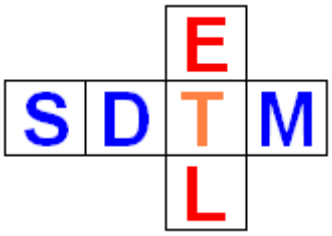
SDTM-ETL 4.2: Summary of New Features

Author: Jozef Aerts, XML4Pharma

Last update: **2023-04-20**



Summary

This document contains a summary of the most important new features of SDTM-ETL 4.2
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 4.2.

Table of Contents

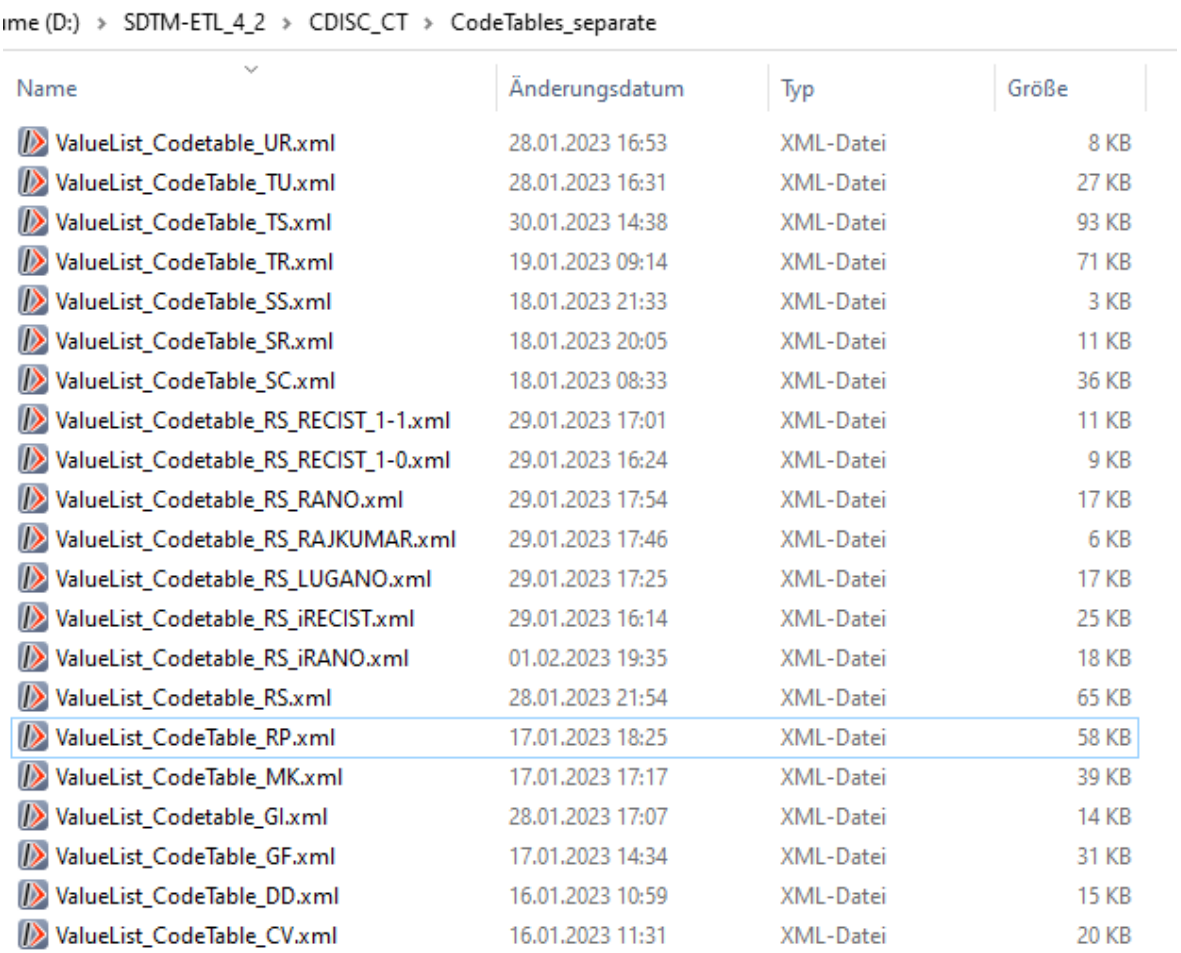
Implementation of CDISC "Code Tables" as ValueLists

Due to the "hypervertical" structures of especially the Findings domains, there are many dependencies between variable values of especially --ORRES, --ORRESU, --STRESC and --STRESU, meaning that the properties of e.g. RPSTRESC (Character Result/Finding in Standard Format) and of RPSTRESU (Unit) in the RP (Reproductive Systems Findings) depends on the value of RPTESTCD (Test Code). From the by CDISC published Excel file:

	A	B	C	D	E	F	G	H	I	J	K	L
1	C-code (Concept Code)	Reproductive System Findings Test Code (RPTESTCD) (codelist code =C106479)	Reproductive System Findings Test Name (RPTEST) (codelist code = C106478)		C-code (Concept Code)	No Yes Response (NY) (codelist code = C66742)		C-code (Concept Code)	Mode of Delivery (MODDLV) (codelist code = C181165)		C-code (Concept Code)	Unit of Measure (UNIT) (codelist code = C71
59	C73435	GRAVIND	Gravida Indicator		C49488	Y						
60	C156588	INABOIND	Induced Abortion Indicator		C49487	N						
61	C156588	INABOIND	Induced Abortion Indicator		C17998	U						
62	C156588	INABOIND	Induced Abortion Indicator		C49488	Y						
63	C120835	INABORTN	Number of Induced Abortions									
64	C41255	INTP	Interpretation									
65	C81257	LMPSTDTC	Last Menstrual Period Start Date									
66	C154886	MASSIND	Mass Indicator		C49487	N						
67	C154886	MASSIND	Mass Indicator		C17998	U						
68	C154886	MASSIND	Mass Indicator		C49488	Y						
69	C19666	MENARAGE	Menarche Age									
70	C156586	MENFDUR	Menses Flow Duration								C25301	day
71	C156585	MENFRD	Menses Flow Rate Description									
72	C106497	MENOAGE	Menopause Age									
73	C106541	MENOSTAT	Menopause Status									
74	C106541	MENOSTAT	Menopause Status									
75	C106541	MENOSTAT	Menopause Status									
76	C156587	MENREG	Menstrual Cycle Regularity									
77	C119549	MENSDUR	Menstrual Cycle Duration								C25301	day
78	C122191	MSCRGIND	Miscarriage Indicator		C49487	N						
79	C122191	MSCRGIND	Miscarriage Indicator		C17998	U						
80	C122191	MSCRGIND	Miscarriage Indicator		C49488	Y						
81	C154887	MTENDIND	Motion Tenderness Indicator		C49487	N						
82	C154887	MTENDIND	Motion Tenderness Indicator		C17998	U						

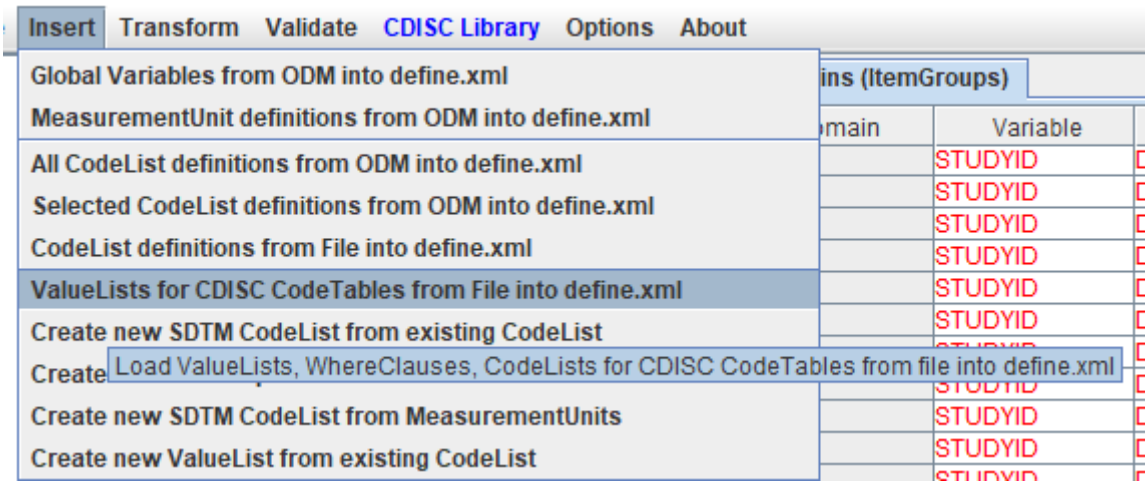
However, for actual mapping work, such Excel files are unusable, and the mapper would need to copy-paste information from them into the wizards for setting up "ValueList"s in the underlying define.xml, which can easily mean hours of work and is error prone.

Therefore, we made most of the "code tables" from CDISC available as define.xml "snippets" which can then be imported in any define.xml structure during the mapping.
These "ValueList" define.xml files are located in the directory "CDISC_CT/CodeTables_separate":

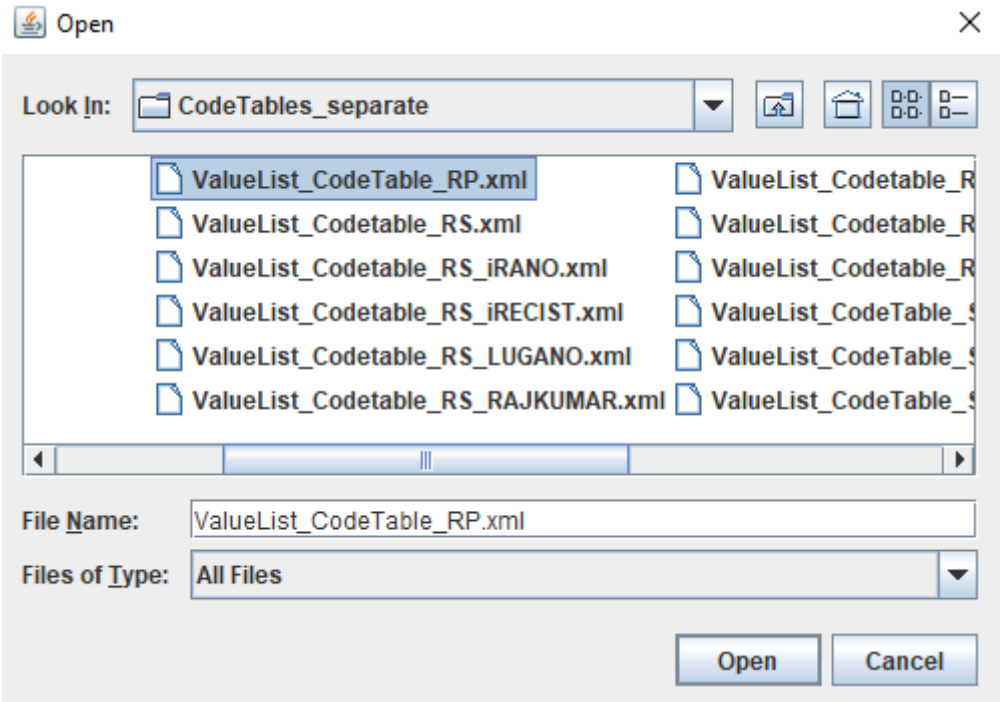


Code tables that have not been implemented (yet) are the code tables for IS (Immunogenicity Specimen Assessments) with their dependencies between LB (Laboratory) and MB (Microbiology) domains and the IS domain. Also the SEND code tables have not been implemented as they seem to have been retreated by the SEND team.
Remark that all these "ValueList" define.xml files come as Define-XML 2.0 format. However, when using define.xml 2.1, they are automatically transformed to v.2.1 before being imported.

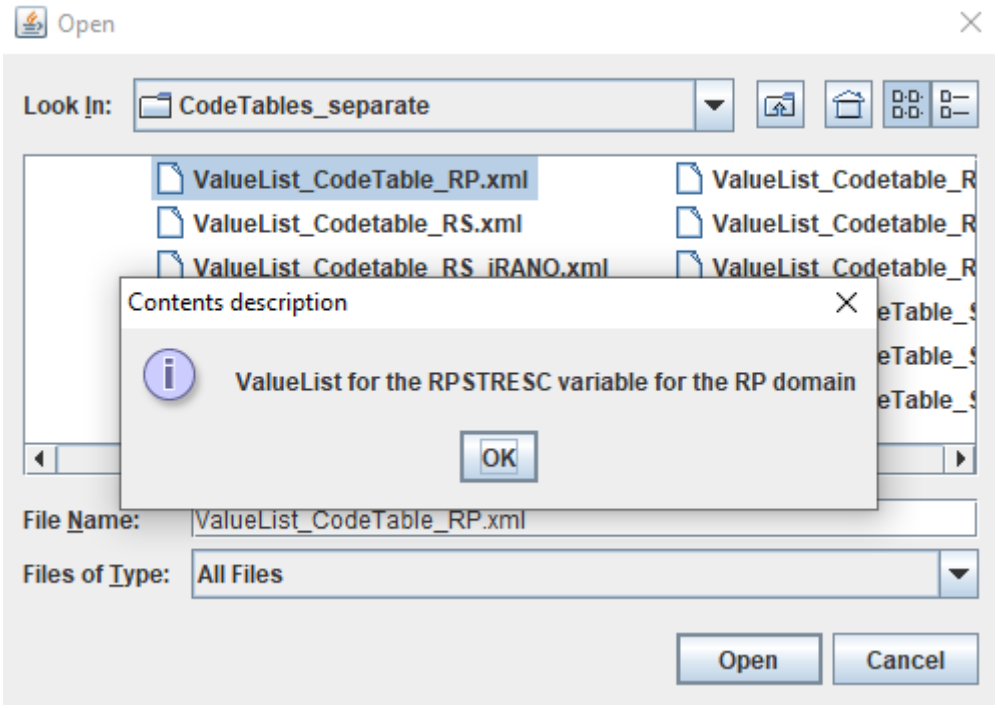
In order to import one of these ValueLists, use the new menu "Insert - ValueLists for CDISC CodeTables from File into define.xml":



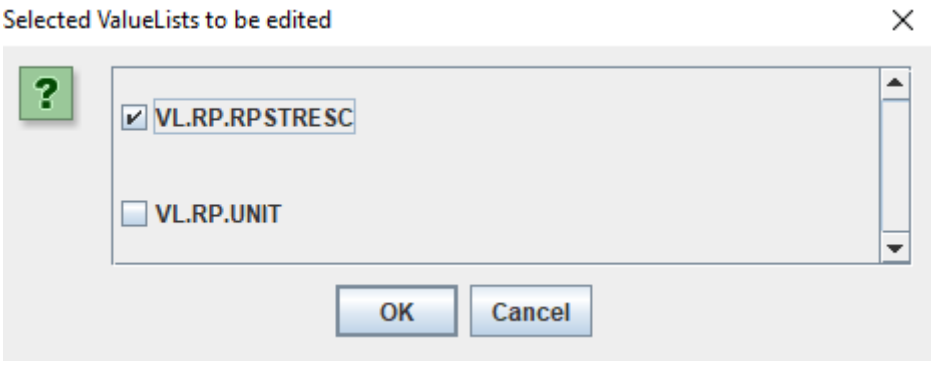
One is then prompted to select a file from the "CDISC_CT/CodeTables_separate" directory. For example, for RP (Reproductive Systems Findings):



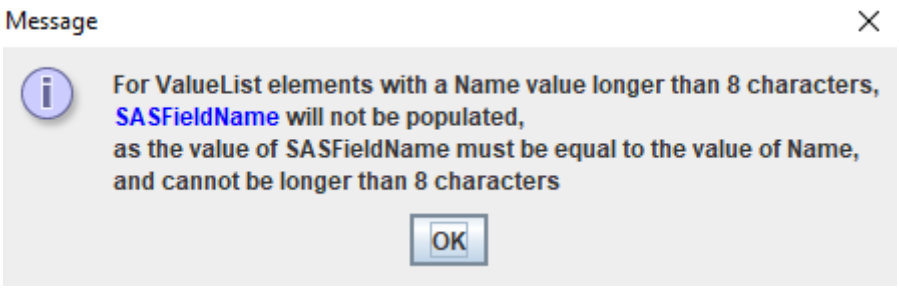
In order to help selecting the right one, when one clicks on a file name, some explanation is provided, e.g.:



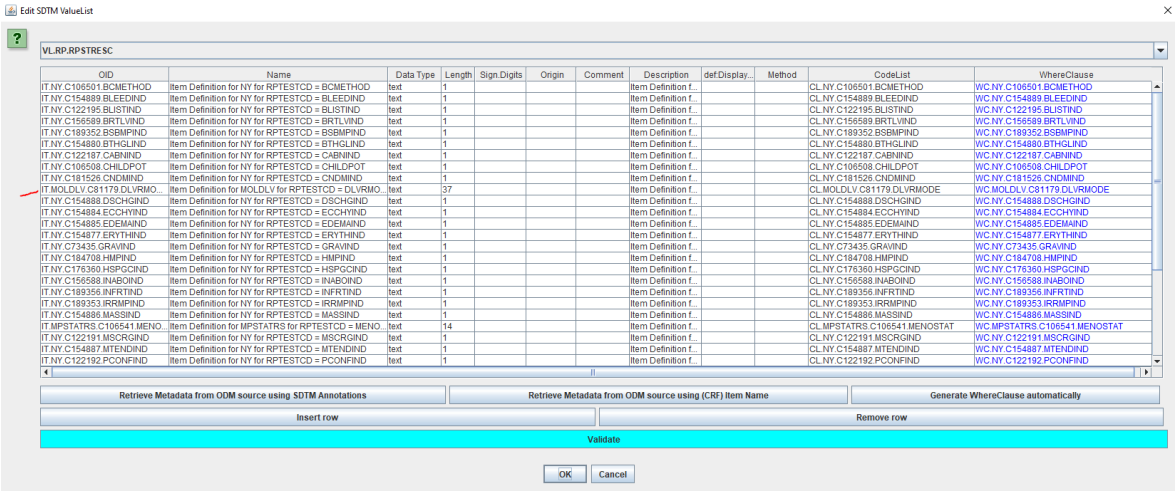
Then clicking "Open" imports the "ValueList define.xml" and inserts it into the existing define.xml. Immediately after this, the user is given the opportunity to still edit (or also only inspect) the imported value lists:



which is then opened with the "ValueList Editor" (in the case of complicated ValueLists this can take some time) after a message is shown:



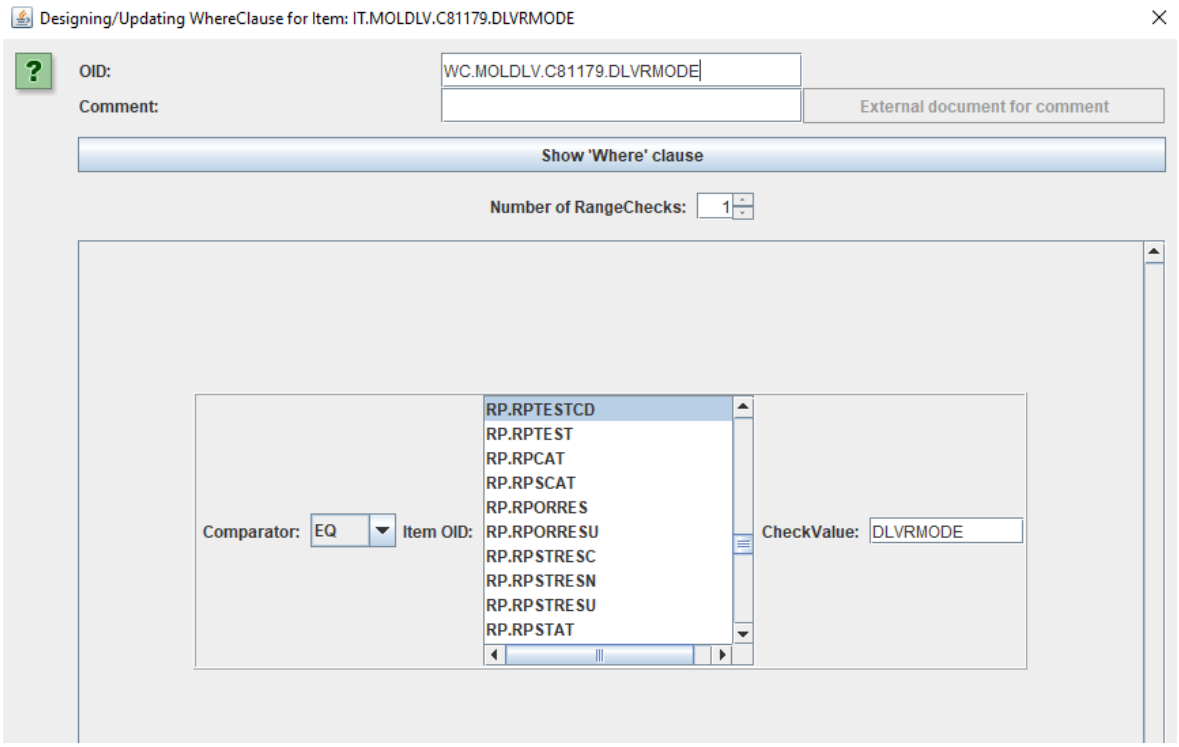
(Remark that "SASFieldName" is another relict of the outdated SAS Transport 5 format). Followed by the editor itself:



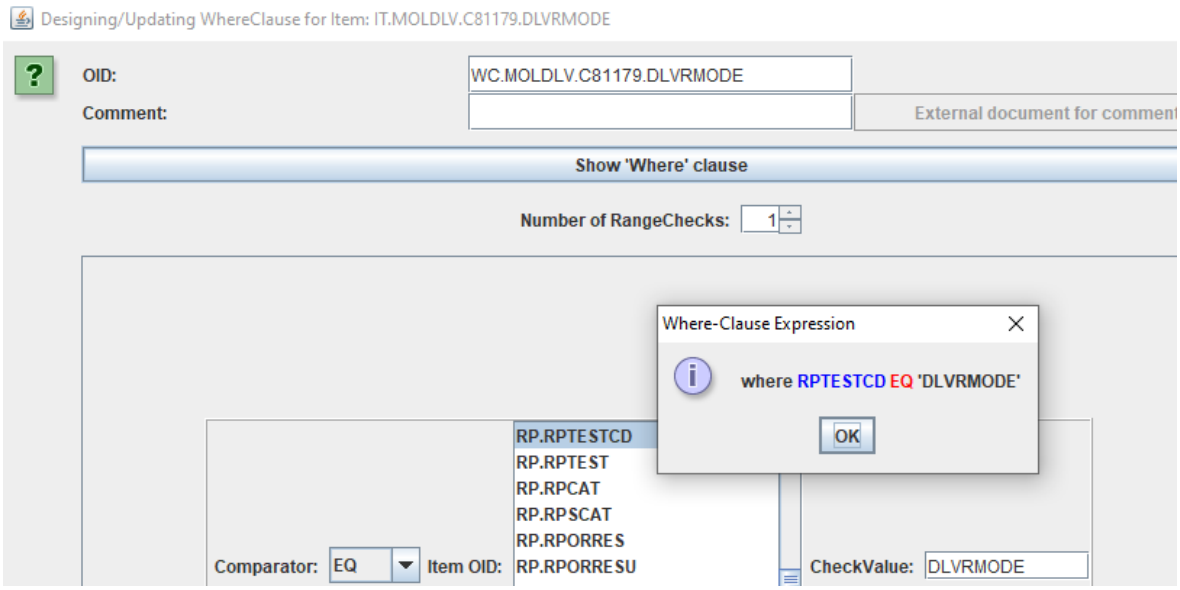
One sees e.g. that for the first 9 "use cases", a "Yes-No" codelist is associated, i.e. the value for RPSTRESC can only be "Y" (Yes) or "N" (No). For the use case that RPESTCD = DLVRMODE (Mode of Delivery) the associated codelist for RPSTRESC is the CDISC "C81179.DLVRMODE" codelist with values:

```
<CodeList OID="CL.MOLDLV.C81179.DLVRMODE" Name="CodeList for MOLDLV for the case that RPTESTCD = DLVRMODE" DataType="text">
    <Description>
        <TranslatedText xml:lang="en">CodeList for MOLDLV for the case that RPTESTCD = DLVRMODE</TranslatedText>
    </Description>
    <EnumeratedItem CodedValue="BREECH EXTRACTION">
        <Alias Context="nci:ExtCodeID" Name="C114136"/>
    </EnumeratedItem>
    <EnumeratedItem CodedValue="DELIVERY BY DESTRUCTIVE OPERATION">
        <Alias Context="nci:ExtCodeID" Name="C114143"/>
    </EnumeratedItem>
    <EnumeratedItem CodedValue="CESAREAN SECTION">
        <Alias Context="nci:ExtCodeID" Name="C46088"/>
    </EnumeratedItem>
    <EnumeratedItem CodedValue="VAGINAL ASSISTED DELIVERY">
        <Alias Context="nci:ExtCodeID" Name="C81301"/>
    </EnumeratedItem>
    <EnumeratedItem CodedValue="VAGINAL BIRTH AFTER CESAREAN DELIVERY">
        <Alias Context="nci:ExtCodeID" Name="C81302"/>
    </EnumeratedItem>
    <EnumeratedItem CodedValue="VAGINAL DELIVERY">
        <Alias Context="nci:ExtCodeID" Name="C81303"/>
    </EnumeratedItem>
    <Alias Context="nci:ExtCodeID" Name="C181165"/>
</CodeList>
```

Also the "where clause" can now be inspeced (or even edited) by clicking on it. For example for "Delivery Mode", clicking on "WC.MOLDLV.C81179.DLVRMODE" opens the wizard for that "where clause":

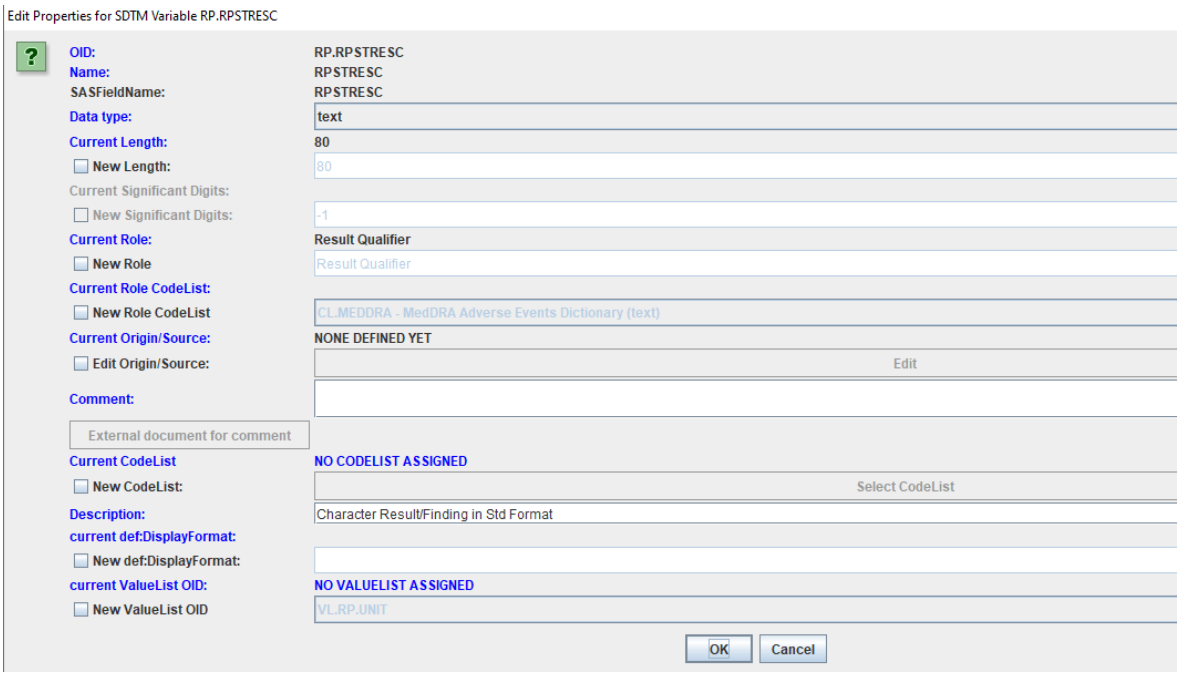


and clicking the "Show 'Where' clause" provides a human-understandable description of it:

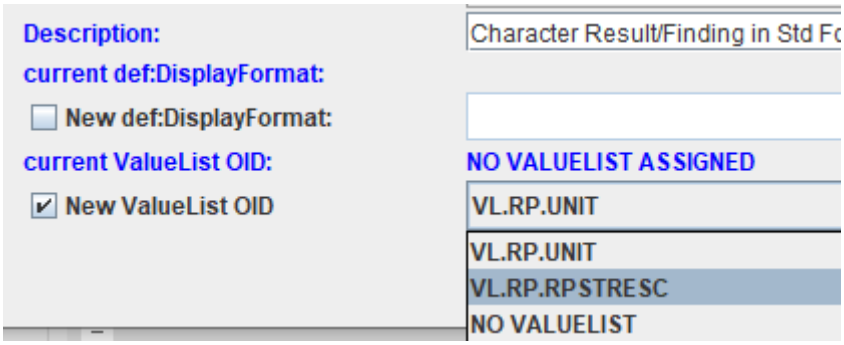


After having imported the "ValueList define.xml", one can assign it to the corresponding variable.

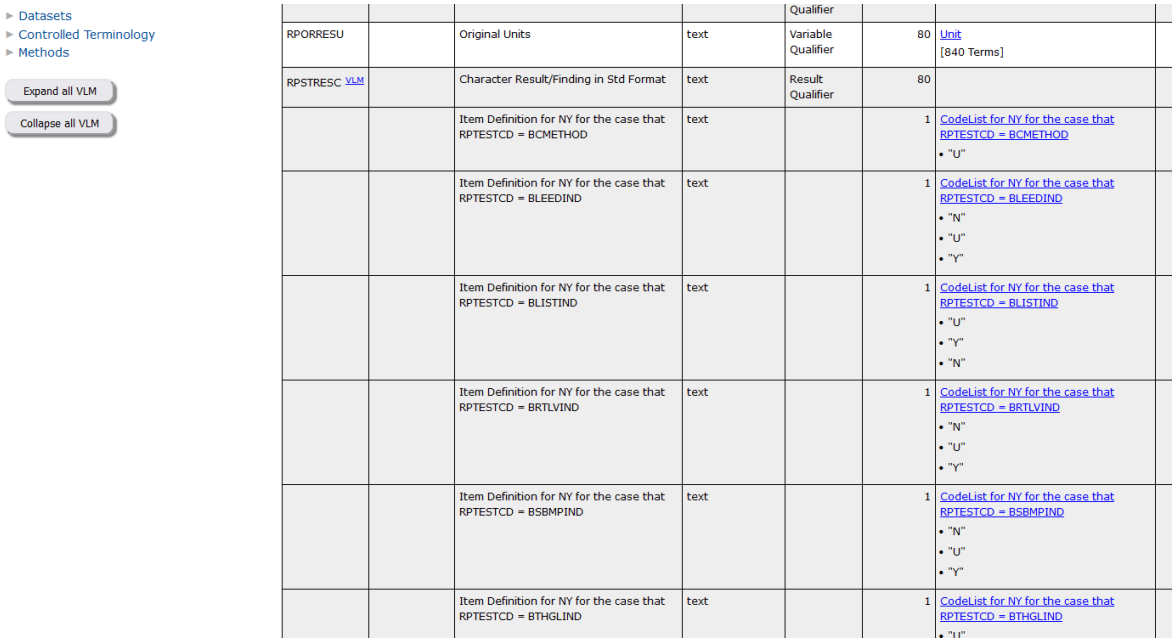
For RPSTRESC by selecting the cell, and then use the menu "Edit - SDTM Variable Properties", leading to:



and assigning the ValueList to it by selecting the "New ValueListOID" (near the bottom):



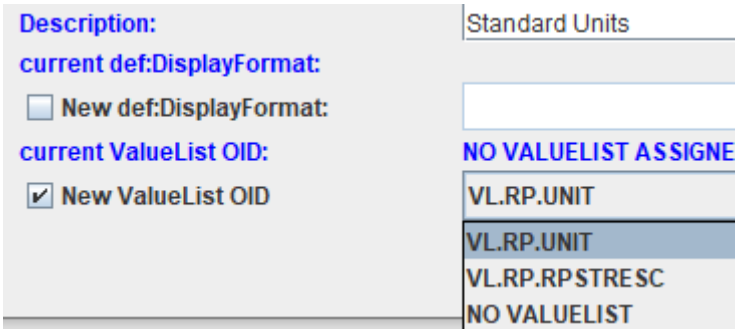
After clicking "OK", the valuelist VL.RP.RPSTRESC is then assigned to the SDTM variable RPSTRESC. This can then also be seen in the "view" by using the menu "View - View define.xml in browser":



				Qualifier		
RPORRESU		Original Units	text	Variable Qualifier	80	Unit [840 Terms]
RPSTRESC VLM		Character Result/Finding in Std Format	text	Result Qualifier	80	
		Item Definition for NY for the case that RPTESTCD = BCMETHOD	text		1	CodeList for NY for the case that RPTESTCD = BCMETHOD • "U"
		Item Definition for NY for the case that RPTESTCD = BLEEDIND	text		1	CodeList for NY for the case that RPTESTCD = BLEEDIND • "N" • "U" • "Y"
		Item Definition for NY for the case that RPTESTCD = BLISTIND	text		1	CodeList for NY for the case that RPTESTCD = BLISTIND • "U" • "Y" • "N"
		Item Definition for NY for the case that RPTESTCD = BRTLVIND	text		1	CodeList for NY for the case that RPTESTCD = BRTLVIND • "N" • "U" • "Y"
		Item Definition for NY for the case that RPTESTCD = BSBMPIND	text		1	CodeList for NY for the case that RPTESTCD = BSBMPIND • "N" • "U" • "Y"
		Item Definition for NY for the case that RPTESTCD = BTHGLIND	text		1	CodeList for NY for the case that RPTESTCD = BTHGLIND • "U"

Of course, one can then still always edit the valuelist, although this will only be necessary in seldom cases.

Similar for RPSTRESU:

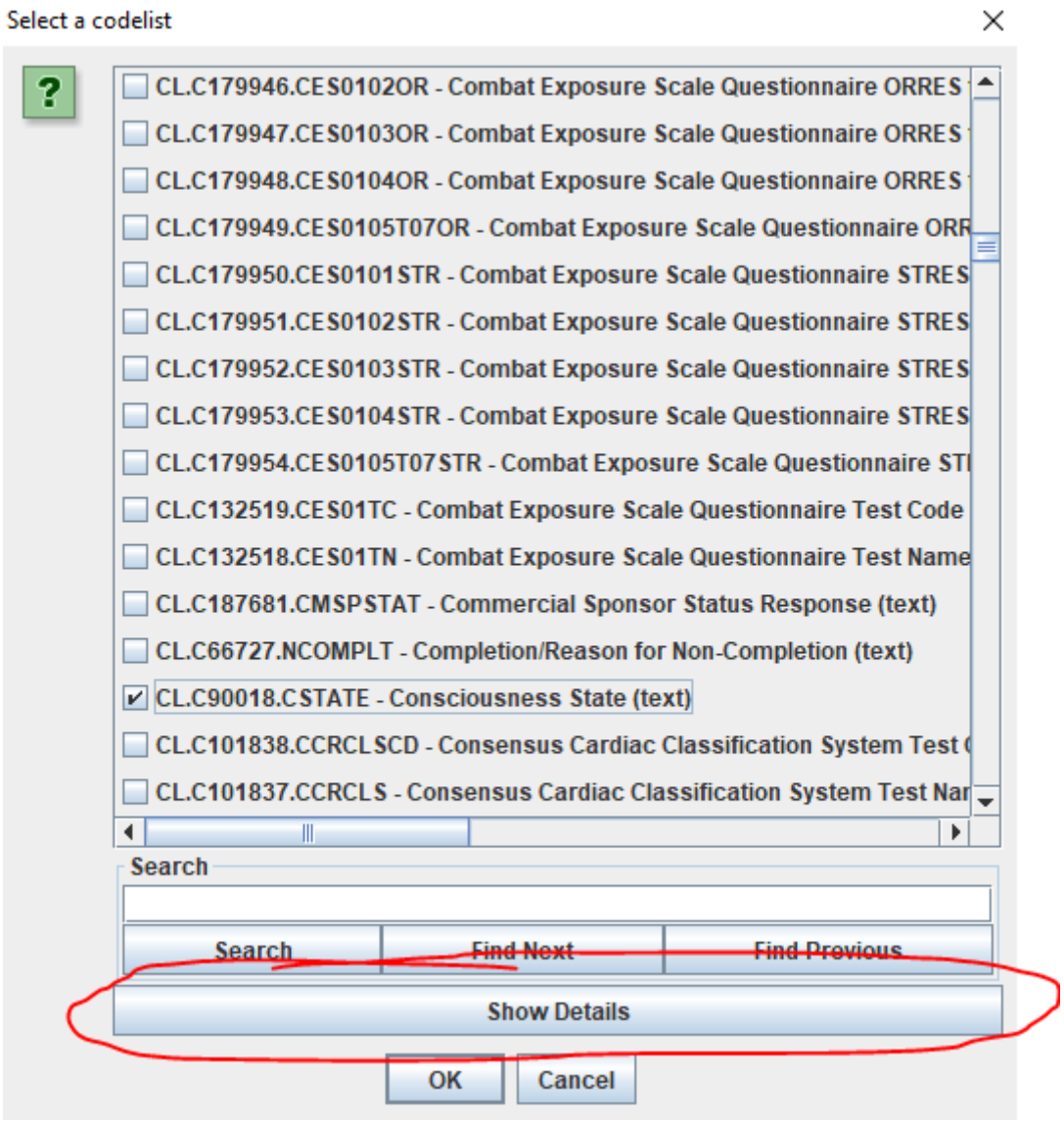


leading to (in the view):

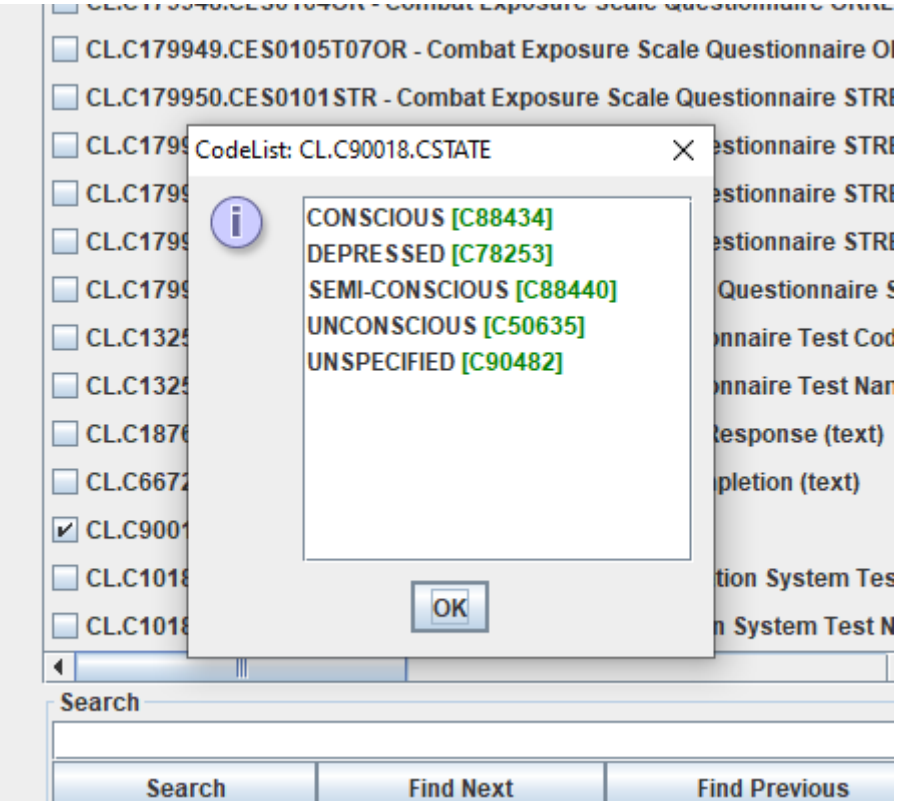
RPSTRESN		Numeric Result/Finding in Standard Units	integer	Result Qualifier	80		
RPSTRESU VLM		Standard Units	text	Variable Qualifier	80	Unit [840 Terms]	
	RPTESTCD = "ADRNRAGE" (Adrenarche Age)	Item Definition for AGEU for the case that RPTESTCD = ADRNRAGE	text		5	CodeList for AGEU for the case that RPTESTCD = ADRNRAGE • "YEARS"	
	RPTESTCD = "EGESTAGE" (Estimated Gestational Age)	Item Definition for AGEU for the case that RPTESTCD = EGESTAGE	text		5	CodeList for AGEU for the case that RPTESTCD = EGESTAGE • "WEEKS" • "DAYS"	
	RPTESTCD = "FUNDHT" (Fundal Height)	Item Definition for UNIT for the case that RPTESTCD = FUNDHT	text		2	CodeList for UNIT for the case that RPTESTCD = FUNDHT • "mm" • "cm" • "in"	
	RPTESTCD = "MENOAGE" (Menopause Age)	Item Definition for AGEU for the case that RPTESTCD = MENOAGE	text		5	CodeList for AGEU for the case that RPTESTCD = MENOAGE • "YEARS"	
	RPTESTCD = "MENARAGE" (Menarche Age)	Item Definition for AGEU for the case that RPTESTCD = MENARAGE	text		5	CodeList for AGEU for the case that RPTESTCD = MENARAGE • "YEARS"	
	RPTESTCD = "MENFDUR" (Menses Flow Duration)	Item Definition for UNIT for the case that RPTESTCD = MENFDUR	text		3	CodeList for UNIT for the case that RPTESTCD = MENFDUR • "day"	
	RPTESTCD = "MENSDUR" (Menstrual Cycle Duration)	Item Definition for UNIT for the case that RPTESTCD = MENSDUR	text		3	CodeList for UNIT for the case that RPTESTCD = MENSDUR • "day"	
	RPTESTCD = "THICK" (Thickness)	Item Definition for UNIT for the case that RPTESTCD = THICK	text		2	CodeList for UNIT for the case that RPTESTCD = THICK • "mm" • "cm" • "in"	

CodeList selection and editing: new "Show CodeList Details" button

When assigning a codelist to an SDTM variable, it is not always immediately obvious which one to choose from the large list available from CDISC or added or subsetted. So one often wants to have a "preview" of what is in the codelist. In order to make this easier, a new button "Show Details" has been added to the "CodeList Selection" wizard:



Clicking the "Show Details" button for the selected codelist (in this case "CSTATE") opens a new window with the details:



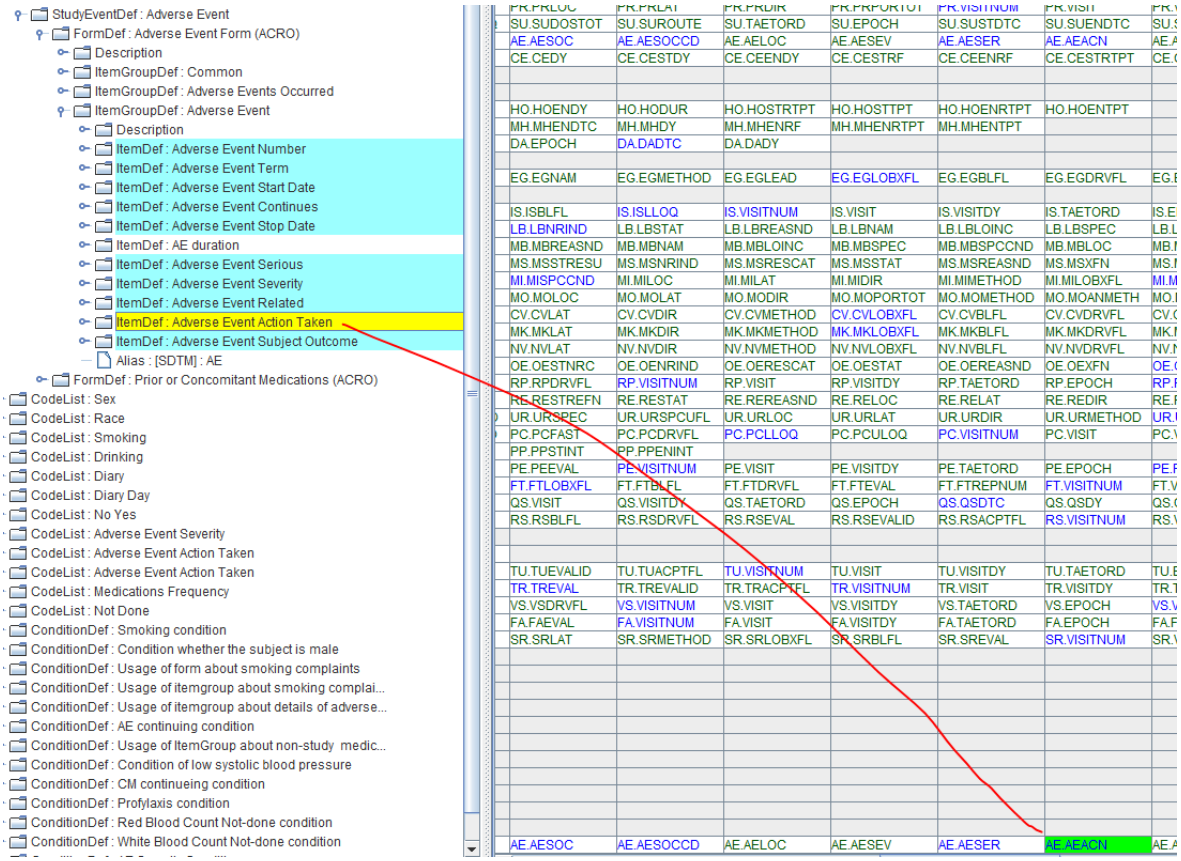
providing the term as well as the CDISC-NCI code (in square brackets).

Further improvements on the CodeList-CodeList mapper wizard

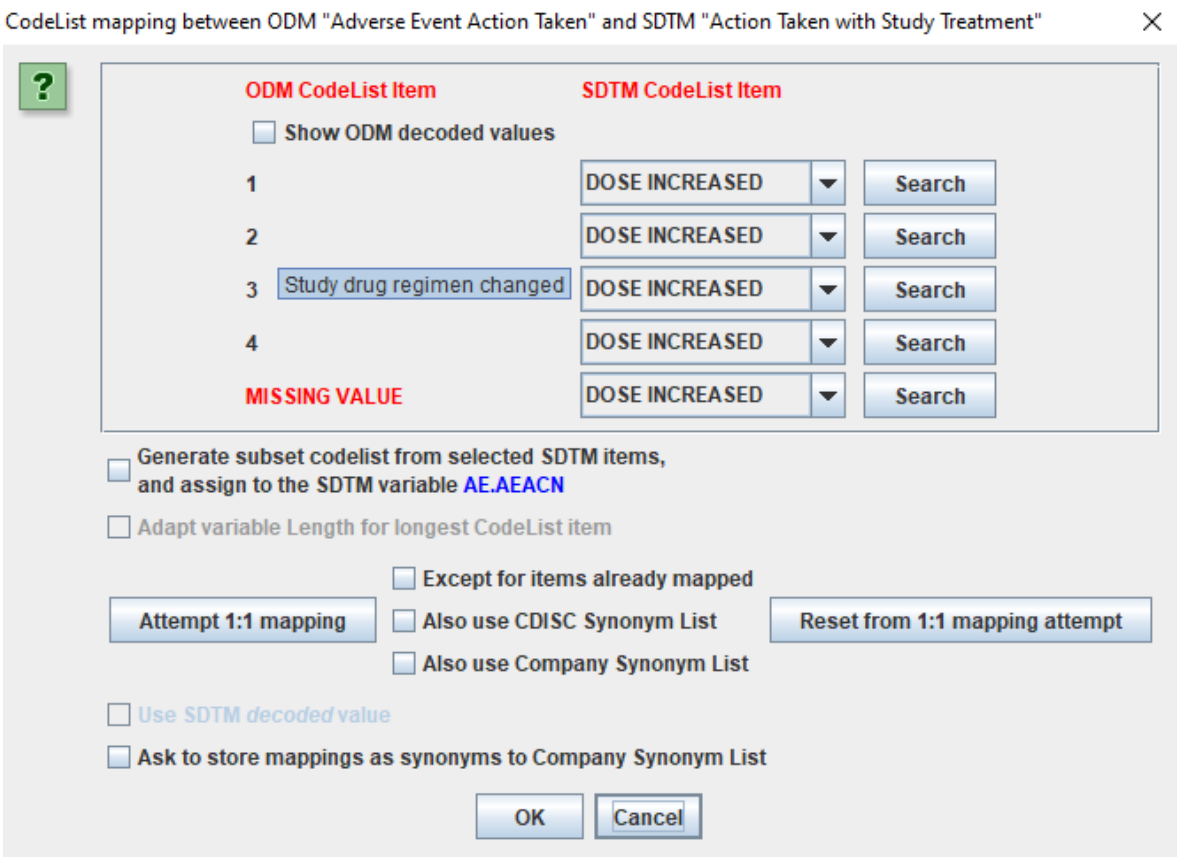
The CodeList-CodeList mapper wizard has always been one of the features mostly appreciated by our users: it saves hours and hours of otherwise tedious (and error prone) script writing work (as is required when e.g. using SAS for developing mappings).

One of our customers asked us to add some features making the wizard even more user-friendly.

When mapping an item "Adverse Event Action Taken" to AEACN by drag-and-drop:

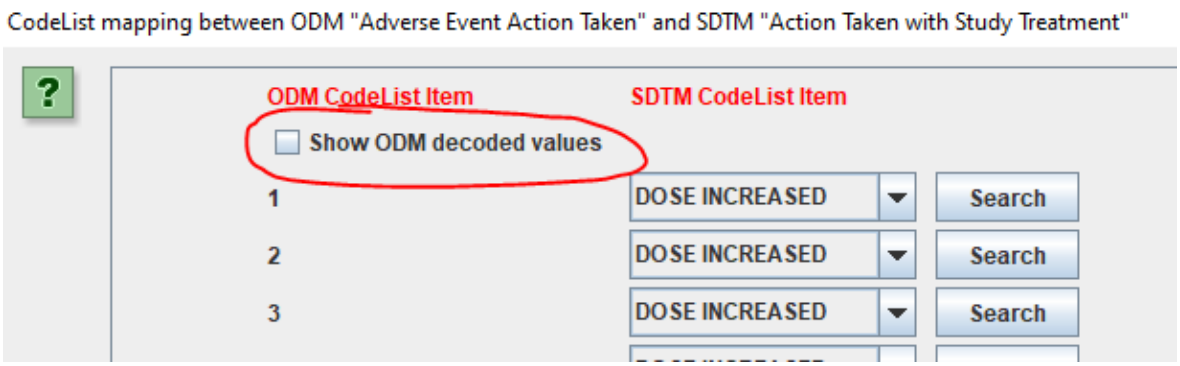


and choosing to use the CodeList-CodeList mapper wizard, one comes to:

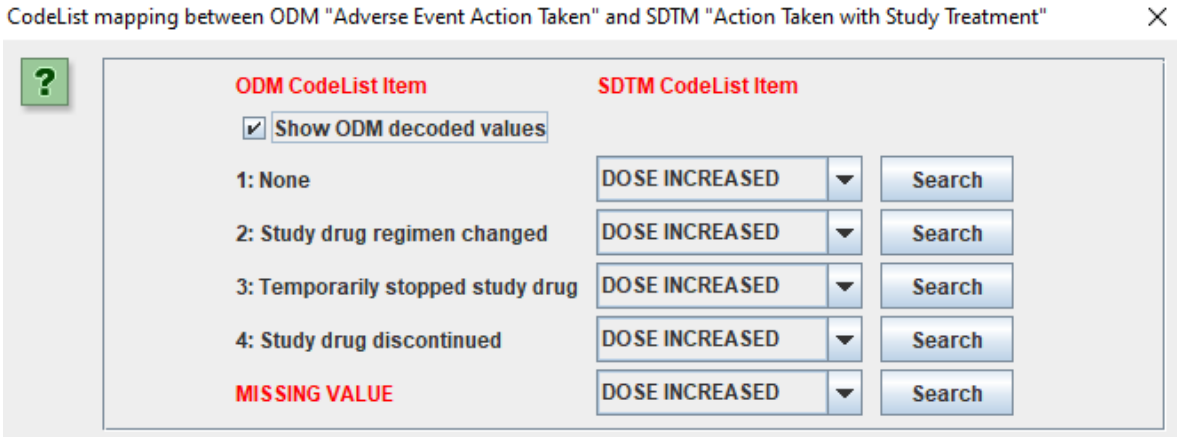


One sees that on the ODM side, numeric values 1-4 were used for the codes, which then must

be mapped to the SDTM codelist items like "DOSE INCREADED". Pointing the mouse over an item on the ODM side then shows the "decode", i.e. the meaning of the code. For example for "2", the decode is "Study drug regimen changed".
On request of a user, we now added the possibility to also show the "decode" directly on the label itself. For this, an additional checkbox "Show ODM decoded values" was added:

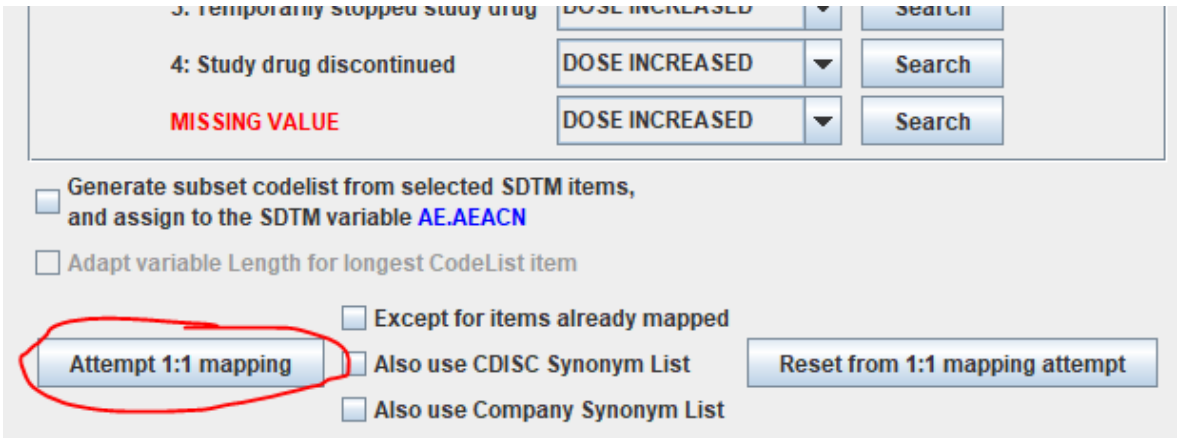


When it is checked, the view changes into:

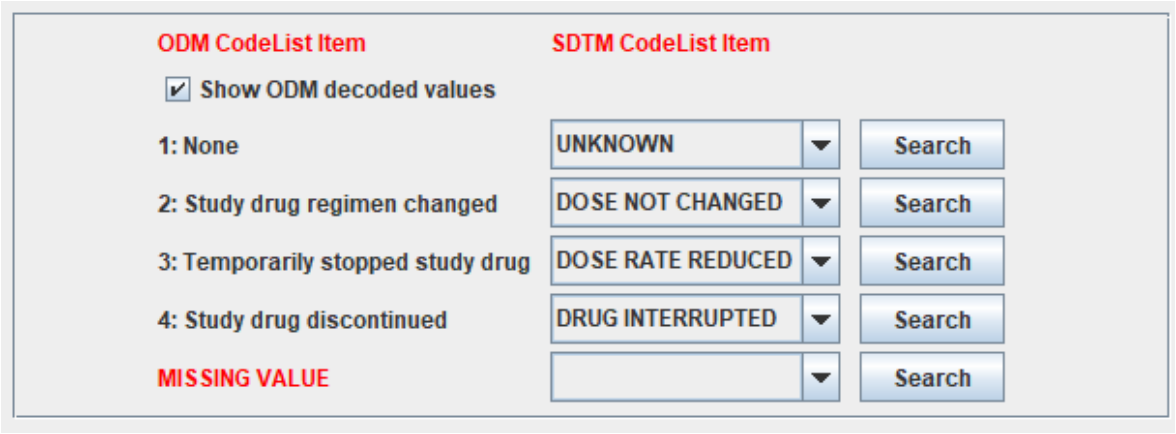


which makes it easier to make the necessary decisions.

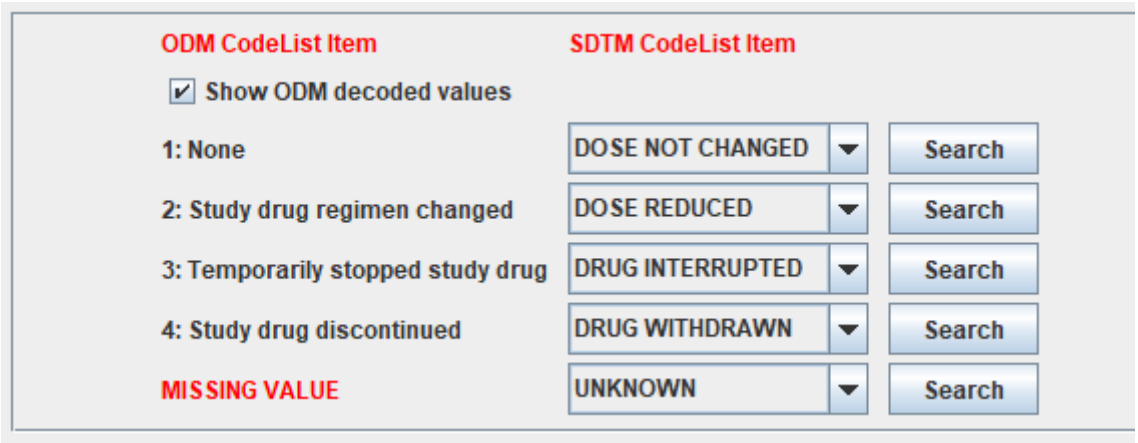
Another highly appreciated feature of the CodeList-CodeList mapper wizard is the button "Attempt 1:1 mapping", which, when clicked, makes proposals for a mapping based on word similarity.



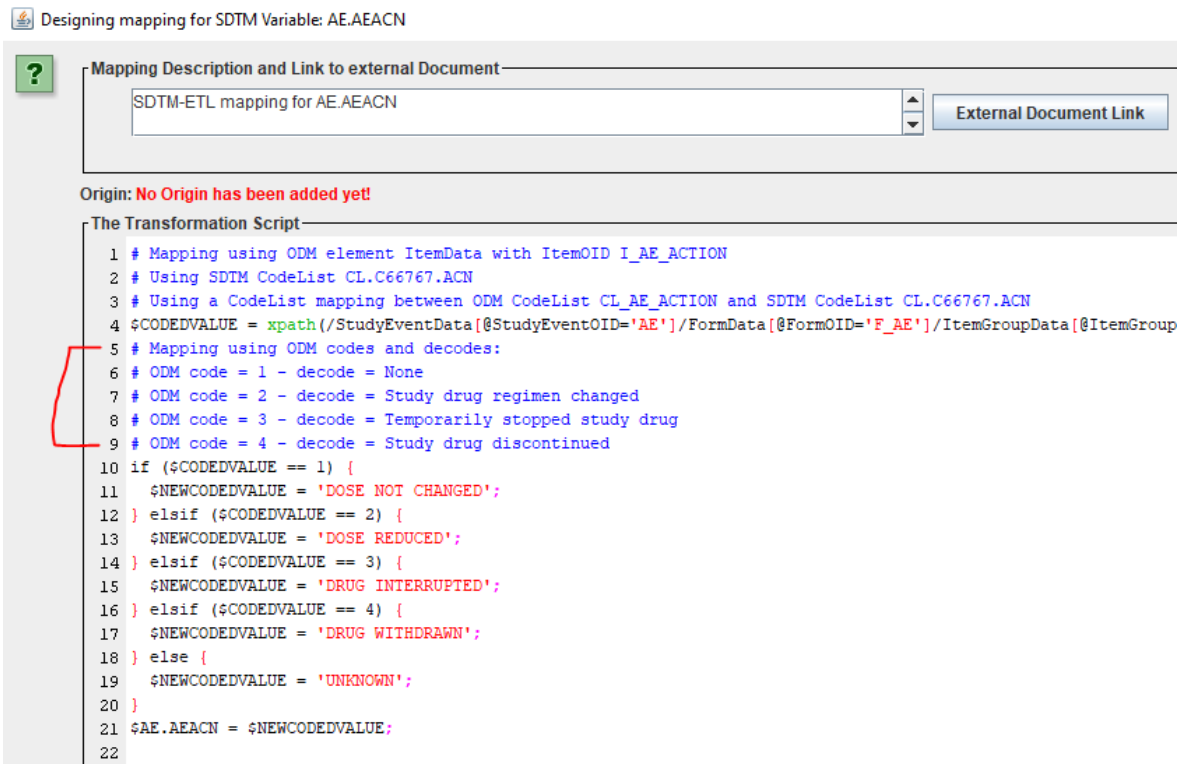
In our case, this leads to:



which is clearly wrong...
However, when one already had another study that used the same codes (on the ODM side), one would have stored the mapping in the "Company Synonym List", which would then lead to the correct mapping, e.g.:



After clicking "OK", the mapping script is automatically generated:



New is, that when the user asked for also displaying the "decoded values" in the wizard, additional comment lines are added to the mapping script, explaining what the codes are, and thus allowing to better understand the mapping script.

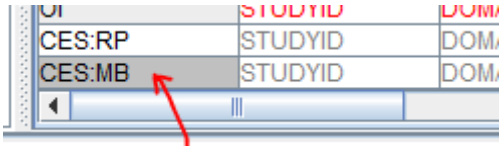
Sorting of SDTM/SEND records using the "Keys" in the underlying define.xml

On request of one of our customers, we also added the feature allowing to resort the generated records in the SDTM/SEND datasets according to the "keys" (def:KeySequence attribute) in the underlying define.xml.

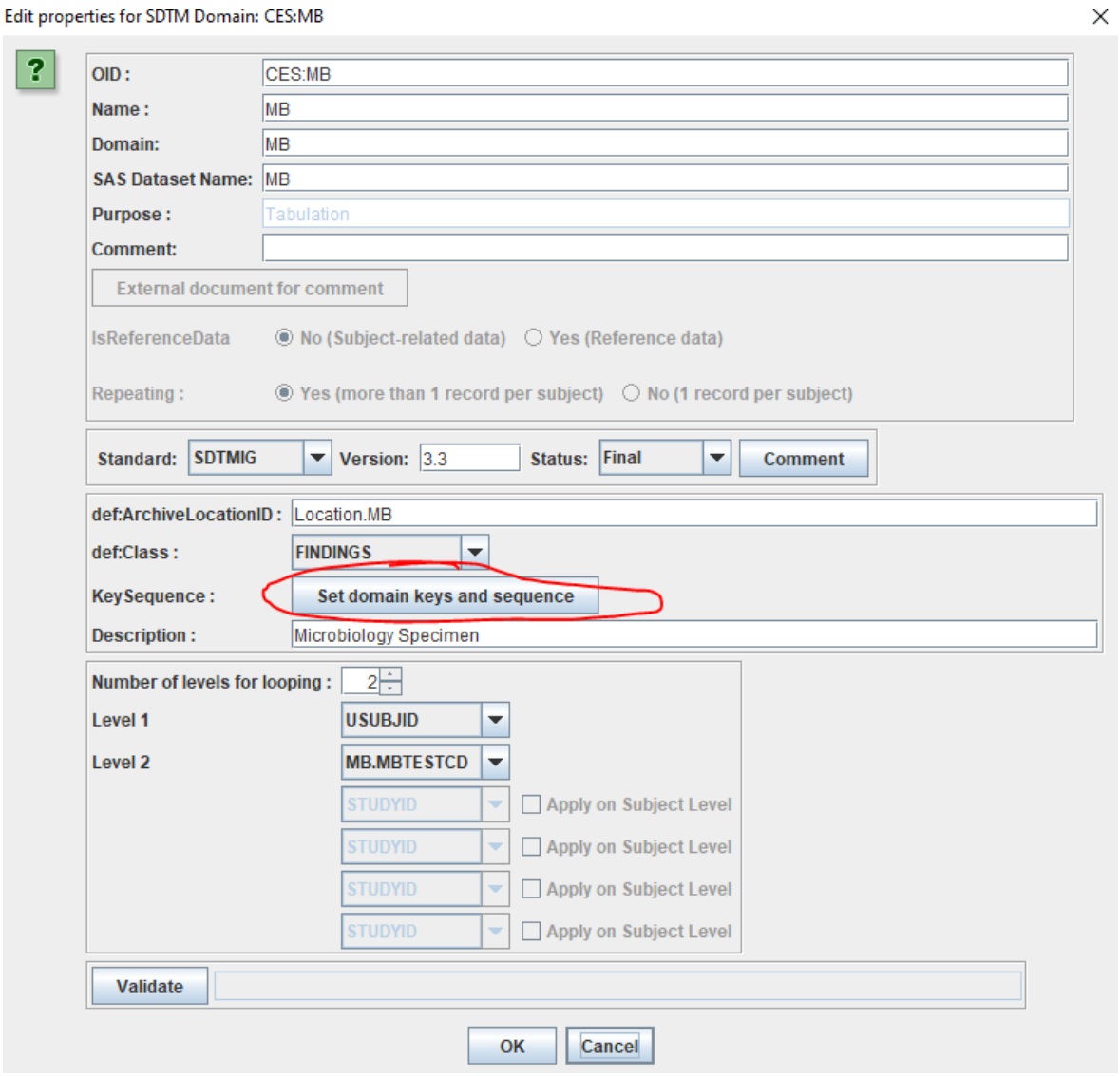
Essentially, the keys in the define.xml were never meant for sorting, they were introduced for providing the reviewer information about record uniqueness, i.e. the set of keys used will guarantee the uniqueness of the records.
Remark that also the combination of USUBJID and xxSEQ provides this, but one needs to take into account that xxSEQ is an "artificial" key, which needs to be generated in a post-processing step, whereas the keys defined by the "def:KeySequence" attribute are "natural keys".

Some users however also use the keys as defined in the define.xml for sorting the records in the datasets when these are generated. Therefore, we added the feature to enable this.

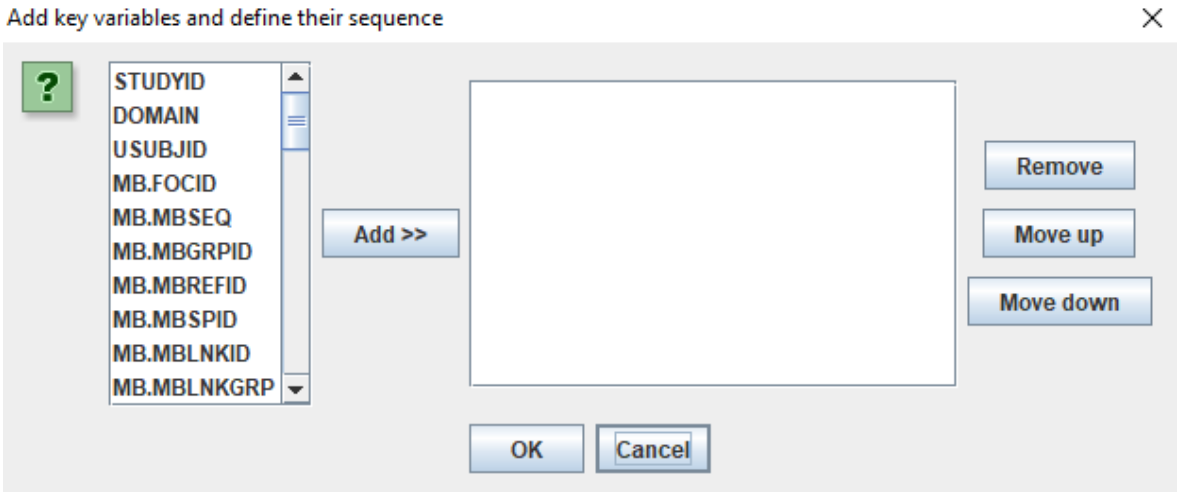
Keys can be added to a dataset by double-clicking the first cell in the row (designating the dataset). E.g. for MB:



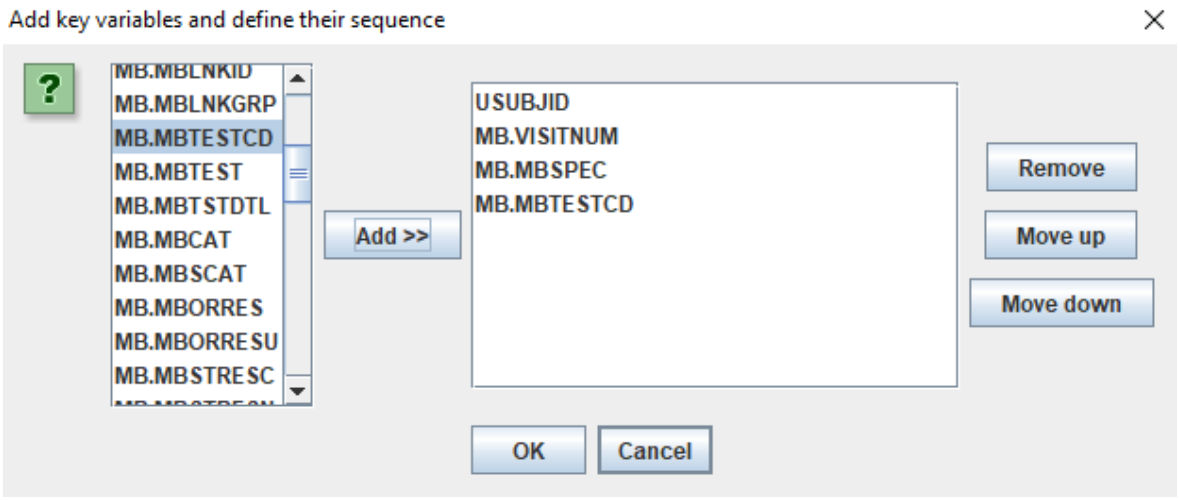
which leads to the editor dialog for the dataset:



One notices the button "Set domain keys and sequence". Clicking it e.g. leads to:

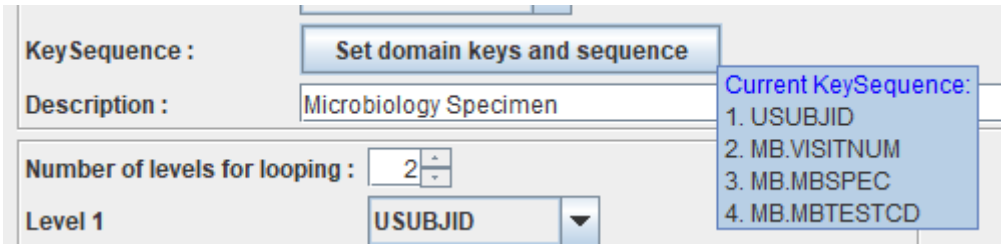


When one would now like to sort according to "One record per subject per visit per specimen per test", one adds the corresponding variables from the list on the left, using the button "Add". In our case:

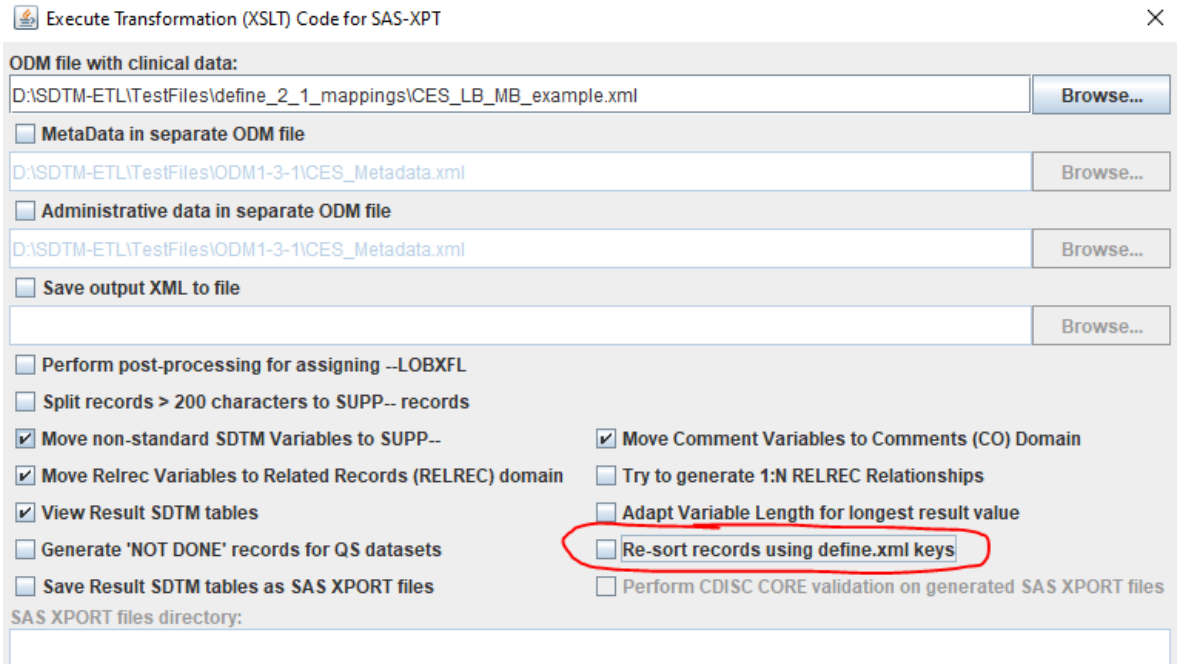


One can then still change the order using the "Move up" and "Move down" buttons.

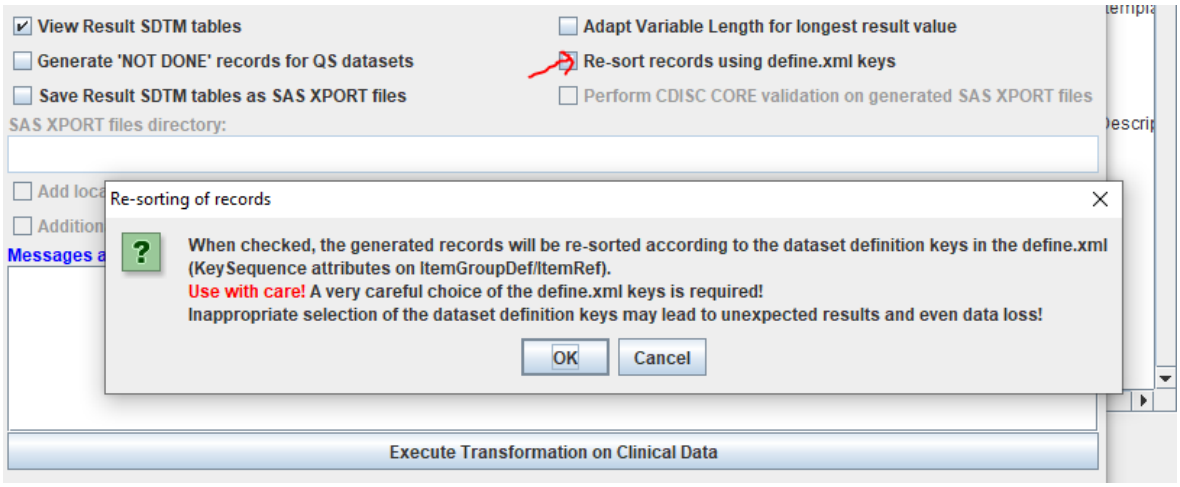
After clicking OK, the keys have been added in the provided order to the dataset definition in the define.xml:



When now starting executing the mappings, an additional checkbox appears:



when it is checked, a message appears:



reminding the user that use of this feature requires a careful choice of the keys for sorting!

IMPORTANT remark:
The "key variables" are not the same thing as the "looping variables". The latter define how the iteration over the data points in the source (i.e. ODM) data is to accomplished as defined by the XPath in the mapping for that variable. For example, in "Findings" domains, it will often be sufficient to set xxTESTCD as the "looping variable" and take care in the XPath for xxTESTCD that all suitable tests, for all or a specific set of visits, for that domain/dataset is represented.

Better handling of hypervertical structures - new wizards

Especially in phase 1 studies, one often sees the use of "hypervertical" structures, e.g. that

"flat" tables are used with one column containing the parameter (code), often a column containg the parameter label, some columns containing further attributes, and a column containing the values. In database science, this model is named the EAV model (Entity-Attribute-Value) and is advantageous when there is a large number of parameters.
When such tables, often stored / exchanged as Excel or CSV files are used in SDTM-ETL, they first need to be transformed into ODM (e.g. using the ODMGenerator software), often leading to a single Form with a single ItemGroup. For example:

```
<ClinicalData StudyOID="MyStudy_HV"
    MetaDataVersionOID="MV.MyStudy_HV">
    <SubjectData SubjectKey="1001">
        <StudyEventData StudyEventOID="SE.0">
            <FormData FormOID="FO.DEFAULT">
                <ItemGroupData ItemGroupOID="IG.DEFAULT" ItemGroupRepeatKey="36">
                    <ItemData ItemOID="IT.StudyID" Value="MyStudyHV"/>
                    <ItemData ItemOID="IT.SubjectNr" Value="3"/>
                    <ItemData ItemOID="IT.AssessmDate" Value="09NOV2022"/>
                    <ItemData ItemOID="IT.AssessmTime" Value="12:13:24">
                    <ItemData ItemOID="IT.AssessmPerfDatetime" Value="09NOV2022:12:13:24"/>
                    <ItemData ItemOID="IT.ActivityName" Value="UrSamStk"/>
                    <ItemData ItemOID="IT.ParameterName" Value="U03Urobi"/>
                    <ItemData ItemOID="IT.ParameterDescription" Value="Urobilinogen (urine)"/>
                    <ItemData ItemOID="IT.ParameterValue" Value="3.2"/>
                    <ItemData ItemOID="IT.Unit" Value="umol/L"/>
                </ItemGroupData>
            </FormData>
            <FormData>
                ...
            </FormData>
            ...
        </StudyEventData>
    </SubjectData>
</ClinicalData>
```

Although completely valid ODM, it is a bit unusable, as "ItemOID" is usually used to store the test code or name (e.g "IT.Bilirubin") and "Value" is used to store the collected value (e.g. "3.2").

In SDTM-ETL, this classic structure allows easy drag-and-drop and selection of items to be mapped to SDTM domains and variables. With hypervertical structures, the XPath expressions are a bit more complicated, as a single "ItemData" does not contain test code and test value anymore, but that is spread over 2 "ItemData" elements within the same ItemGroupData. On the other hand, for the parameter (here in the ItemData with OID "IT.ParameterName", very often one will have a codelist generated, containing an item for each separate test.

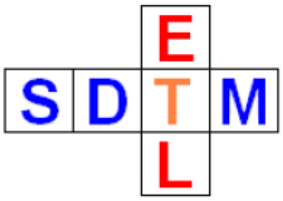
So, a somewhat different mapping mechanism is required, for which a new wizard has now been added.

The complete description of this new feature can be found in the separate tutorial "Working with hypervertical structures" which can be found on the SDTM-ETL website under "tutorials".

SDTM-ETL 4.2 User Manual and Tutorial

Author: Jozef Aerts, XML4Pharma

Last update: 2023-04-09



Tutorial: Working with hypervertical structures

Table of Contents

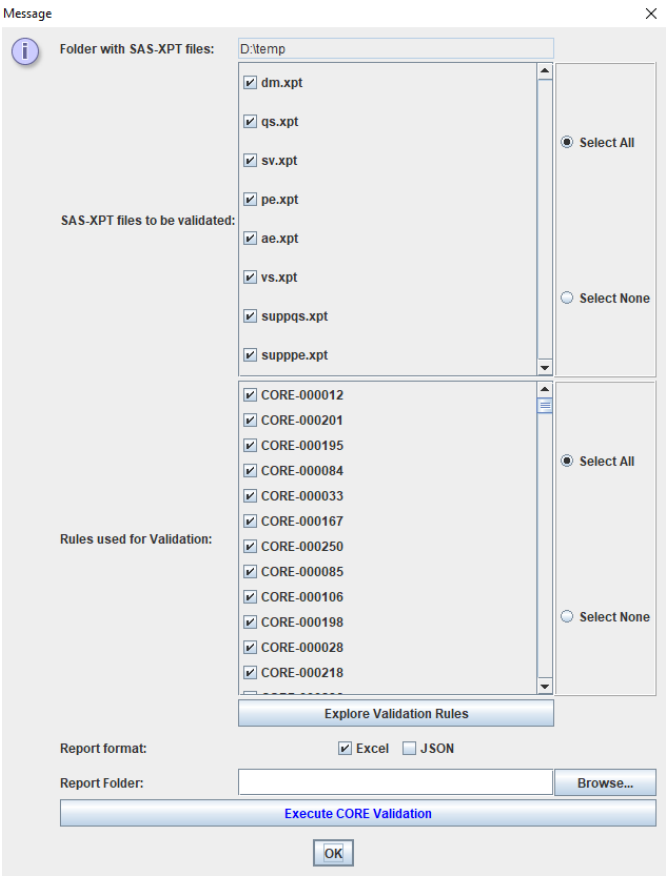
CDISC CORE Validation

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.



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:

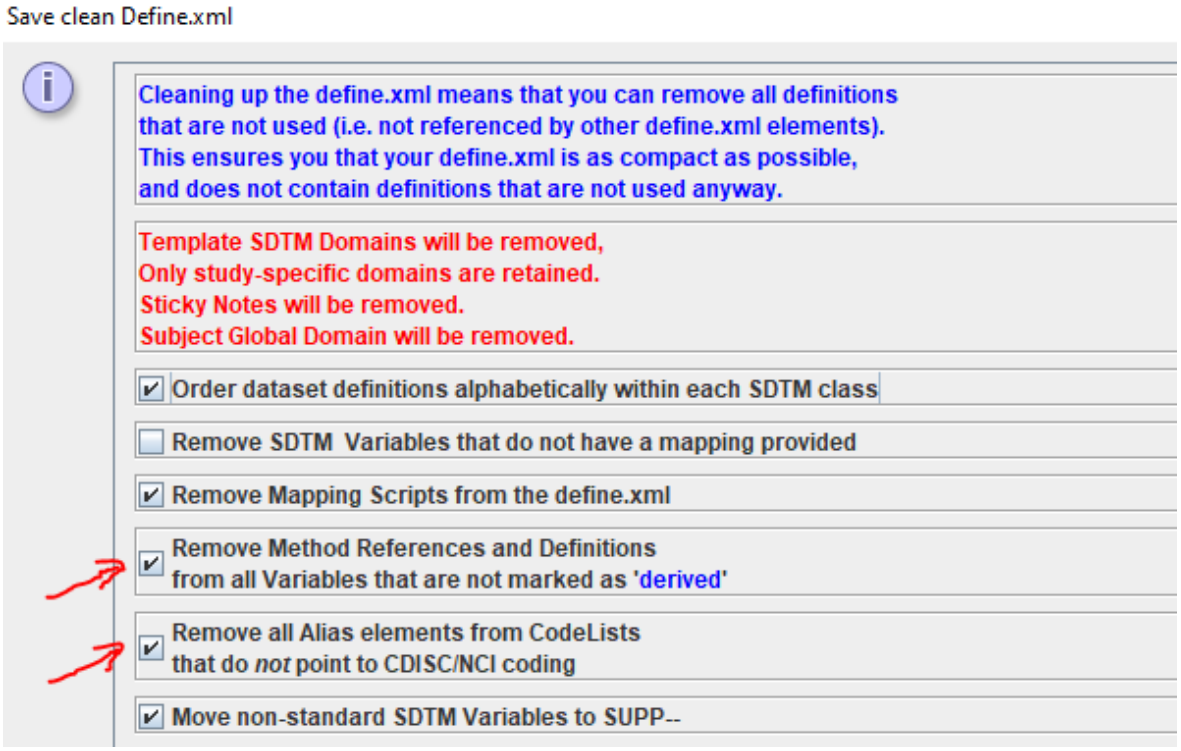
```
<CodeList OID="CL.C66769.AESEV"
          Name="Severity/Intensity Scale for Adverse Events" DataType="text">
    <EnumeratedItem CodedValue="MILD">
        <Alias Context="nci:ExtCodeID" Name="C41338"/>
    </EnumeratedItem>
    <EnumeratedItem CodedValue="MODERATE">
        <Alias Context="nci:ExtCodeID"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



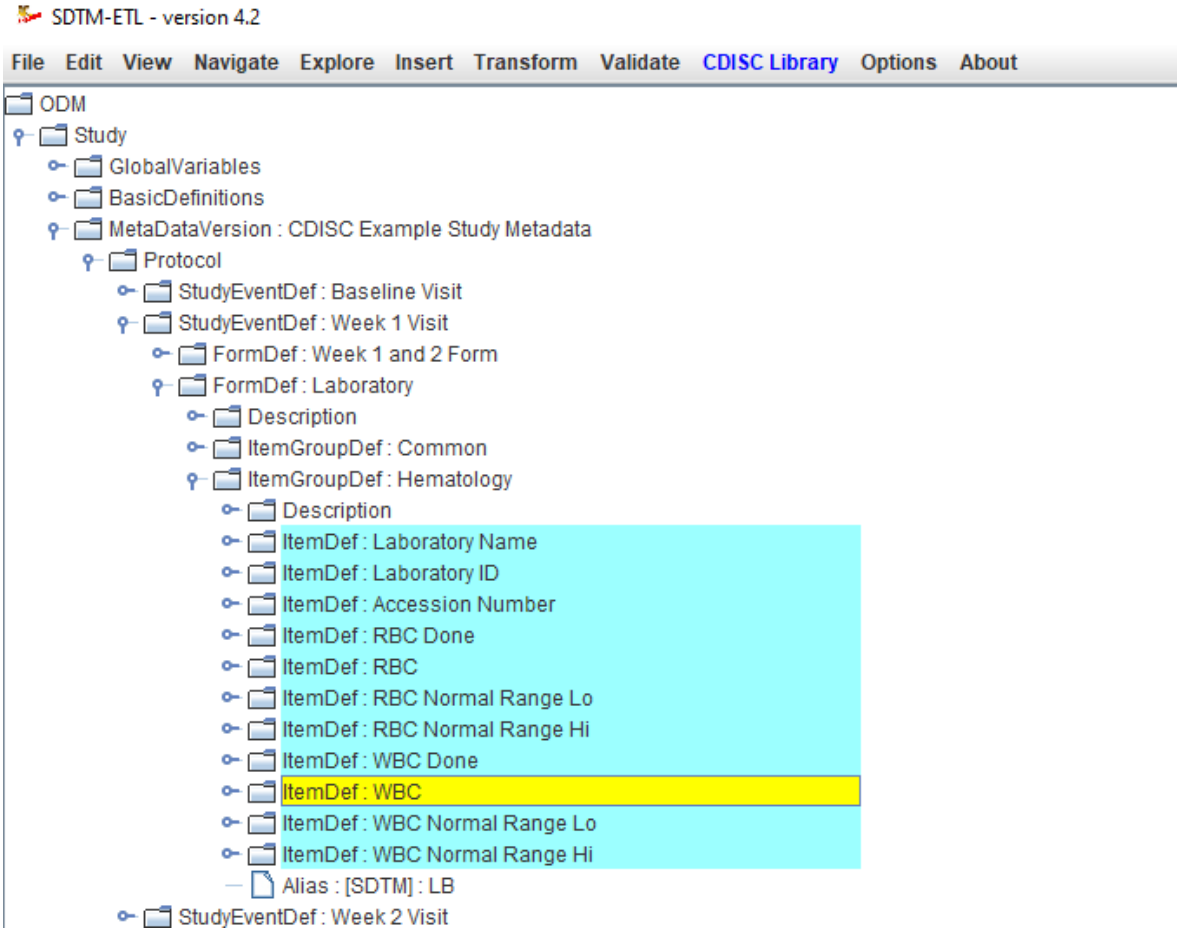
Also new is the checkbox "Remove Method References and Definitions from all Variables that are not marked as "derived". It takes care that all references to, and the method definitions are removed for all SDTM/SEND variables for which the "Origin" is not "derived", so only retaining the ones for "derived"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

ChatGPT and word similarity use for mapping suggestions

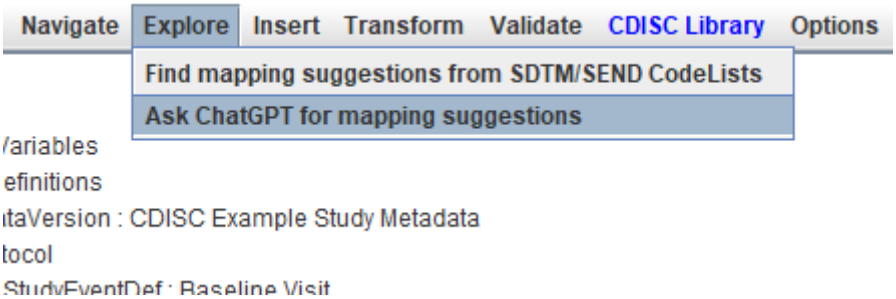
ChatGPT (based on artificial intelligence - AI) has, for many of us, become part of our daily life. Especially for SDTM beginners, it can provide reasonable hints for mappings. Therefore,

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).

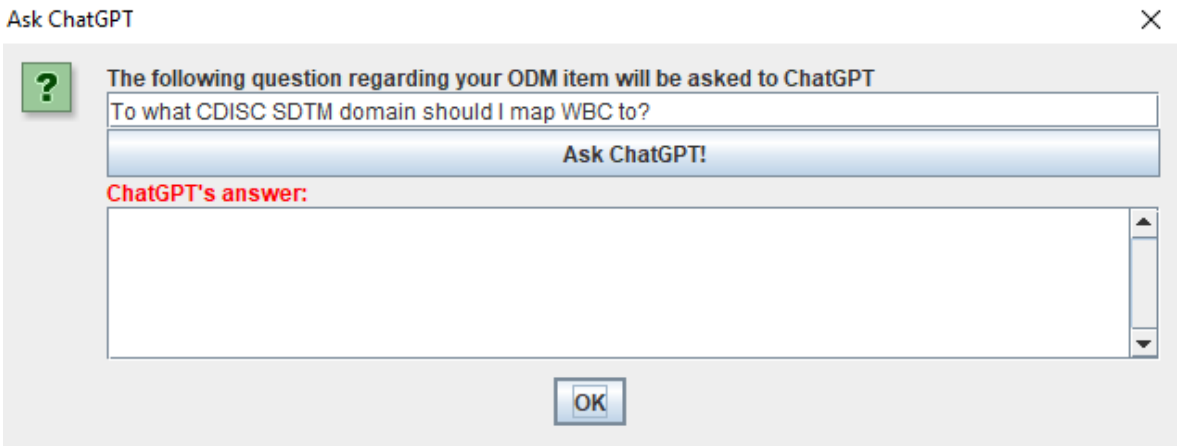
In order to use ChatGPT for obtaining a mapping hint, first select an item in the ODM tree, for example "WBC":



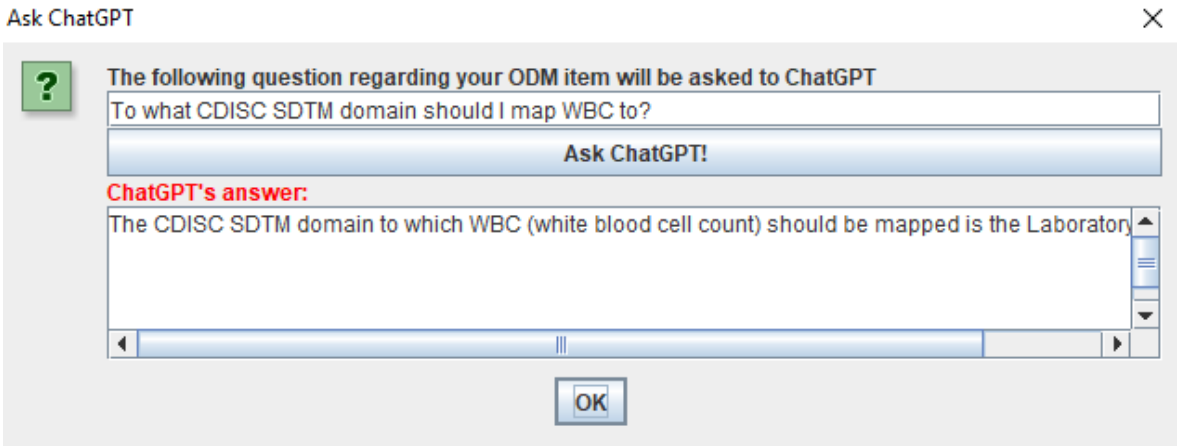
Then use the (new) menu "Explore - Ask ChatGPT for mapping suggestions",



leading to a pre-filled dialog:



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

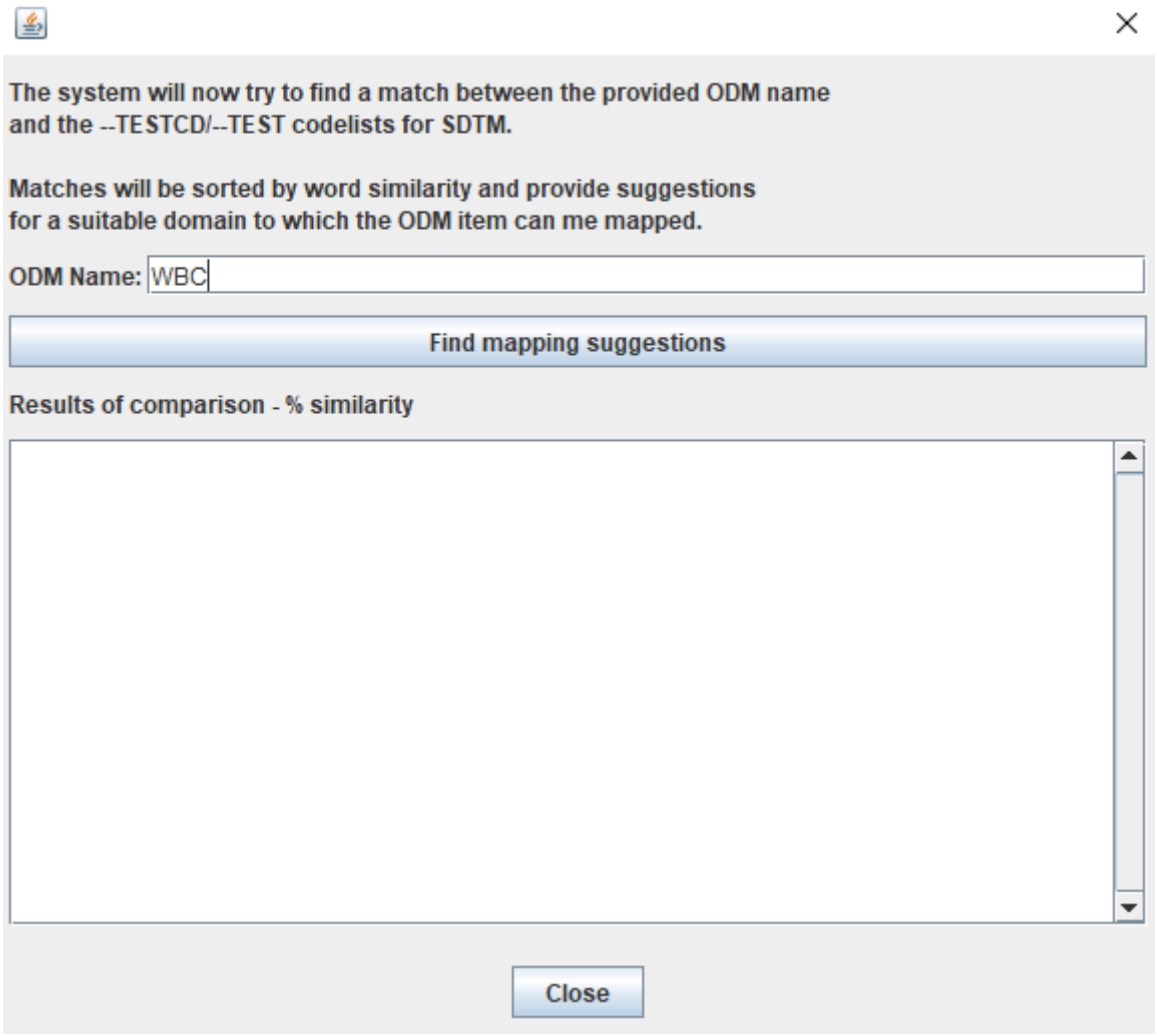


with ChatGPT's answer: "*The CDISC SDTM domain to which WBC (white blood cell count) should be mapped is the Laboratory (LB) domain.*"

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:



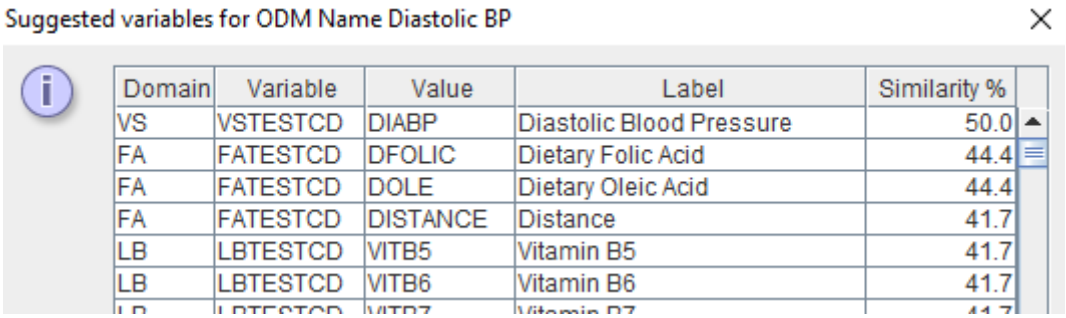
Clicking "Find mapping suggestions" then starts a process, the system going to all the CDISC codelists, and then sorting according to word similarity. This may take a few minutes, so maybe time to go for a cup of tea or coffee.

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

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:

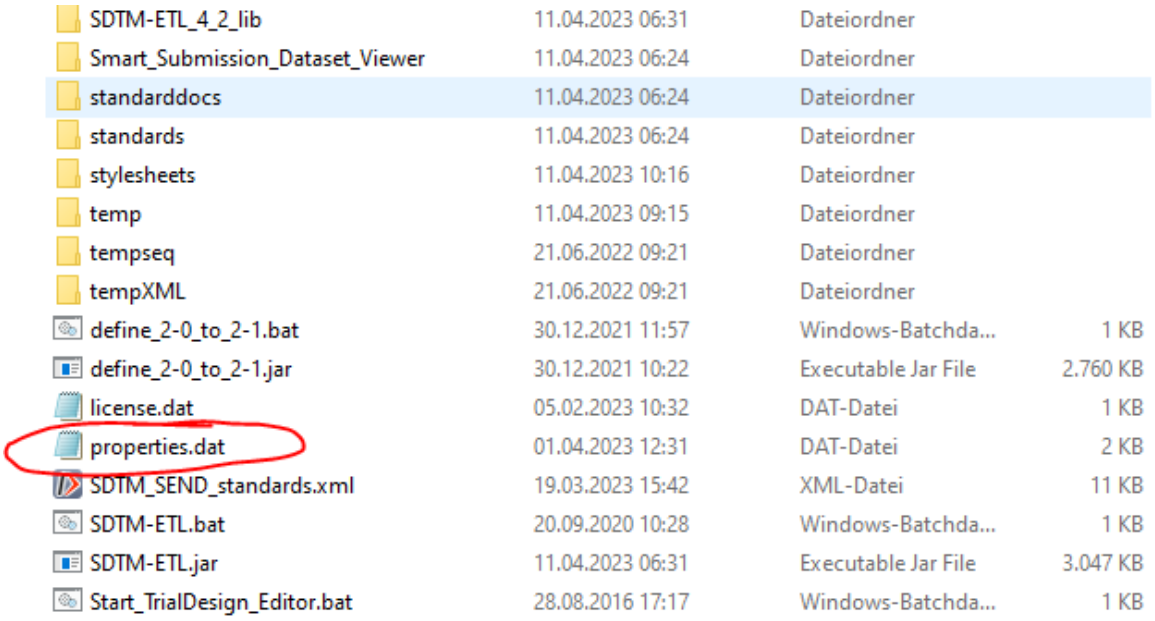


Domain	Variable	Value	Label	Similarity %
VS	VSTESTCD	DIABP	Diastolic Blood Pressure	50.0
FA	FATESTCD	DFOLIC	Dietary Folic Acid	44.4
FA	FATESTCD	DOLE	Dietary Oleic Acid	44.4
FA	FATESTCD	DISTANCE	Distance	41.7
LB	LBTESTCD	VITB5	Vitamin B5	41.7
LB	LBTESTCD	VITB6	Vitamin B6	41.7

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:



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:

*properties.dat - Editor

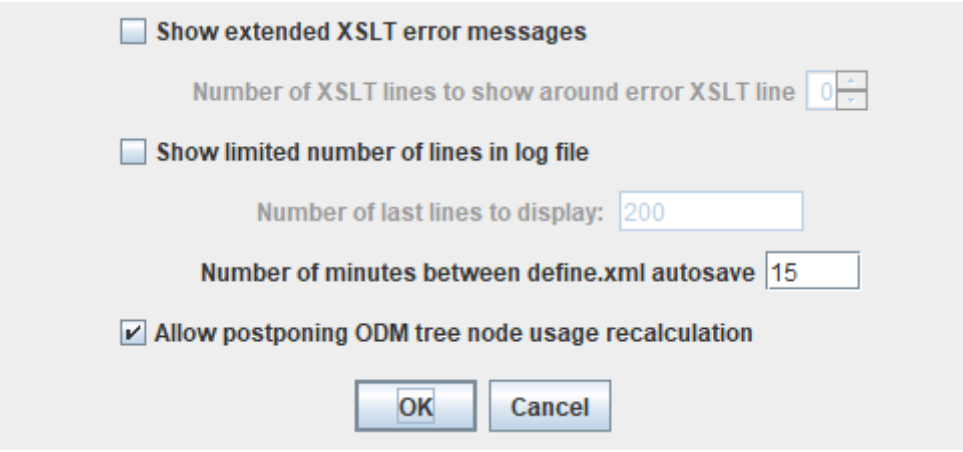
Datei Bearbeiten Format Ansicht Hilfe

```
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":



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.

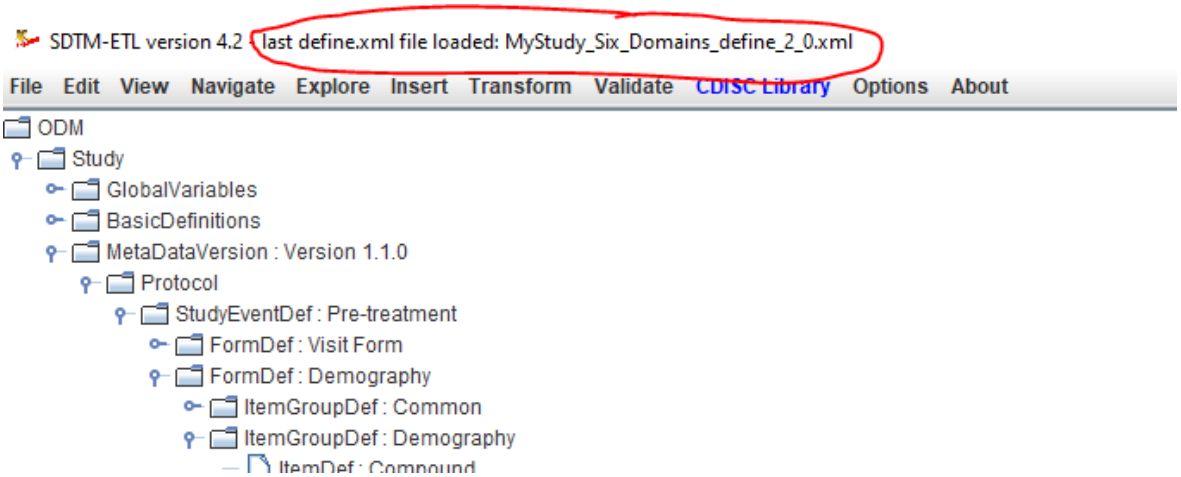
We also removed the property to indicate where the Pinnacle21 validation software is located from the "properties.dat file", as it is unclear whether starting up this software from another software is still allowed by the Pinnacle21 license. Furthermore, we strongly believe there is no place in future 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:



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.