocyte Cell Differential	42.9
LB	LBTESTCD	ABNCE	Abnormal Cells	40.0
LB	LBTESTCD	CSRBC	RBC Casts	40.0

with 2 good hits (100% similarity) for CPTESTCD (in domain CP - Cell Phenotype Findings) and LBTESTCD (in domain LB - Laboratory Test Findings).
Using e.g. "Diastolic BP" will lead to:

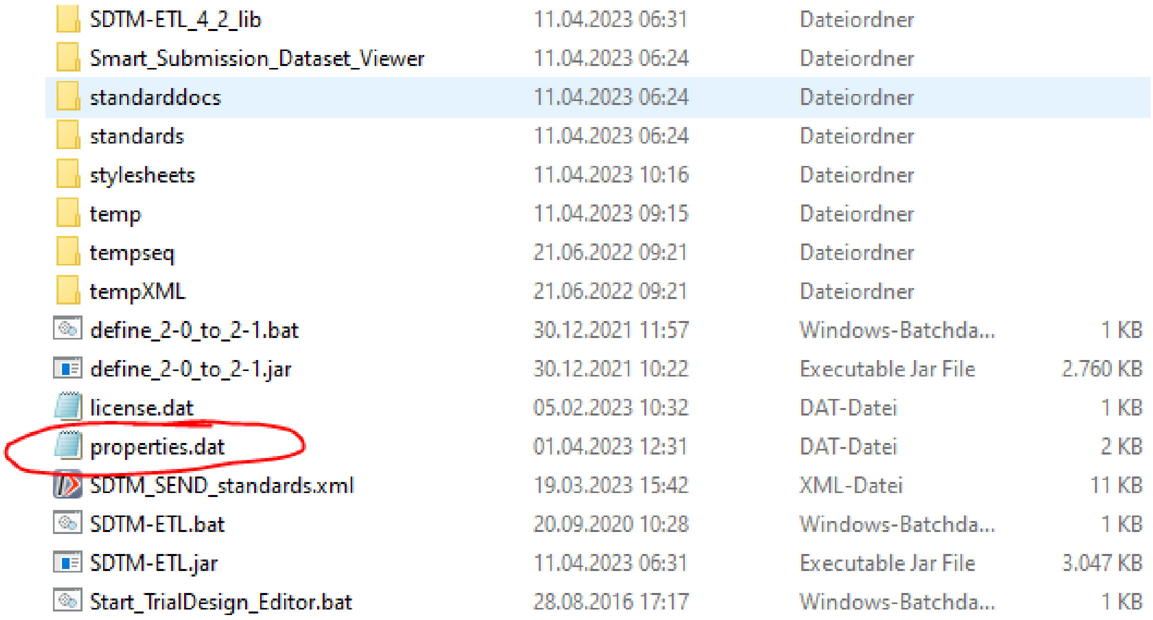

Suggested variables for ODM Name Diastolic BP

Domain	Variable	Value	Label	Similarity %
VS	VSTESTCD	DIABP	Diastolic Blood Pressure	50.0
FA	FATESTCD	DFOLIC	Dietary Folic Acid	44.4
FA	FATESTCD	DOLE	Dietary Oleic Acid	44.4
FA	FATESTCD	DISTANCE	Distance	41.7
LB	LBTESTCD	VITB5	Vitamin B5	41.7
LB	LBTESTCD	VITB6	Vitamin B6	41.7
LB	LBTESTCD	VITB7	Vitamin B7	41.7

We expect that the use of AI and similar technologies for helping in SDTM and SEND mapping will in future further grow. Due to the modular design of SDTM-ETL, we can easily add interfaces with systems that provide such mapping suggestions, e.g. through private or public APIs.

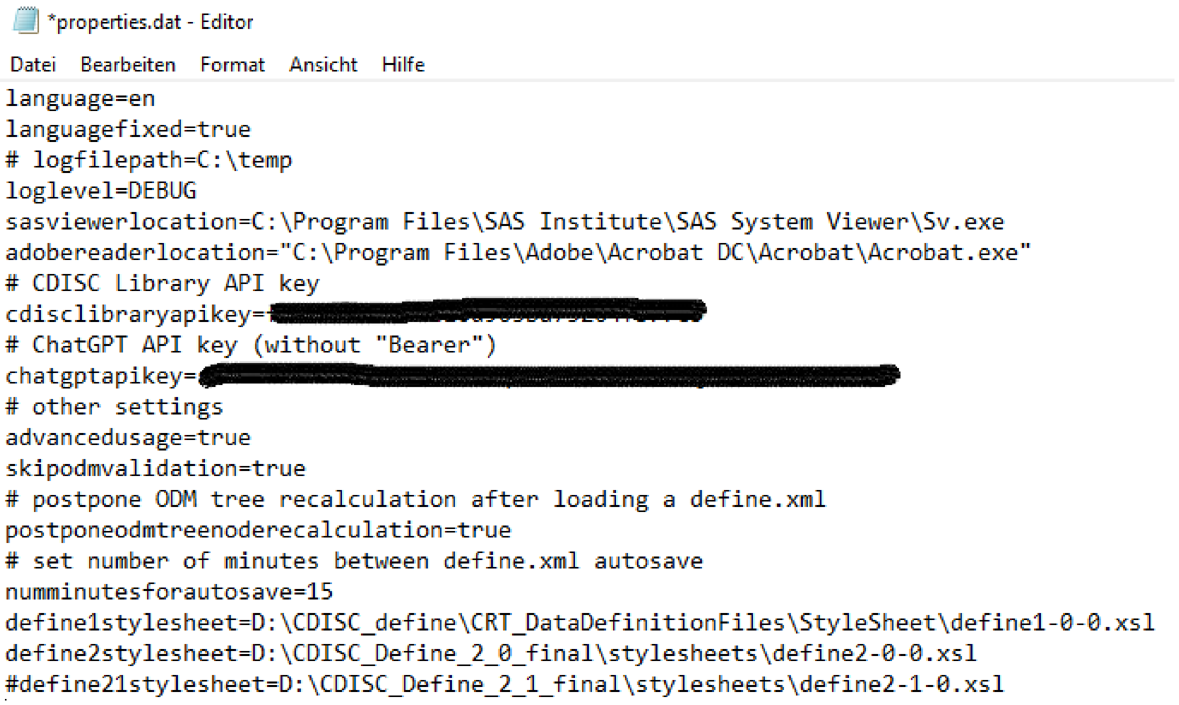
Additional parameters in the "properties.dat" file

When starting up the software, one of the first things done is to read the "properties.dat" file which can be found in the directory where the software was installed:



SDTM-ETL_4_2_lib	11.04.2023 06:31	Dateiordner	
Smart_Submission_Dataset_Viewer	11.04.2023 06:24	Dateiordner	
standarddocs	11.04.2023 06:24	Dateiordner	
standards	11.04.2023 06:24	Dateiordner	
stylesheets	11.04.2023 10:16	Dateiordner	
temp	11.04.2023 09:15	Dateiordner	
tempseq	21.06.2022 09:21	Dateiordner	
tempXML	21.06.2022 09:21	Dateiordner	
define_2-0_to_2-1.bat	30.12.2021 11:57	Windows-Batchda...	1 KB
define_2-0_to_2-1.jar	30.12.2021 10:22	Executable Jar File	2.760 KB
license.dat	05.02.2023 10:32	DAT-Datei	1 KB
properties.dat	01.04.2023 12:31	DAT-Datei	2 KB
SDTM_SEND_standards.xml	19.03.2023 15:42	XML-Datei	11 KB
SDTM-ETL.bat	20.09.2020 10:28	Windows-Batchda...	1 KB
SDTM-ETL.jar	11.04.2023 06:31	Executable Jar File	3.047 KB
Start_TrialDesign_Editor.bat	28.08.2016 17:17	Windows-Batchda...	1 KB

One can edit this file with any simple text editor (but do not use MS-Word), for example with the simple MS "Editor" or with NotePad or NotePad++. For example:

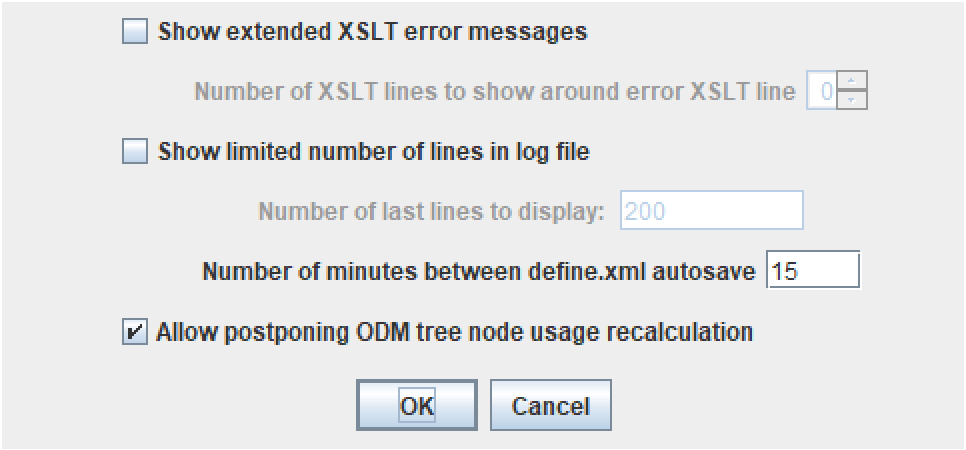


```
language=en
languagefixed=true
# logfilepath=C:\temp
loglevel=DEBUG
sasviewerlocation=C:\Program Files\SAS Institute\SAS System Viewer\Sv.exe
adobereaderlocation="C:\Program Files\Adobe\Acrobat DC\Acrobat\Acrobat.exe"
# CDISC Library API key
cdisclibraryapikey=[redacted]
# ChatGPT API key (without "Bearer")
chatgptapikey=[redacted]
# other settings
advancedusage=true
skipodmvalidation=true
# postpone ODM tree recalculation after loading a define.xml
postponeodmtreenoderecalculation=true
# set number of minutes between define.xml autosave
numminutesforautosave=15
define1stylesheet=D:\CDISC_define\CRT_DataDefinitionFiles\StyleSheet\define1-0-0.xsl
define2stylesheet=D:\CDISC_Define_2_0_final\stylesheets\define2-0-0.xsl
#define21stylesheet=D:\CDISC_Define_2_1_final\stylesheets\define2-1-0.xsl
```

where lines starting with a "#" are comment lines.

It also allows to set the API keys for the CDISC Library (parameter "cdisclibrarykey") and ChatGPT (parameter "chatgptkey").
New parameters as of SDTM-ETL v.4.2 that can be set are:

- "postponeodmtreenoderecalculation": when setting to "true" (default is "false"), when loading a define.xml with mappings, the use of the ODM items in the mappings (color coding in the ODM tree) will not automatically be done immediately, but the user will be asked whether he/she wants to further postpone it, or execute it immediately.
The same can also be achieved by the menu "Options - Settings" by checking the checkbox "Allow postponing ODM tree node usage recalculation":



☐ Show extended XSLT error messages

Number of XSLT lines to show around error XSLT line 0

☐ Show limited number of lines in log file

Number of last lines to display: 200

Number of minutes between define.xml autosave 15

☑ Allow postponing ODM tree node usage recalculation

Users however have asked us to have this as a startup property, so we added the corresponding parameter to the "properties.dat" file.

- "numminutesforautosave": allows to set the number of minutes between autosaving the define.xml structure with the mappings at startup time. Some users have complained that the default of 5 minutes between autosaving is too short, and they want to have a higher default

value (e.g. 15 minutes). This can now be set in the "properties.dat" file.
Also remark that during loading or merging the define.xml from/to file, autosaving is automatically skipped, in order to avoid interference with the loading process.

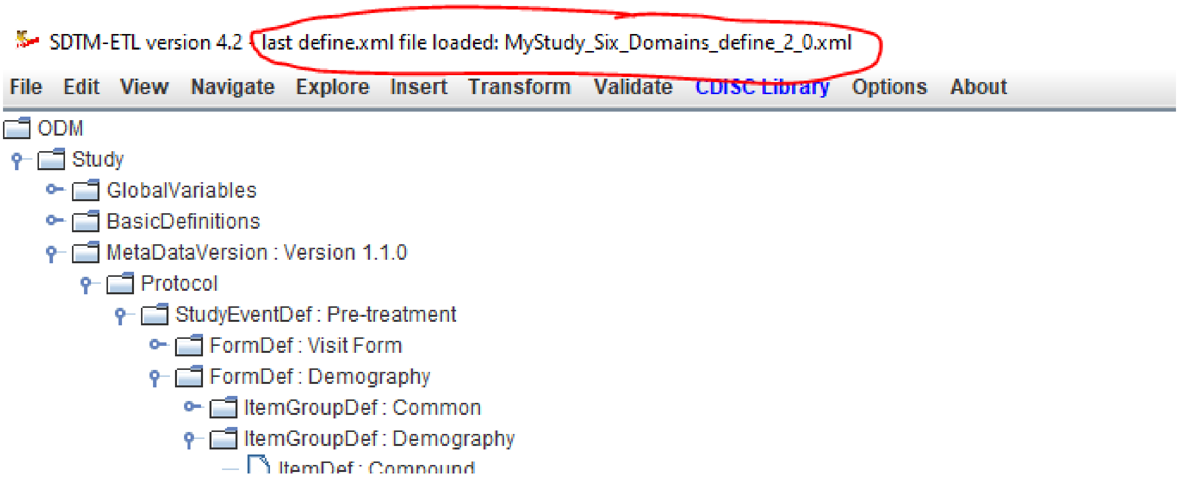
We also removed the property to indicate where the Pinnacle21 validation software is located from the "properties.dat file", as it is unclear whether starting up this software from another software is still allowed by the Pinnacle21 license. Furthermore, we strongly believe there is no place in future anymore for Pinnacle21 Community in the world of validation. CDISC CORE is certainly the future, for which we now have our own implementation. Interfacing with validation software of other vendors (that implement CORE) is of course still possible. Just ask us.

Support for SENDIG v.3.1.1

Templates for SENDIG v.3.1.1 have now been added, meaning that when starting a new project, SENDIG-3.1.1 can be used right from the start.

Display of last define.xml loaded

Often, it is advantageous to work on one, or group of, SDTM/SEND domains only. In such a case, especially when then still merging with define.xml-s for other domains, one may loose oversight. Therefore, we added a new feature that shows what define.xml was last loaded, which is displayed on the title of the main window. For example:



SDTM-ETL version 4.2 last define.xml file loaded: MyStudy_Six_Domains_define_2_0.xml

New CDISC-CORE Validation Engine v.0.6.3

SDTM-ETL 4.3 comes with the new CDISC CORE (CDISC Open Rules Engine) validation software which is considerably superior to Pinnacle21. Version 0.6.3 of CORE was published on October 12, 2023. Customers who acquired SDTM-ETL 4.3 before that date, or upgraded to it, automatically become a free patch update available.

The new CORE version has considerably more rules implemented, as well CDISC as FDA rules, and must be regarded as the "reference implementation" of the CDISC and FDA rules. First efforts are now already starting to also implement all the PMDA rules.

For further details regarding CORE v.0.6.3, please see the release website.

CLI / Batch execution: new parameters

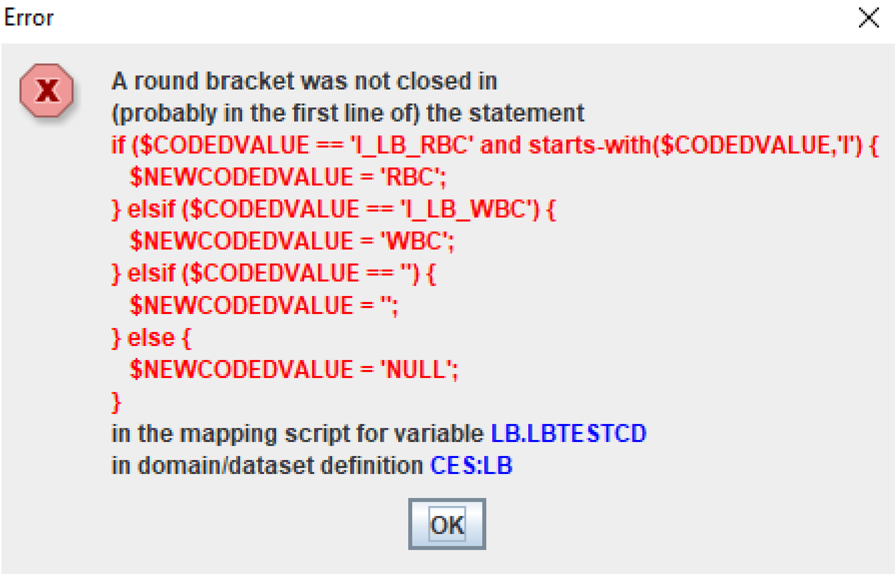
New parameters for batch / CLI execution have been added:

Parameter / keyword	Parameter value	Explanation
-POSTPROCESSINGFORLOBXFL	none	keyword to indicate that --LOBXFL values must be generated using postprocessing for Findings datasets. Requires that the corresponding --LOBXFL variable is define in the define.xml that drives the dataset generation.
-GENERATEMERGEDDATASETSSPLIT DOMAINS	none	keyword to indicate that in case that there are "split domain" datasets to be generated (e.g. LBUR, LBCH, LBHE), also a "merged" dataset (and possible the corresponding merged SUPP-- dataset) needs to be generated. This dataset than gets the name of the domain, extended by "_merg" (e.g. LB_merg.xpt)

Bug fixes

- In the AP (Associated Persons) domain, there is no USUBJID variable, causing that the APSEQ value in the post-processing step was not correctly calculated. This has been fixed. The calculation of the APSEQ value is now based on the value of APID.
- When having assigned an "Origin" to a "Non-Standard Variable" (NSV), and using the option "Move non-standard Variables to SUPP--", QORIG was not populated with the value of "Origin" in the case of Define-XML 2.0 and 2.1 (it was in the case of 1.0). This has been fixed.
  Remark: QORIG is essentially a design failure: the origin is metadata, not data, QORIG should never have been added to SDTM.
- A problem with referencing ValueList-s for NSV (non-standard variables as "supplemental qualifiers") in the define.xml has been fixed.
- When clicking the checkbox "Remove SDTM variables that do not have a mapping" was checked, but no such variables were present, an empty list was presented. This has been fixed.

- When in an if-elsif-else construct, the user forgot to "close" a round bracket, this caused an error in the transformation to XSLT, and resulted in a message "A serious error has occurred", leaving the user with no idea what was happening or where to start looking for an error in one of the mapping scripts.
  This has now been fixed: the user is informed in which mapping script the problem is present. For example:



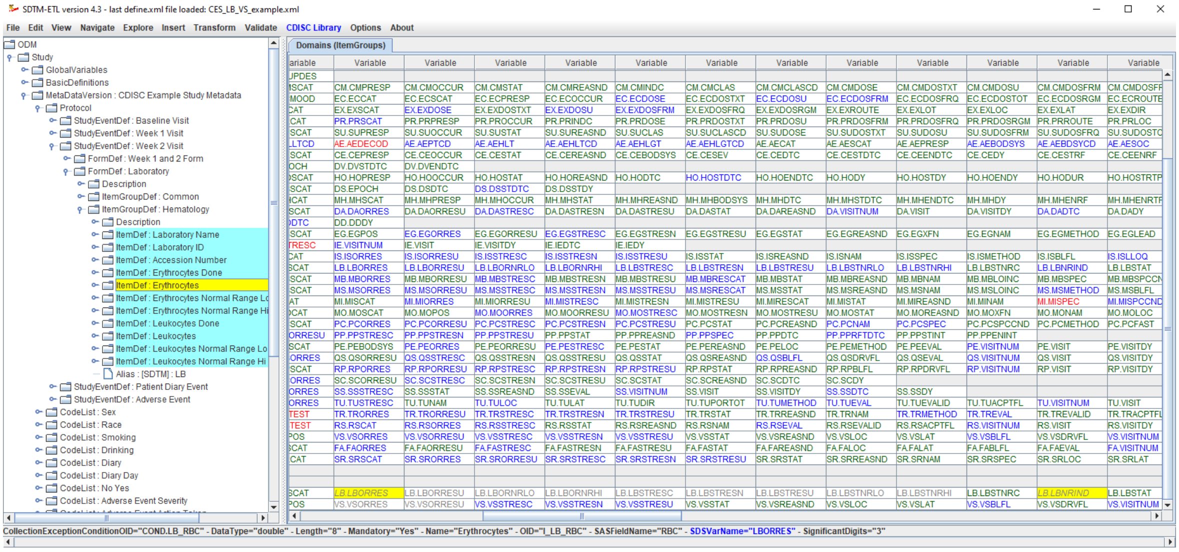
Error

A round bracket was not closed in
(probably in the first line of) the statement

```
if ($CODEDVALUE == 'I_LB_RBC' and starts-with($CODEDVALUE,'I') {
   $NEWCODEDVALUE = 'RBC';
} elsif ($CODEDVALUE == 'I_LB_WBC') {
   $NEWCODEDVALUE = 'WBC';
} elsif ($CODEDVALUE == '') {
   $NEWCODEDVALUE = '';
} else {
   $NEWCODEDVALUE = 'NULL';
}
```

in the mapping script for variable LB.LBTESTCD
in domain/dataset definition CES:LB

- When clicking on an item in the ODM tree, the cells in the SDTM table in which that item is used in the mapping is highlighted. Under some circumstances, only 1 of the SDTM cells was highlighted, even when the ODM item is used in several mappings.
  This has now been fixed.
  For example:



SDTM-ETL version 4.3 - last define.xml file loaded: CES_LB_VS_example.xml

CollectionExceptionConditionOID="COND.LB_RBC" - DataType="double" - Length="8" - Mandatory="Yes" - Name="Erythrocytes" - OID="I_LB_RBC" - SASFieldName="RBC" - SDSVarName="LBORRES" - SignificantDigits="3"

showing that the value for the "erythrocytes concentation" is used in both the mappings for LBORRES (Result or Finding in Original Units) as for the calculation of LBNRIND (Reference Range Indicator).