SDTM-ETL 4.4 User Manual and Tutorial

Author: Jozef Aerts, XML4Pharma

Last update: 2024-02-01

Handling multiple Codelists: CDISC Controlled Terminology Relationships

Table of Contents

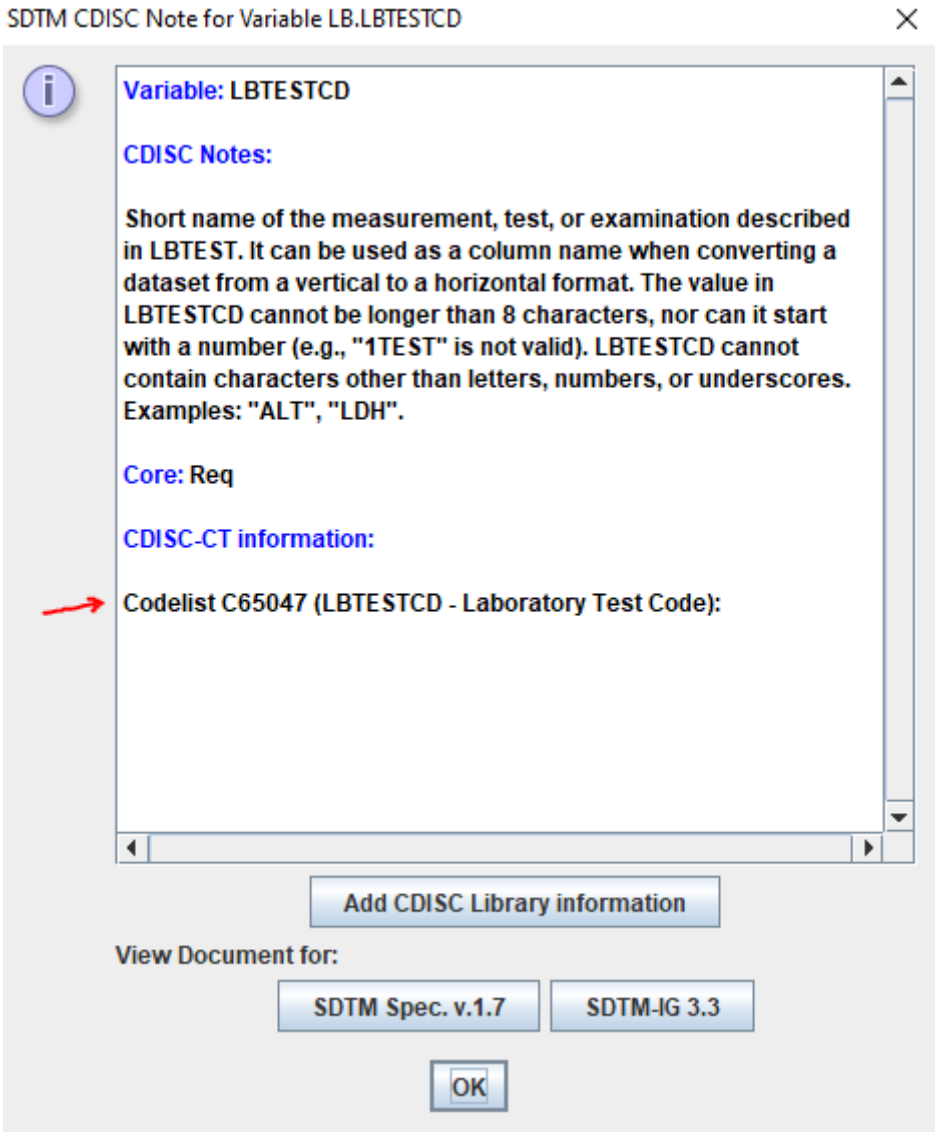
Introduction

For a number of variables in SDTM, there are multiple codelists. Examples are EGSTRESC, and DSDECOD. Also in oncology studies, we may have to choose between different codelists for RSCAT, even having different versions of RECIST.
Usually, the one to be chosen will depend on the use case, or on the value of another variable, like for DSDECOD depending on the value of DSCAT.

These "relationships" have been made available in an electronic form by CDISC, for different versions of the standard, and are available in the form of a number of JSON files.

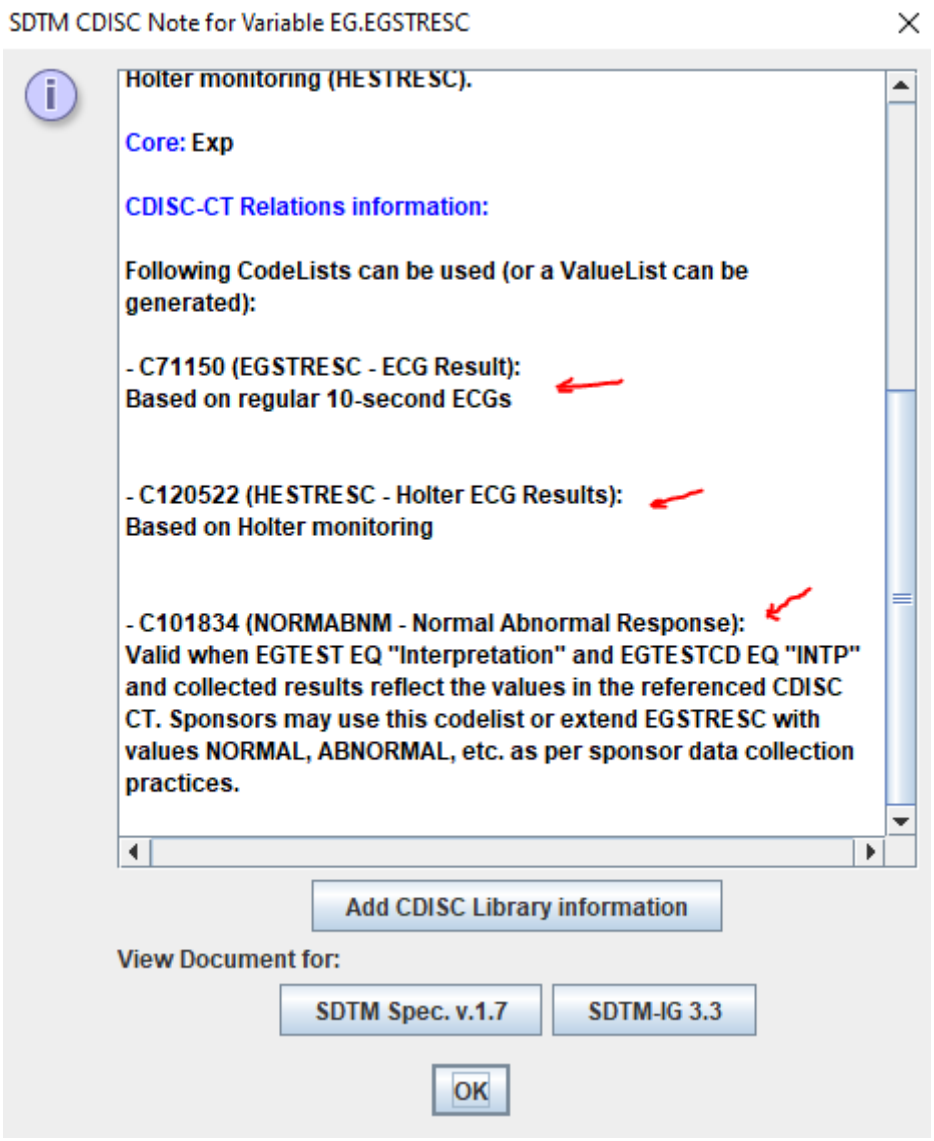
Multiple Codelists

Whether an SDTM variable is under controlled terminology can easily be found out by using the menu "View - SDTM CDISC Notes" (or using CTRL-H). For example, for LBTESTCD, we get:



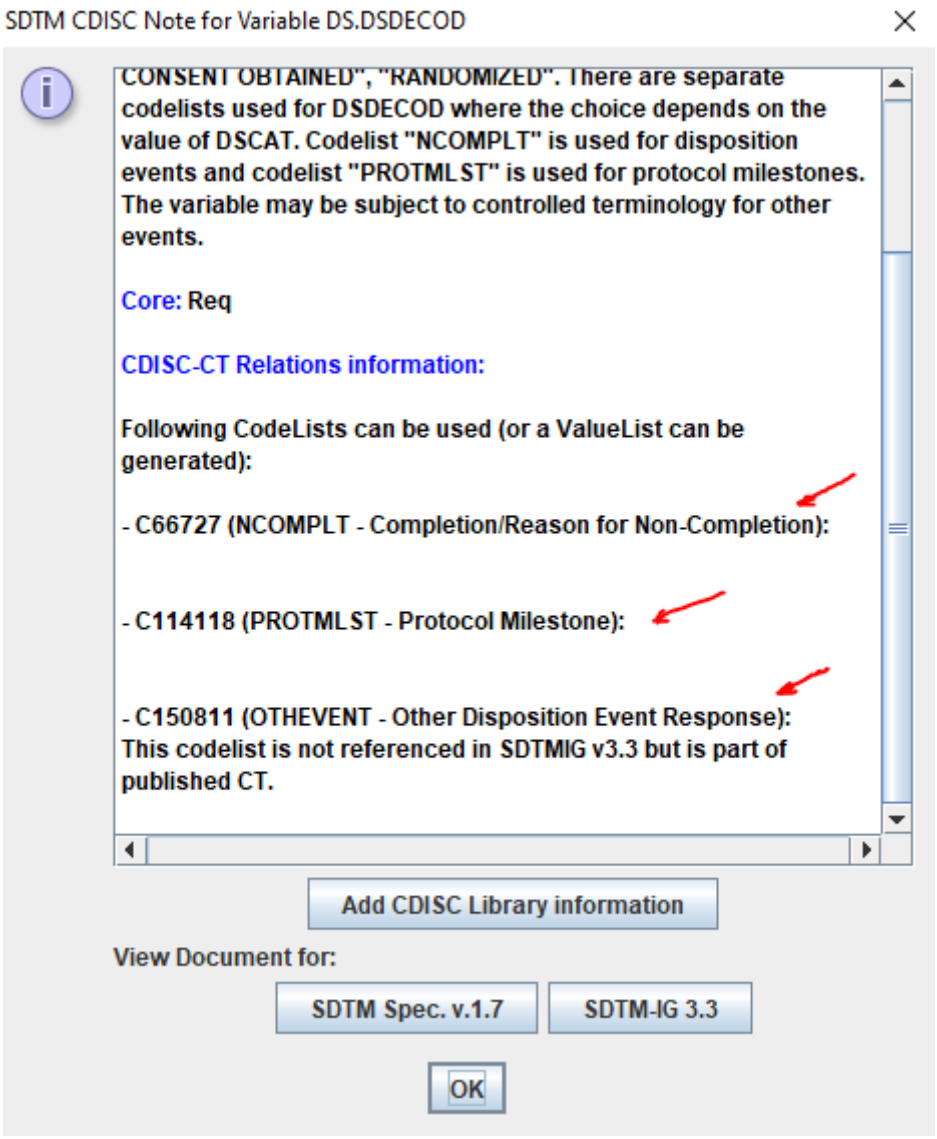
showing that there is only one associated codelist for the variable LBTESTCD.

When is however more than one associated codelist (or better, the user will have to choose between different codelists, depending on the use case or the value of other variables), the "CT-relationships" file for the current SDTMIG version is read, and the information displayed. For example for EGSTRESC, using the menu "View - SDTM CDISC Notes" (CTRL-H), we get:



presenting 3 codelists for EGSTRESC.
It also states that we can either associate one of these 3 codelists, or that we can later generate "value lists" where we will have "where clauses" for determining which codelist is associated for which case (see the separate tutorial "Working with ValuesLists and the WhereClause in define.xml 2.0/2.1").

For DSDECOD, we get:

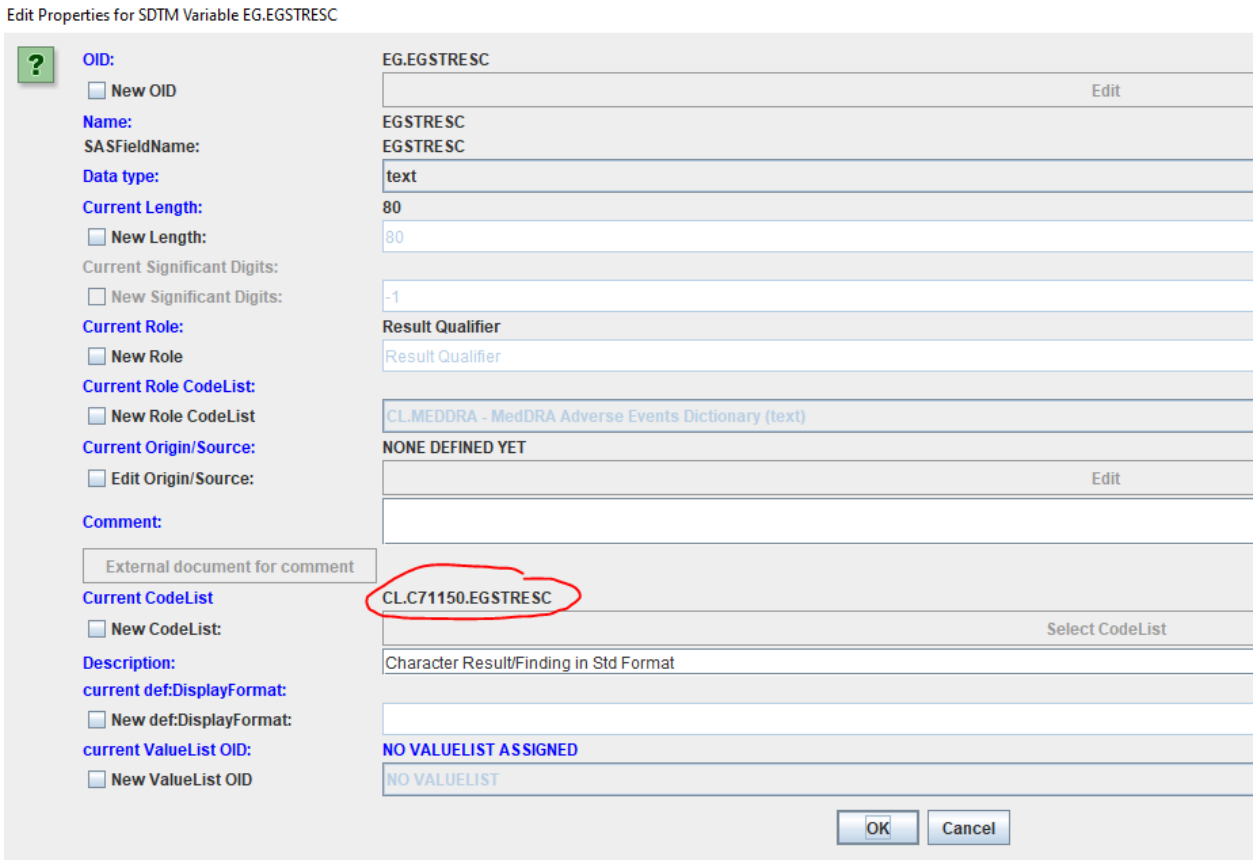


Also presenting us 3 possibilities.

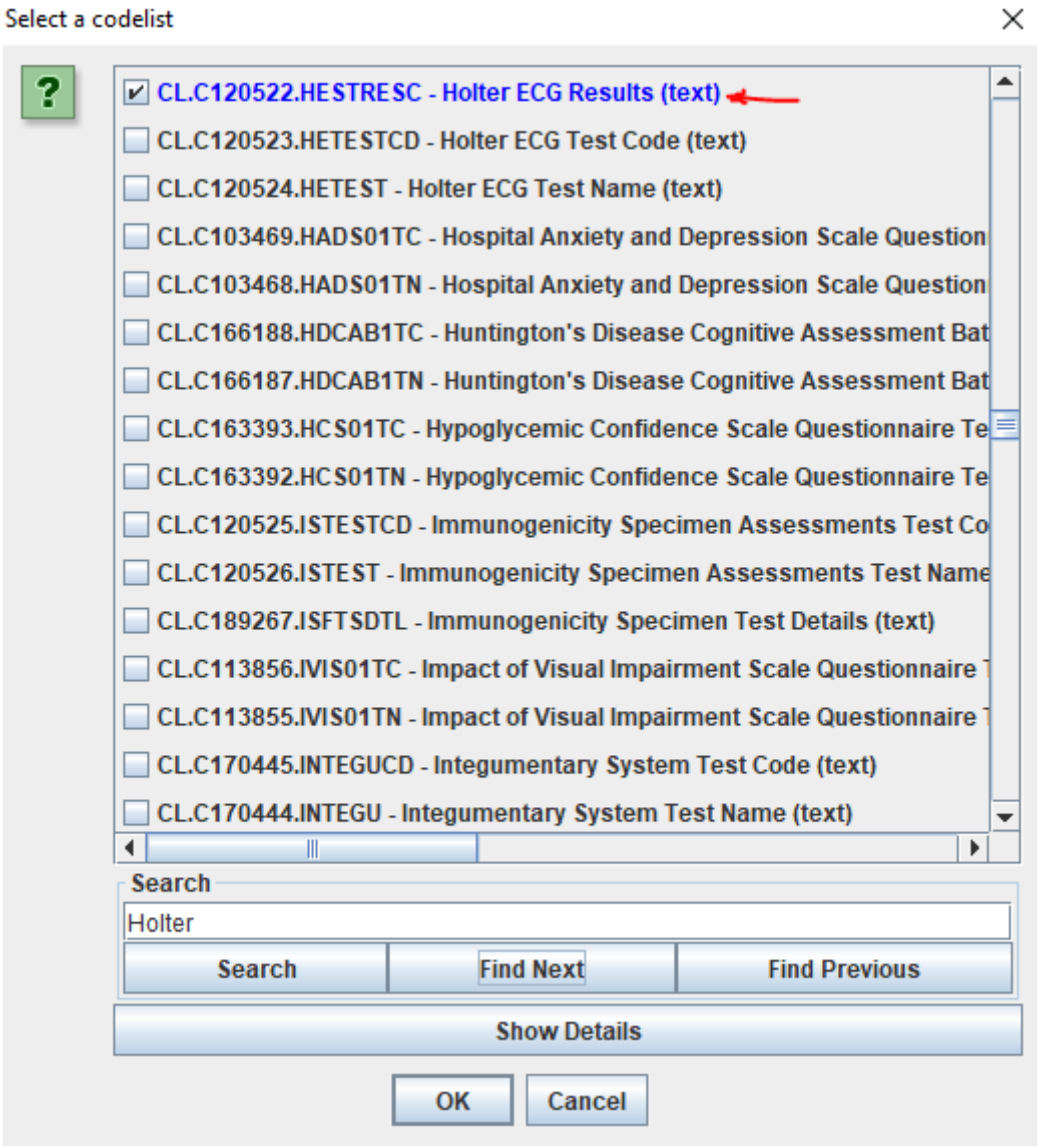
Assigning the codelist

There are now several possibilities for assigning the codelist in the case of multiple codelists.

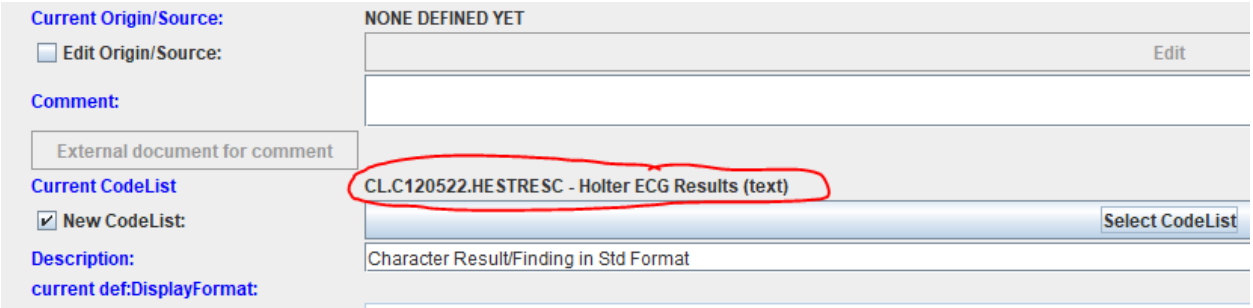
The first is to "edit" the properties of the SDTM variable, using the menu "Edit - SDTM Variable Properties" (CTRL-E). For example, for EGSTRESC, this leads to:



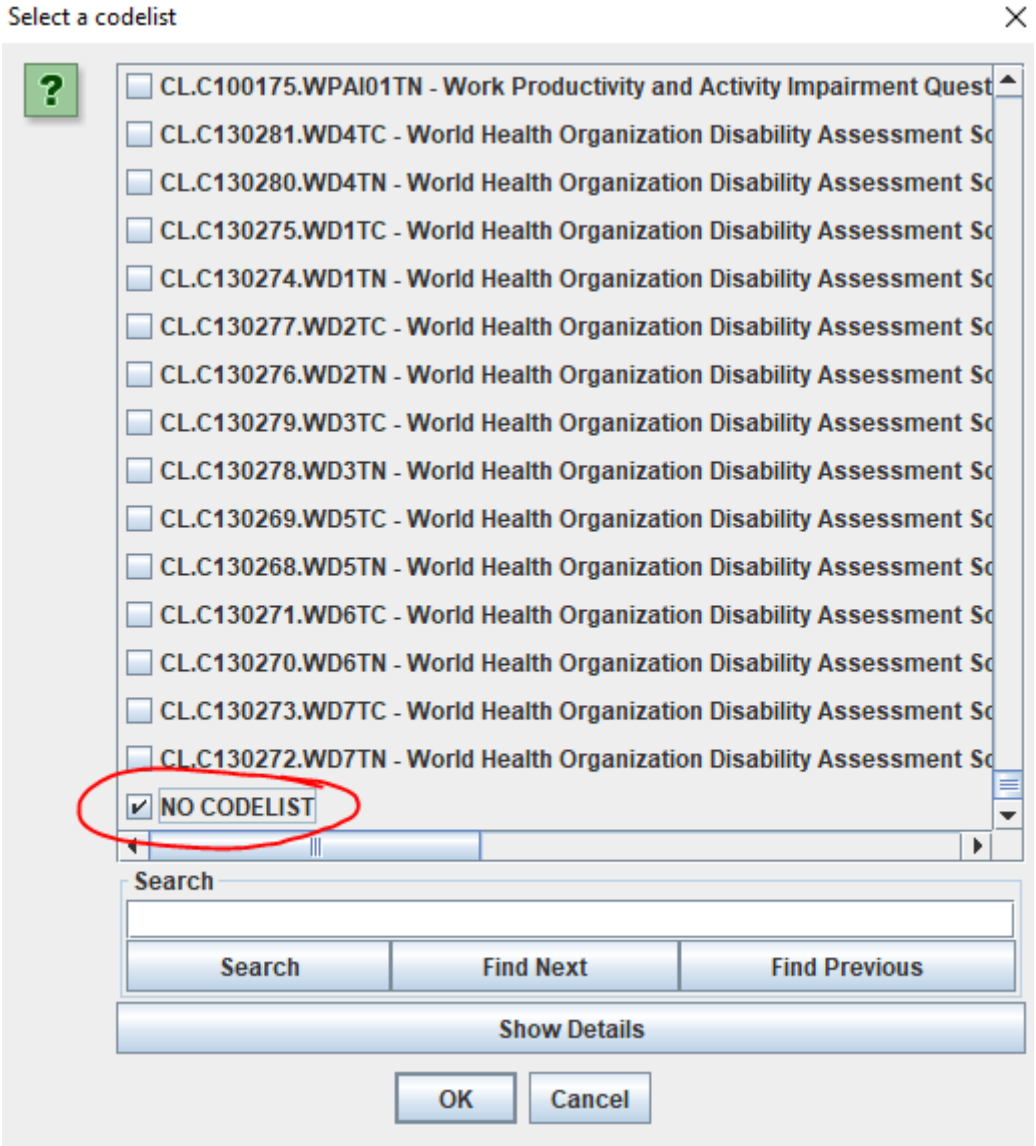
where the most probable value (codelist C71150 - ECG Result based on regular 10-second ECGs) is assigned. By clicking the checkbox "New CodeList" we can now select another, e.g. search for "Holter" and selecting the C120522 codelist:



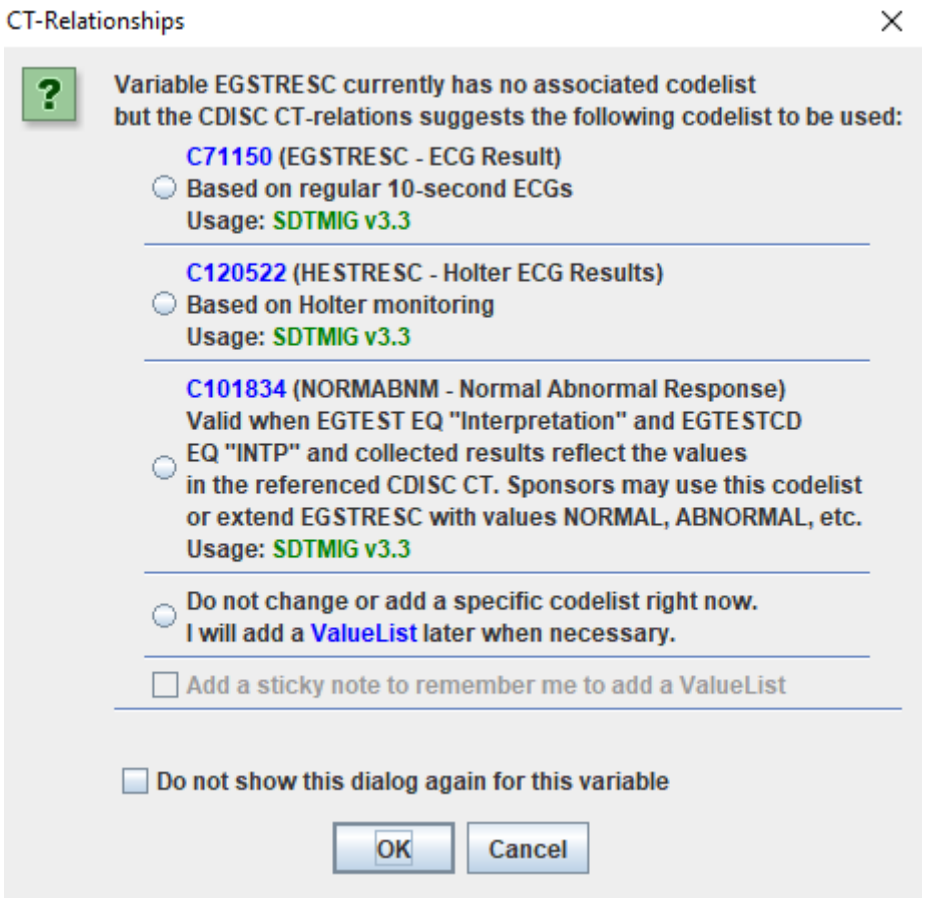
After clicking "OK", the codelist is re-assigned:



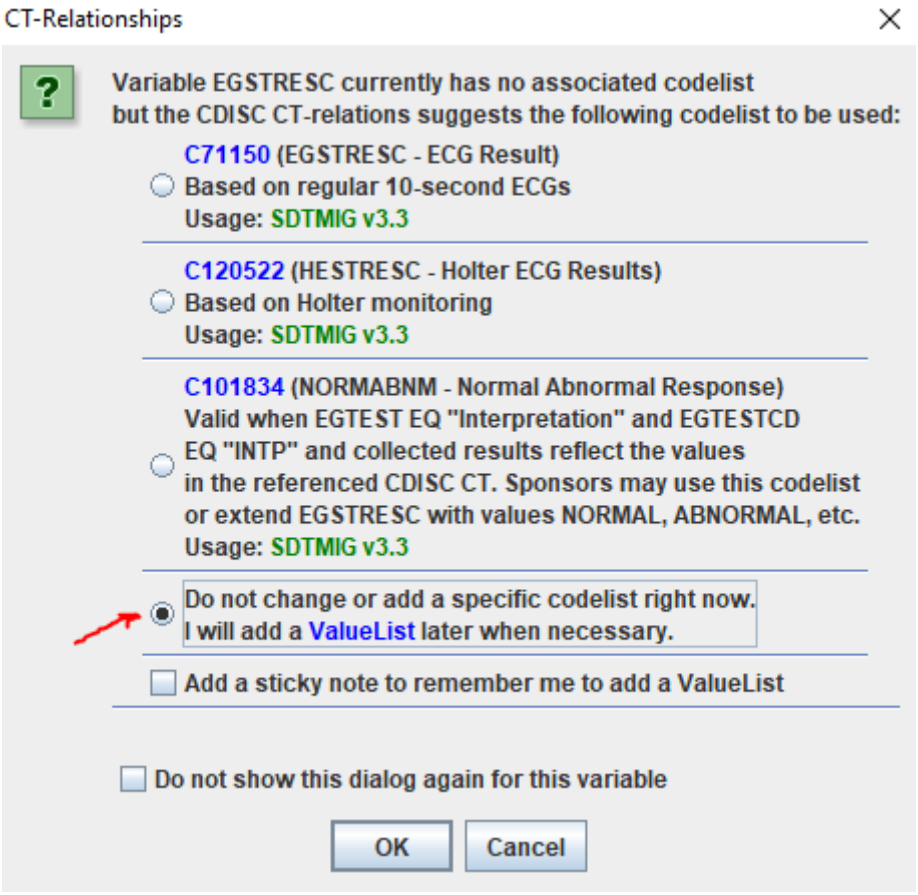
Important now is that in case we want to have a "value list" (ValueList) for the variable, i.e. the codelist to be used for EGSTRESC depends on something else, like the value of EGCAT, we do not assign a codelist on the variable level. To do so, for the codelist selection, select "No CodeList", which is at the bottom of the list:



In case we have no codelist assigned for EGSTRESC, and we start a mapping by drag-and-drop, or by a double-click on the SDTM cell, a dialog will pop up, presenting us the possibilities again:

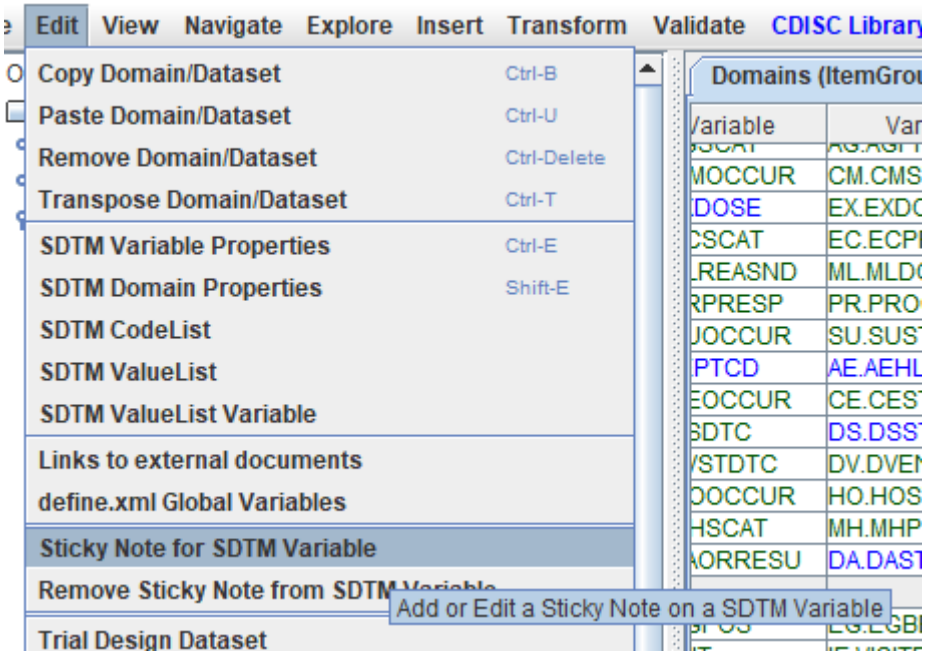


When we indeed want to assign a ValueList later, we select "Do not change or add a specific codelist now …":

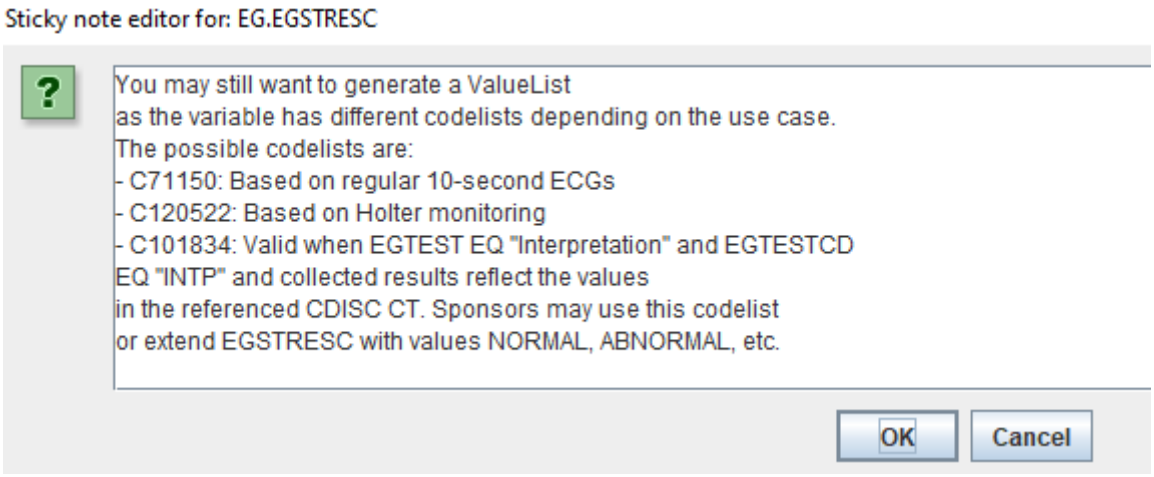


If we then also check the checkbox "Add a sticky note to remember me to add a ValueList", we see that, after closing the mapping, a "pin" is on the SDTM cell:

And when we then use the menu "Edit - Sticky Note for SDTM variable":

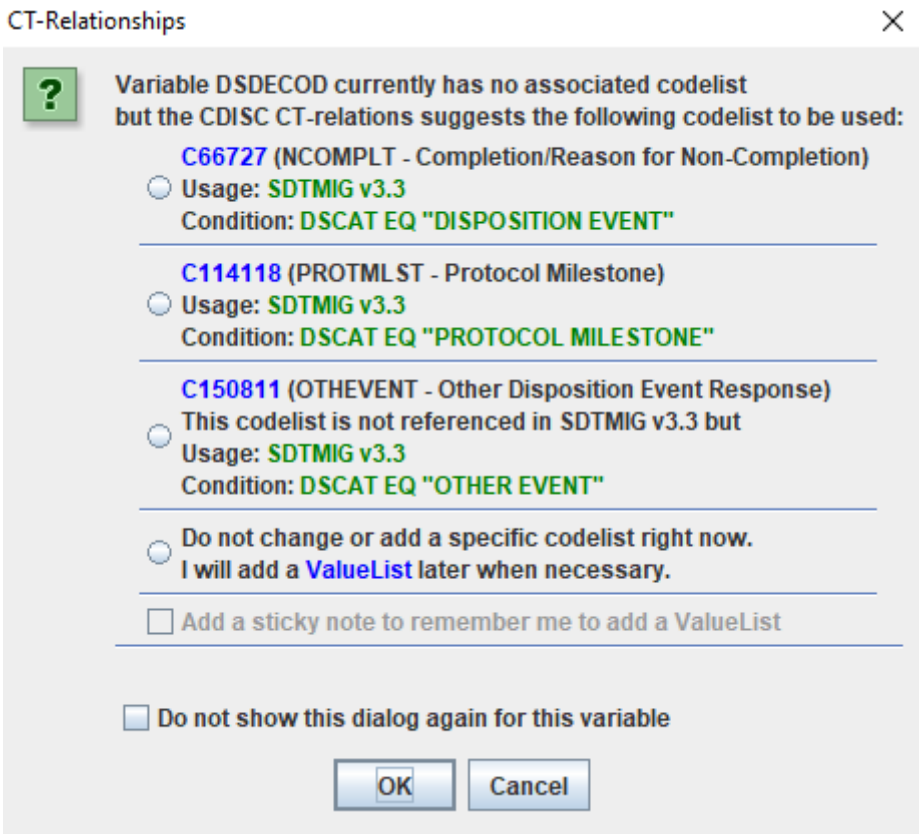


The "sticky note" is visualized and can be edited:



When starting the mapping for EGSTRESC, and a codelist is already associated to the variable, the same dialog will be shown, still allowing to change the choice, but when then selecting "Do not change or add …", the already associated codelist will remain.

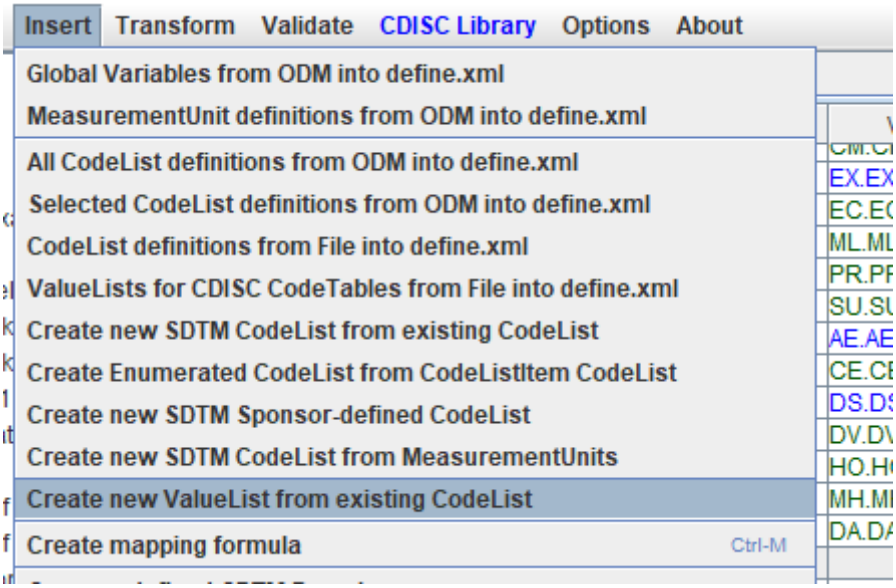
Similarly for DSDECOD, for which no initial codelist is present / assigned, when starting the mapping, the following dialog is displayed:



Creating ValueLists

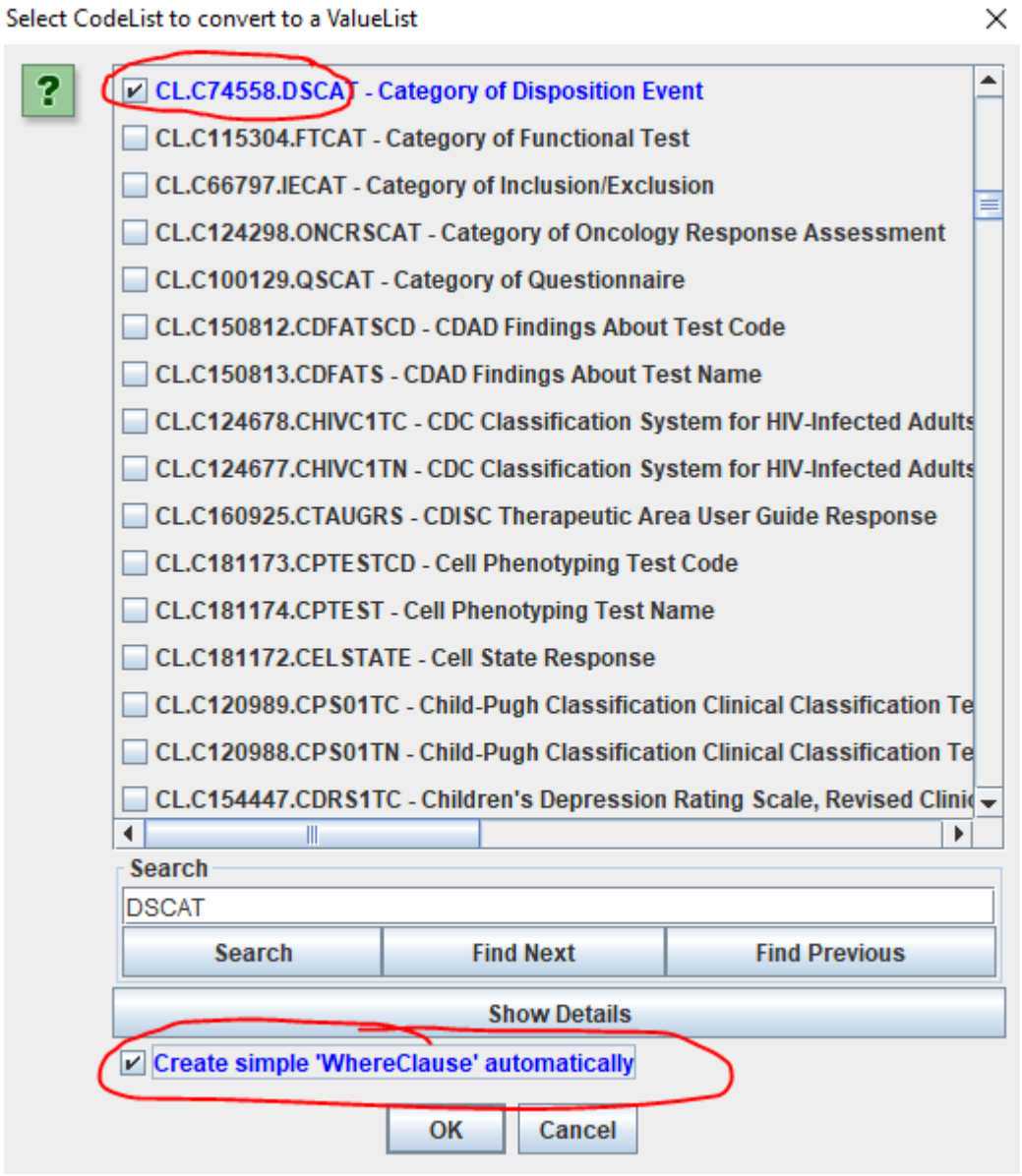
Now, in the case of DS, which contains information about a mixture of events[1], we will typically want to use a ValueList, in which we assign the codelist for DSDECOD depending on the value of DSCAT, as already indicated in the dialog.

We can do this in a semi-automated way by using the menu "Insert - Generate ValueList from Existing CodeList":

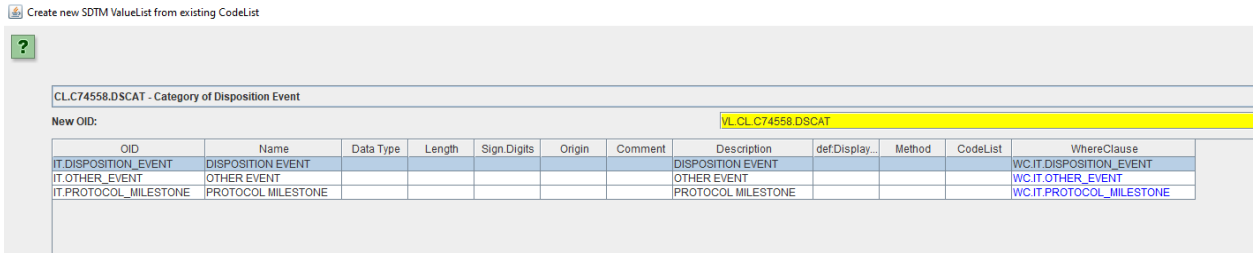


and select the codelist for "DSCAT" as the base of the ValueList:

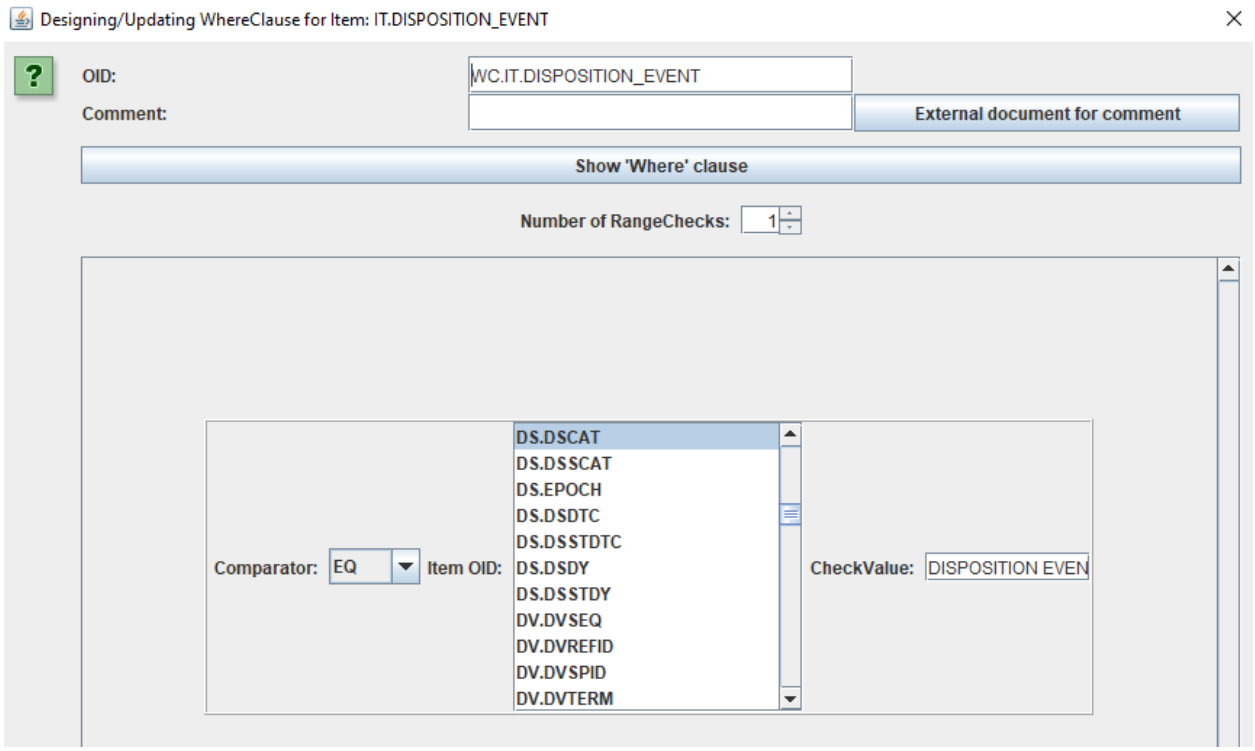
[1] I often call DS "the garbage can of events" ...



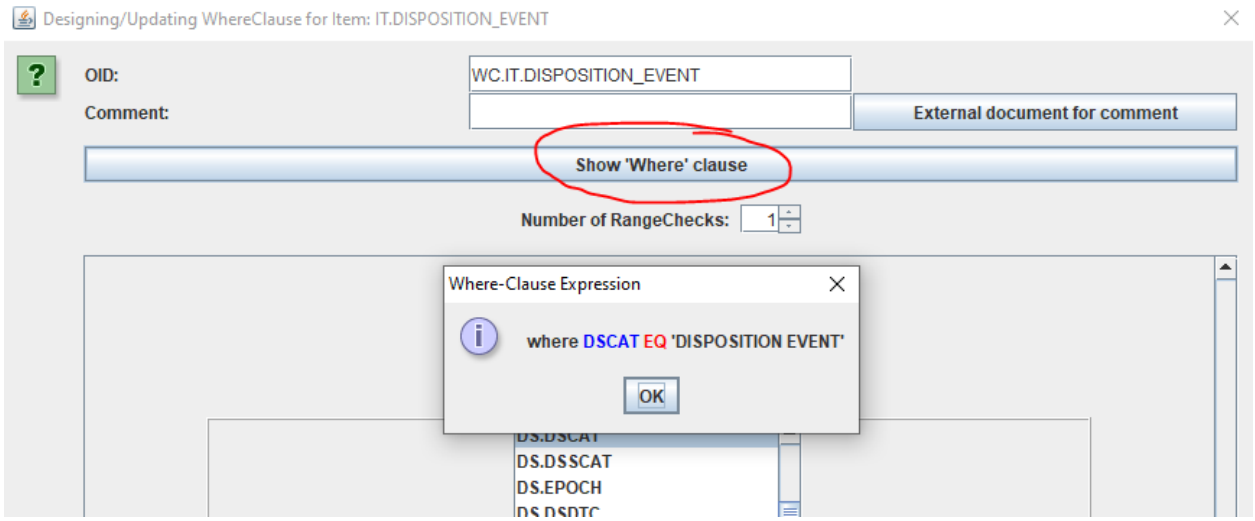
We can also already have the system generate the "WhereClause" automatically by checking the checkbox "Create simple WhereClause automatically". After clicking OK:



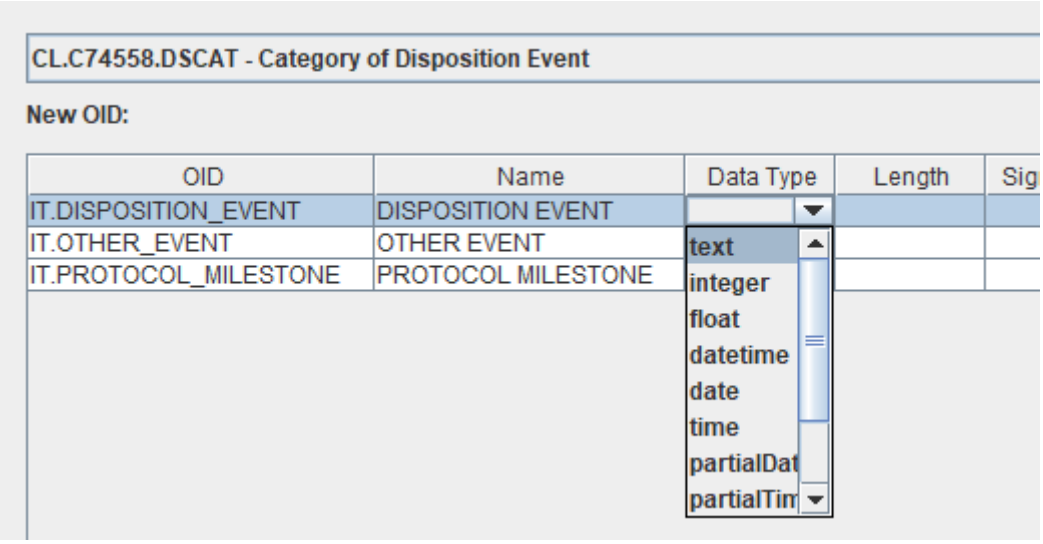
When clicking on one of the "WhereClause" cells, we can edit or visualize the Where-clause:



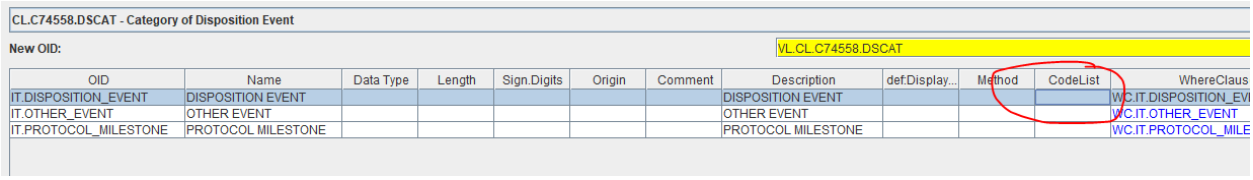
We can add a comment and/or a link to an external document (very uncommon in this case), and getting the human-understandable phrase by clicking the button "Show 'Where' Clause):



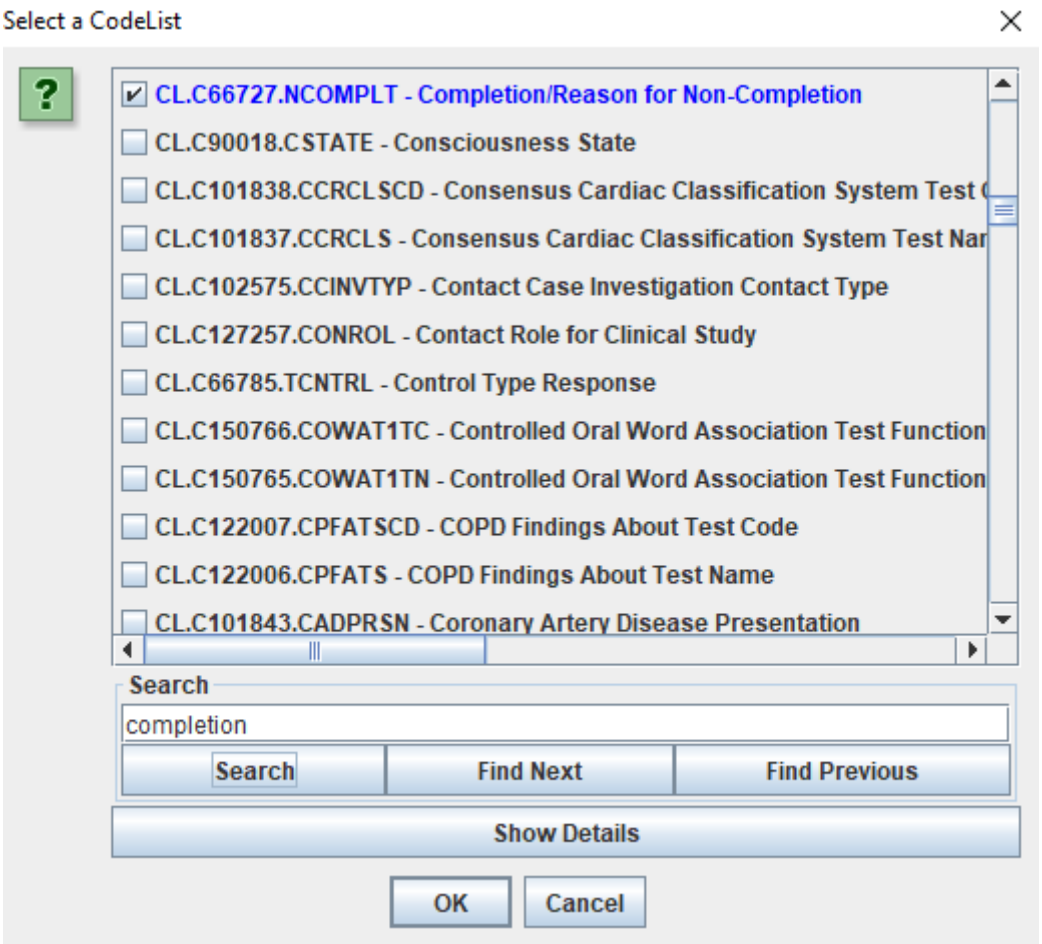
Now going back to the dialog showing the ValueList editor, we can start adding additional information, like the datatype for the 3 use cases:



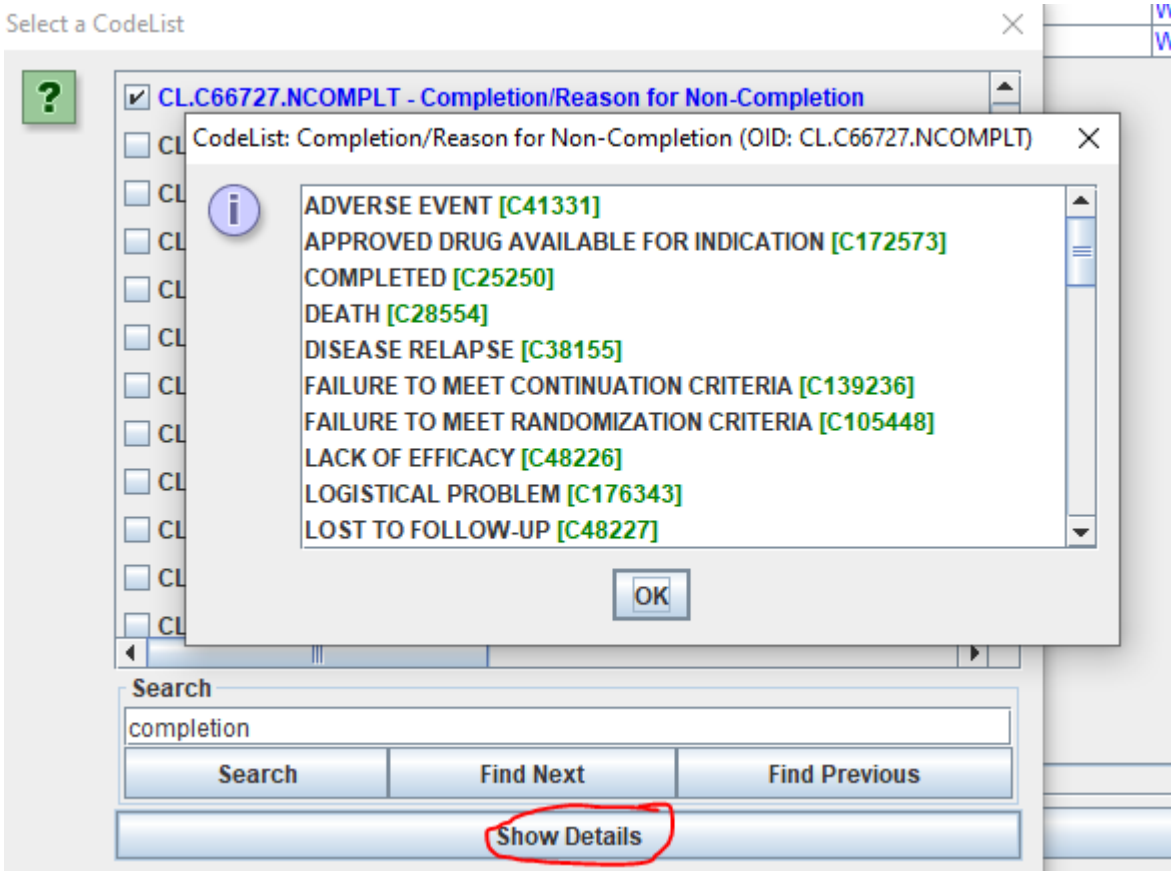
But most importantly, the codelist for each of the use cases. So for the case of "Disposition Event", we can set the associated codelist by clicking in the corresponding cell:



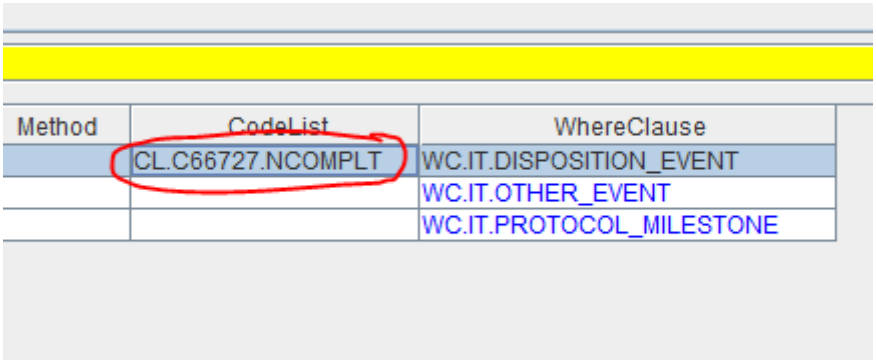
Leading to a "codelist selector":



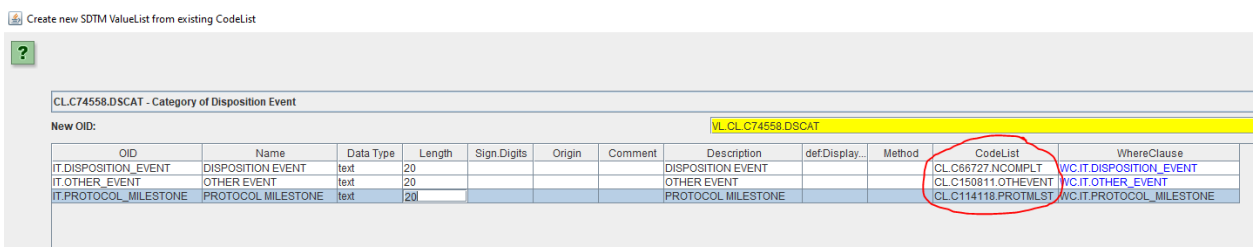
Where we select the codelist "Completion/Reason for Non-Completion".
We can show the details of the codelist by clicking "Show Details" leading to:



And after clicking "OK" twice, we see that the codelist has been assigned for the case of "disposition event":

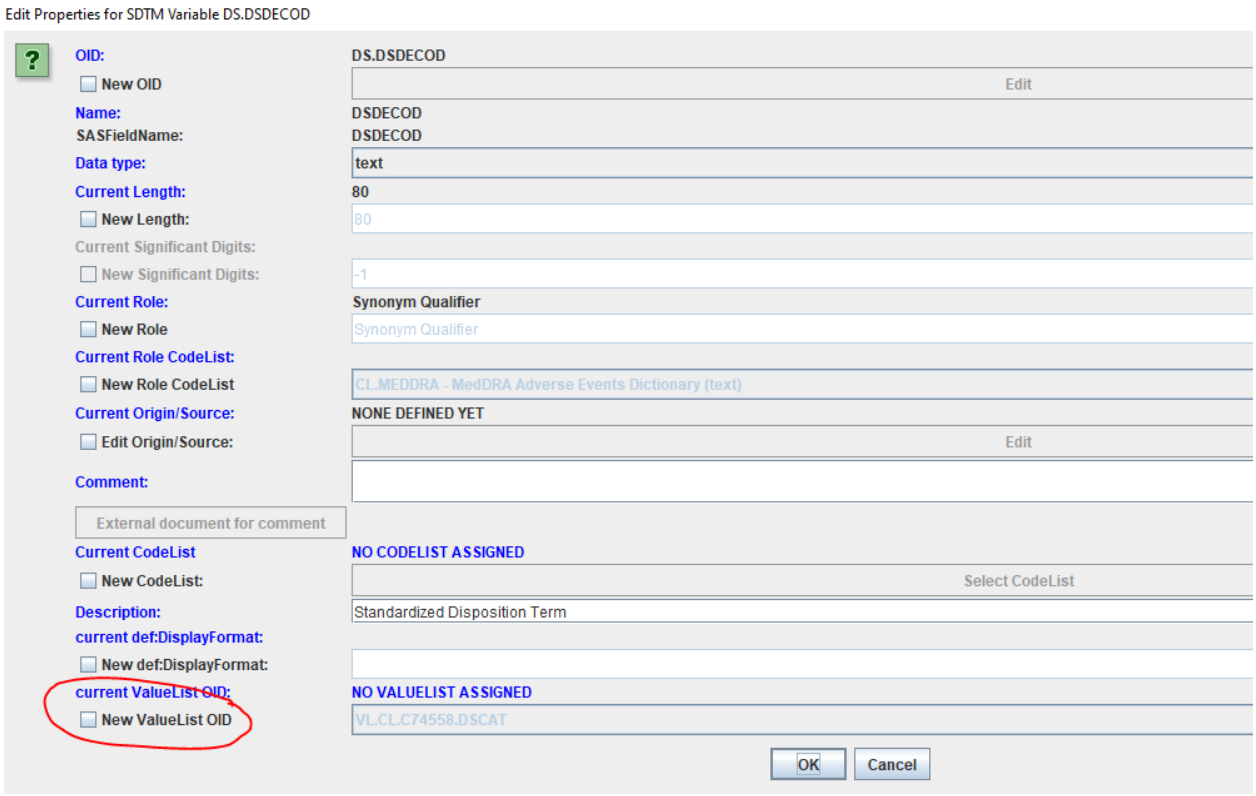


We can then do the same for the cases of "Other Event" and "Protocol Milestone", leading to e.g.:

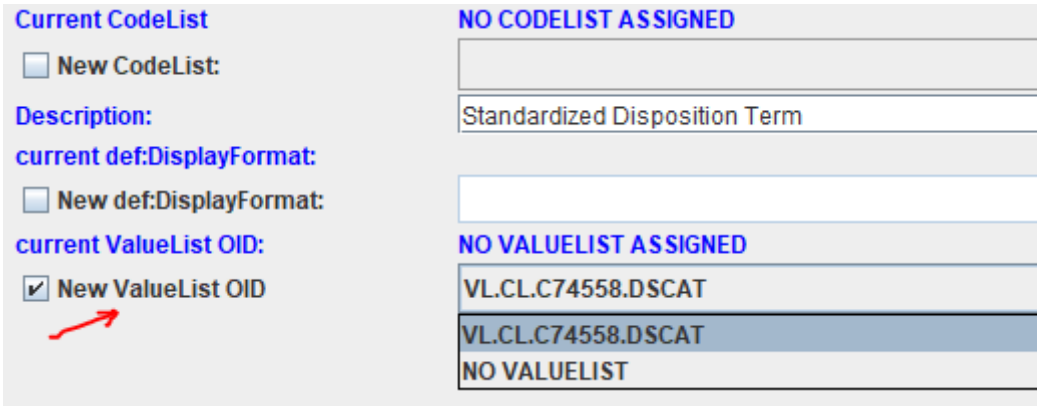


Remark that we just assign a "dummy" value for the maximal length for now, as we do not know yet what the values will be. This is essentially only needed because of the antique SAS Transport 5 format, and irrelevant when using a modern submission format like CDISC Dataset-JSON which will probably the FDA format of choice soon.

We still then need to assign this ValueList (with identifier VL.CL.C74558) to DSDECOD.
To do so, select the DSDECOD cell in the SDTM table, and use the menu "Edit - SDTM Variable Properties" (CTRL-E):



and check the checkbox "New ValueList OID", and then select the one we just created:



When then displaying the define.xml in the browser (using View - View define.xml in Browser), we e.g. get:

DSSPID		Sponsor-Defined Identifier	text	Identifier	80	
DSTERM		Reported Term for the Disposition Event	text	Topic	80	
DSDECOD VLM		Standardized Disposition Term	text	Synonym Qualifier	80	
	DSCAT = "DISPOSITION EVENT"	DISPOSITION EVENT	text		20	Completion/Reason for Non-Completion [42 Terms]
	DSCAT = "OTHER EVENT"	OTHER EVENT	text		20	Other Disposition Event Response • "FINAL CONTACT" • "SITE TRANSFER" • "TREATMENT UNBLINDED" • "WITHDRAWAL OF CONSENT FROM PROTOCOL-SPECIFIED ACTIVITY"
	DSCAT = "PROTOCOL MILESTONE"	PROTOCOL MILESTONE	text		20	Protocol Milestone [11 Terms]
DSCAT		Category for Disposition Event	text	Grouping Qualifier	80	Category of Disposition Event • "DISPOSITION EVENT" • "OTHER EVENT" • "PROTOCOL MILESTONE"

the second column containing the "Where-Clause" and the column on the right, containing the associated codelist with the terms (the latter when the list is not too long).
In this column, clicking on "Protocol Milestone [11 Terms] then jumps to the full codelist:

Protocol Milestone [*C114118*] [CDISC/NCI SDTM 2023-09-29]

Permitted Value (Code)
DECLINED TO CONTINUE INTO NEXT TRIAL ELEMENT [*C186210*]
DECLINED TO CONTINUE INTO SURVIVAL FOLLOW-UP [*C176358*]
ELIGIBILITY CRITERIA MET [*C132447*]
ELIGIBILITY CRITERIA NOT MET [*C164344*]
ENTERED INTO TRIAL [*C161417*]
INFORMED ASSENT OBTAINED [*C161418*]
INFORMED CONSENT OBTAINED [*C16735*]
OPTED TO CONTINUE INTO NEXT TRIAL ELEMENT [*C186211*]
OPTED TO CONTINUE INTO SURVIVAL FOLLOW-UP [*C176357*]
RANDOMIZED [*C114209*]
RE-RANDOMIZED [*C186212*]

In some cases, one will still want to edit or shorten the codelist (e.g. as not all protocol milestones are applicable), using the menu "Edit - SDTM CodeList", or generate a subset (keeping the original one as a 'backup'), or even extend it, using the menu "Insert - Create new SDTM CodeList from existing CodeList", but for DSDECOD, one will usually not do so.

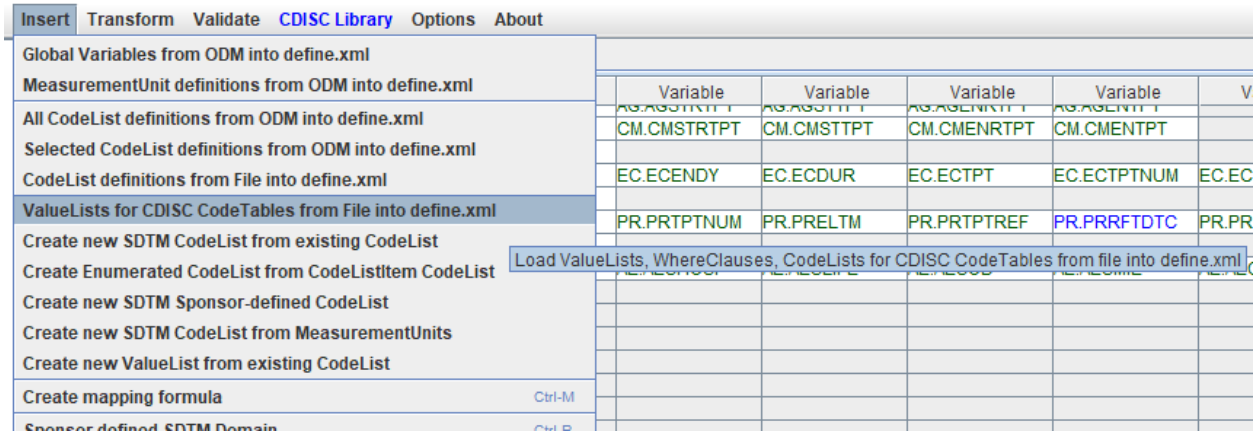
Generating such a valuelist may sound like a lot of work, but once you know how to do this ("it's easy if you know how"), it will take you a few minutes only.
Also remember that you will often use the define.xml with the mappings of one study as a template define.xml for another study. In such a case, you will often not need to go to the above procedure again.
Some of our customers have reported 80% and more "reuse" of the mappings in the case of similar

studies in this way.

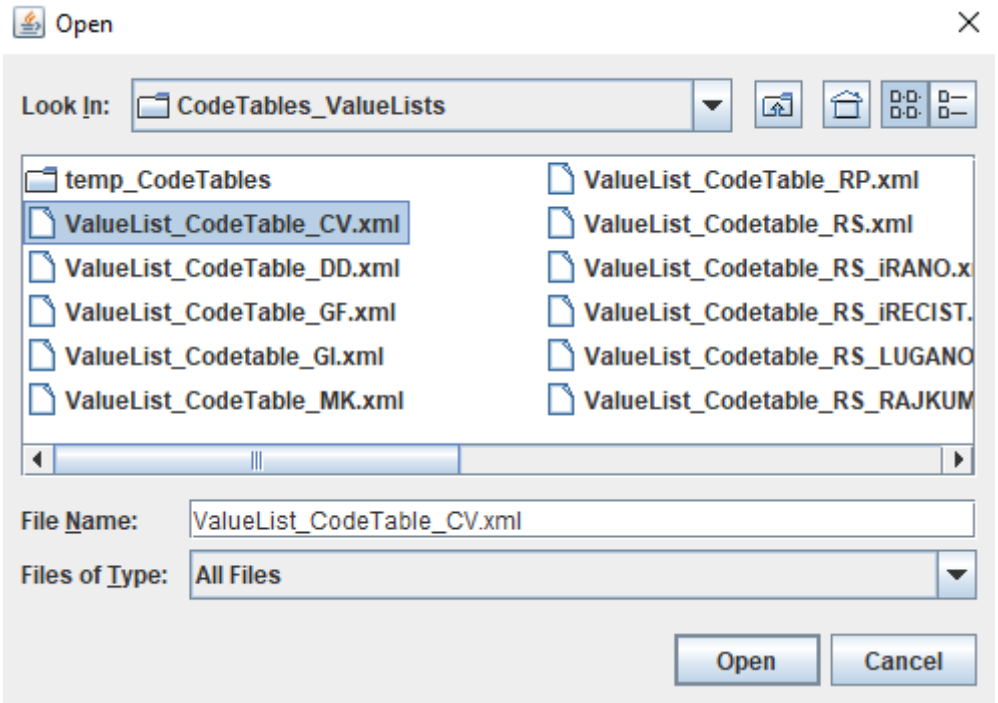
Pre-defined ValueLists

CDISC has published a number of "code tables", unfortunately (but typical for CDISC) as Excel files. We have transformed these into Define-XML valuelists, which can be immediately imported into SDTM-ETL. These can then be used when wanting to assign a ValueList, e.g. when there is more than one associated codelists for an SDTM variable.

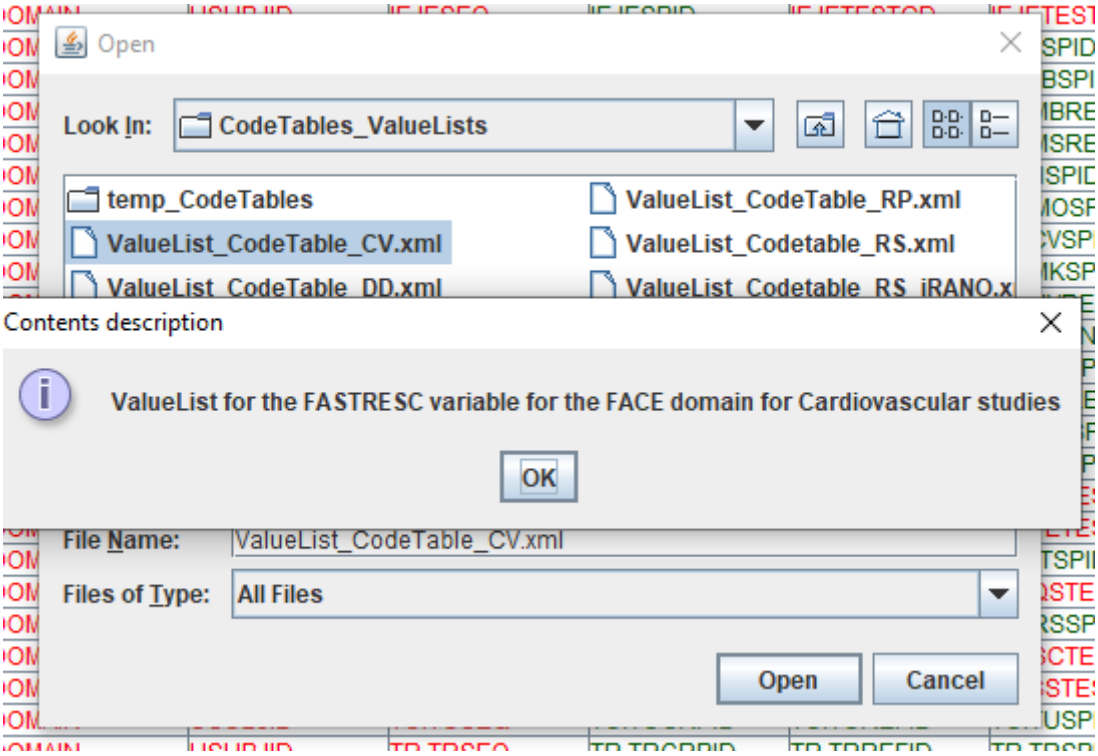
To load such a pre-defined ValueList, use the menu "Insert - ValueLists for CDISC Code Tables from File into define.xml":



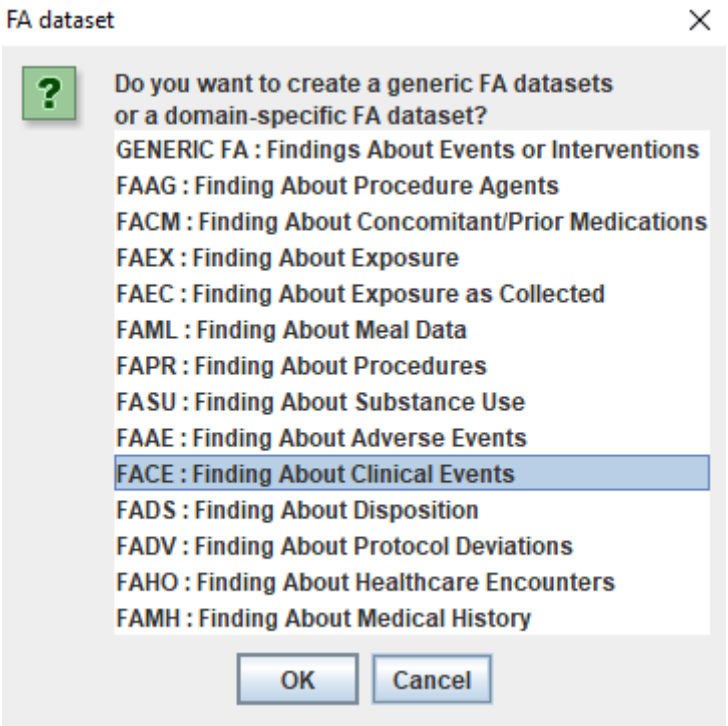
A list of available files will then be displayed:



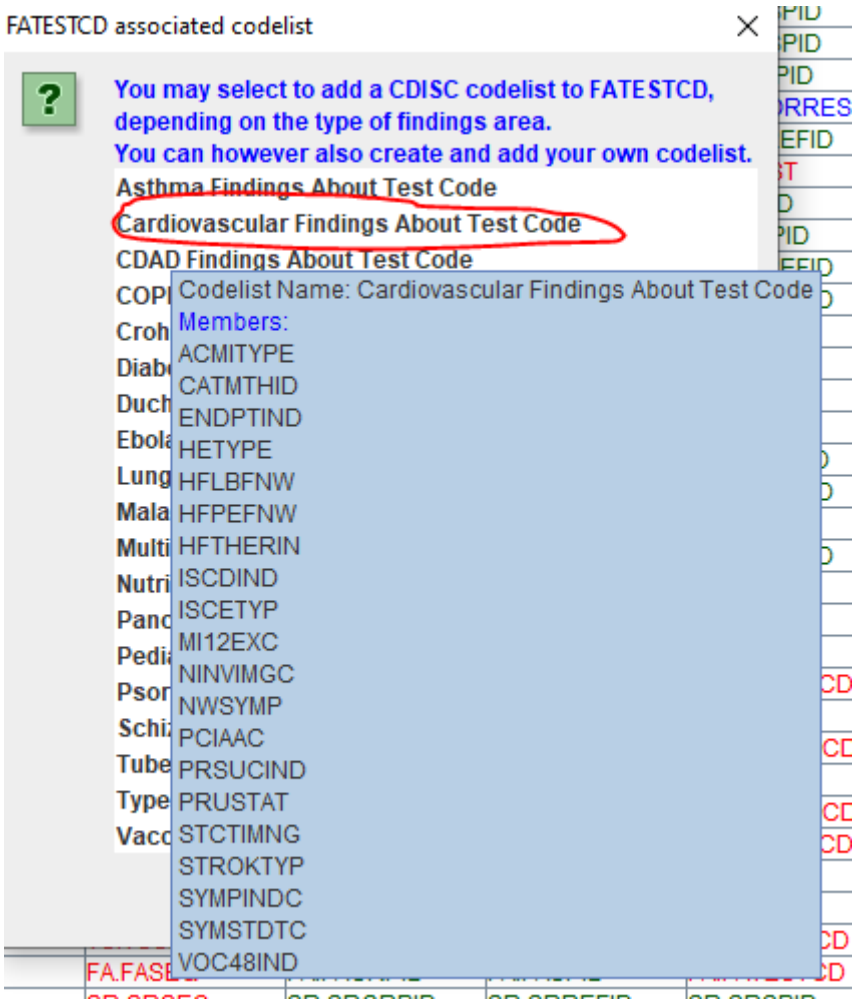
When one clicks on a file name, information about what the file is about is displayed. For example for "ValueList_CodeTable_CV":



So, this one is for FASTRESC for FACE (Findings about Clinical Events), for cardiovascular studies. Just for the exercise, let us create an FACE instance. We can do so by drag-and-drop from the "FA" row from the template, after which a dialog is displayed:



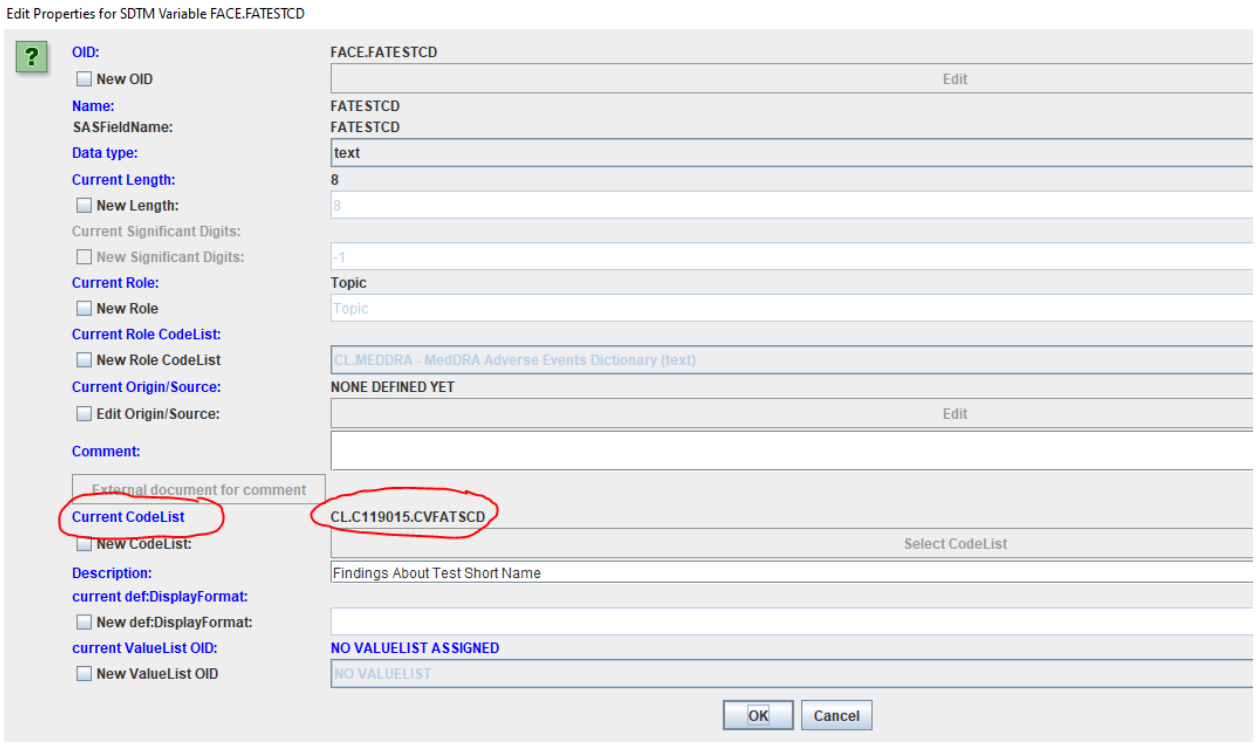
And select "FACE: Findings About Clinical Events".
The system then asks us what codelist we want to assign to FATESTCD. If we hover the mouse over "Cardiovascular Findings About Test Code", we get more details:



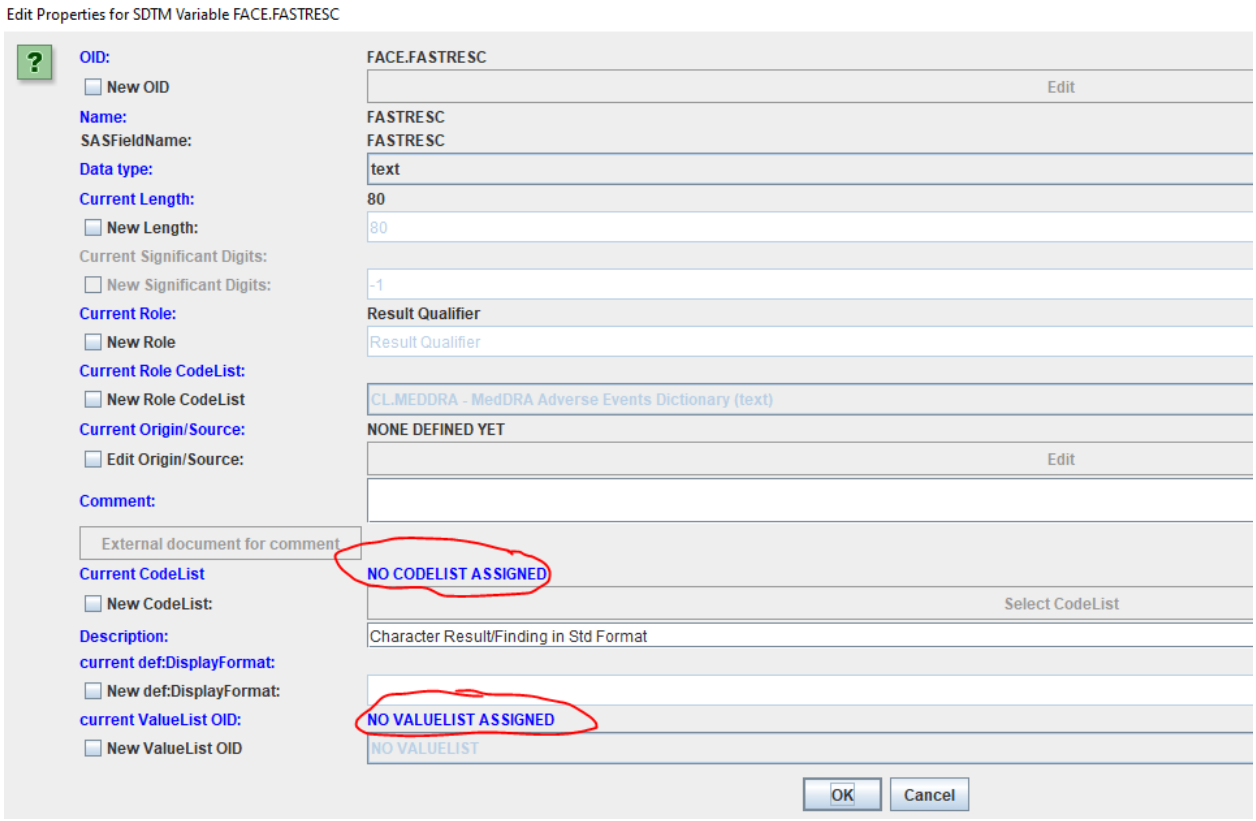
If we then select "Cardiovascular Findings About Test Code", we first get some advice about choosing a suitable structure for the dataset, and the "study-specific FACE instance" is created as a new row at the bottom:

TI	STUDYID	DOMAIN	TI.IETESTCD	TI.IETEST	TI.IECAT	TI.IESCAT	TI.TIRL	TI.TIVERS
TS	STUDYID	DOMAIN	TS.TSSEQ	TS.TSGRPID	TS.TSPARMCD	TS.TSPARM	TS.TSVAL	TS.TSVALNF
RELREC	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	RELTYPE	RELID	
SUPPQUAL	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	SUPPQUAL.QN...	SUPPQUAL.QL...	SUPPQUAL.QVAL
RELSUB	STUDYID	USUBJID	RELSUB.POOLID	RELSUB.RSUB...	RELSUB.SREL			
OI	STUDYID	DOMAIN	OI.NHOID	OI.OISEQ	OI.OIPARMCD	OI.OIPARM	OI.OIVAL	
CES:FACE	STUDYID	DOMAIN	USUBJID	FACE.FASEQ	FACE.FAGRPID	FACE.FASPID	FACE.FATESTCD	FACE.FATEST

If we ask for the properties of FATESTCD (using the menu "Edit - SDTM Variable Properties), we see that the codelist "CVFATSCD" has been assigned to FATESTCD:

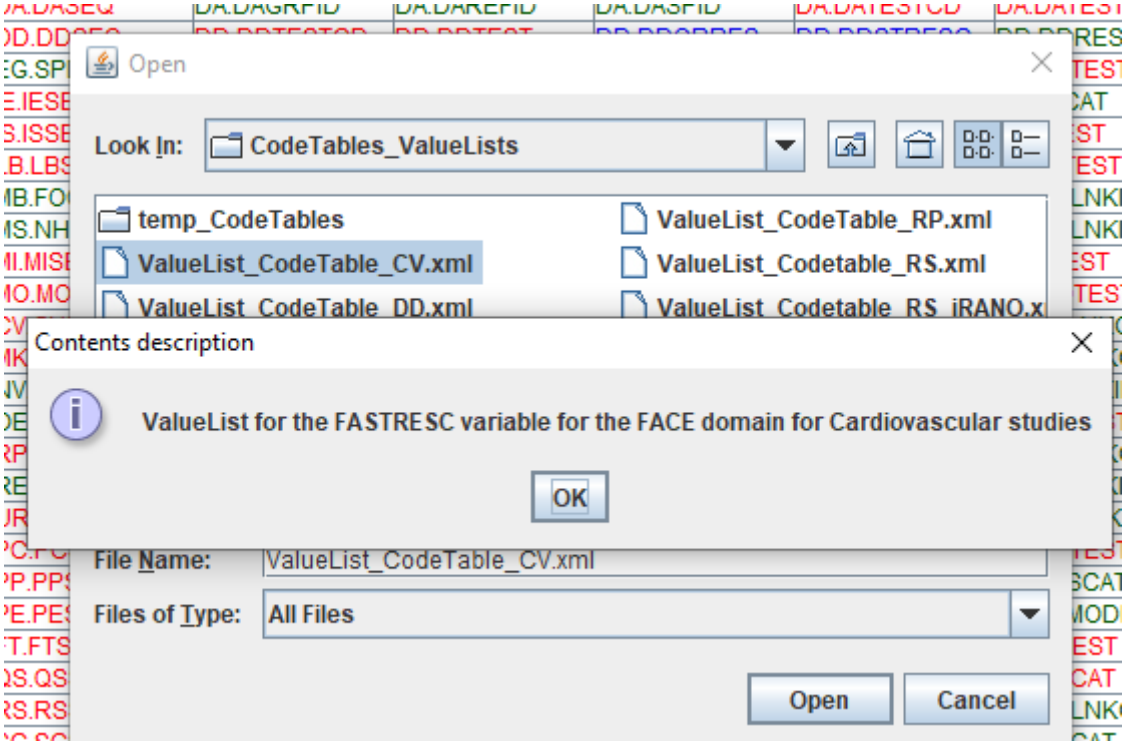


However, no codelist nor valuelist has yet been assigned to FASTRESC …

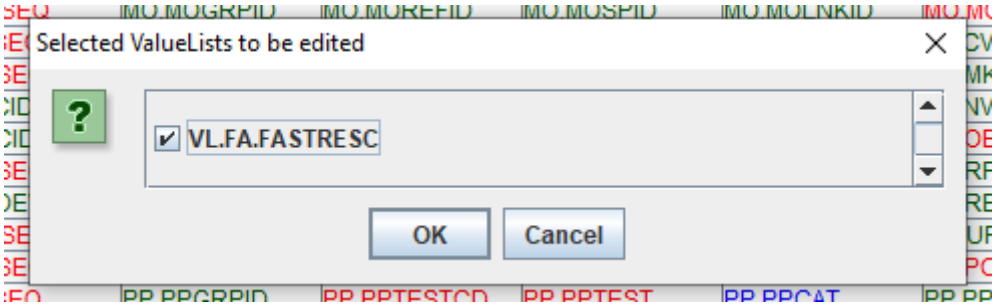


We don't however want to assign a codelist to FASTRESC however, but a valuelist, as the properties of FASTRESC will depend on the value of FATESTCD.

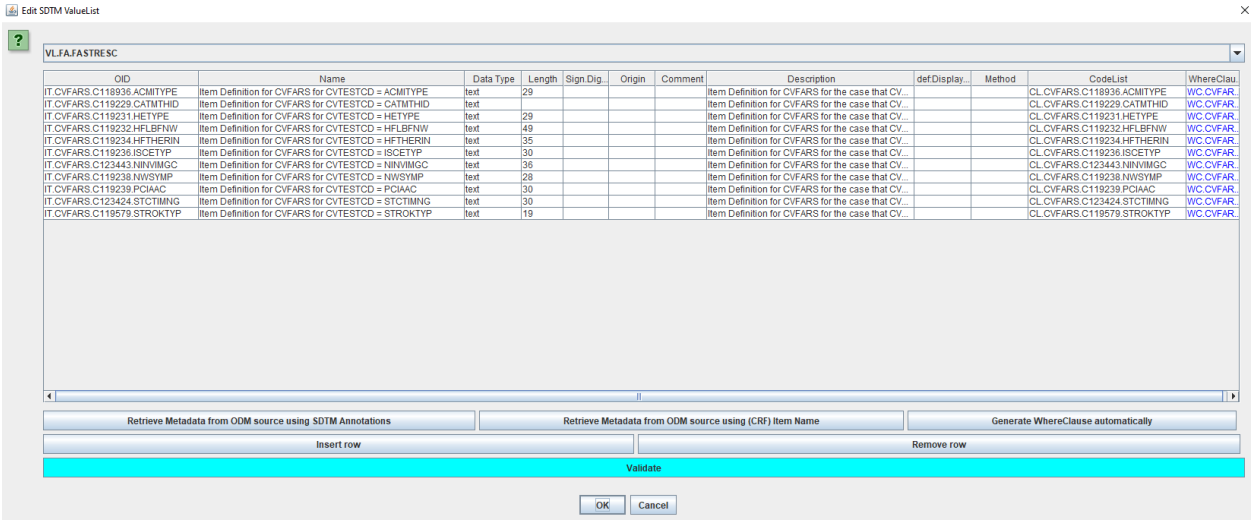
We can now import the ValueList (generated from the CDISC code tables) that we need by using the menu "Insert - ValueLists for CDISC Code Tables from File into define.xml", and select the "CV" one:



After selecting and clicking "OK" and "Open", the system will propose us to edit the imported ValueList:

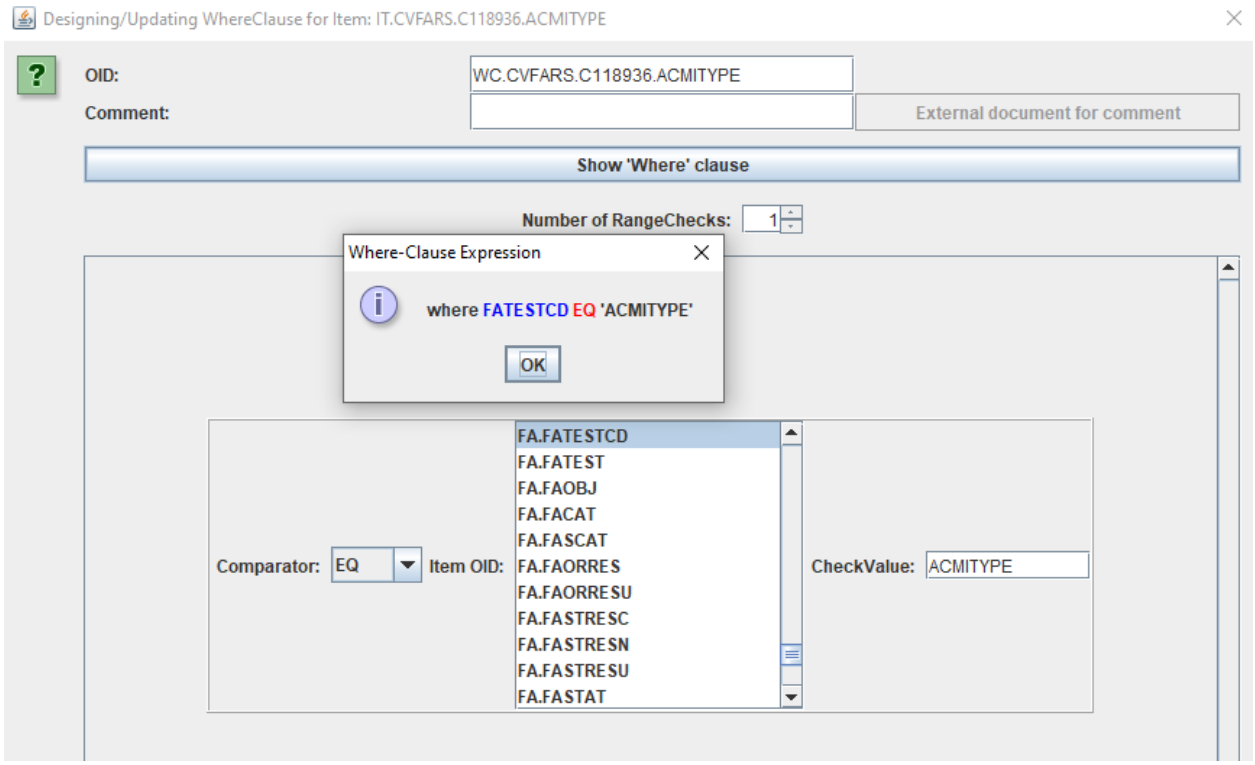


This is a good idea, as we may still want or need to make decisions:

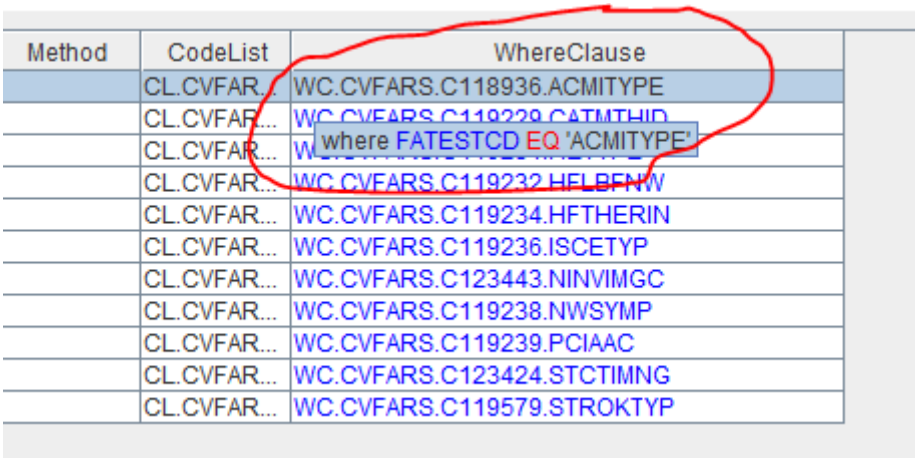


We may e.g. want to remove rows, as we never have the specific test code (in FATESTCD) used in

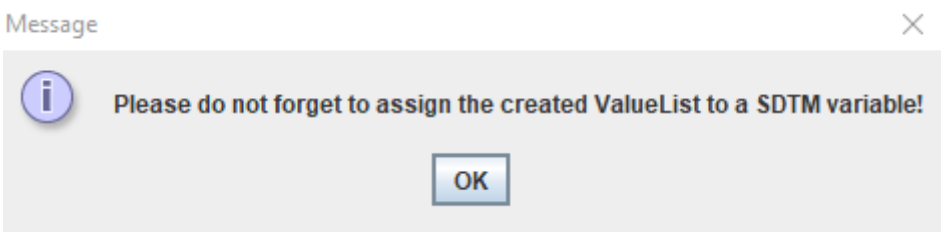
our study. We can also already assign a source and origin (like the CRF page number(s)), add comments, etc..
If we click on one of the "where clause" definitions (last column) and use the button "Show 'Where' clause", we e.g. get:



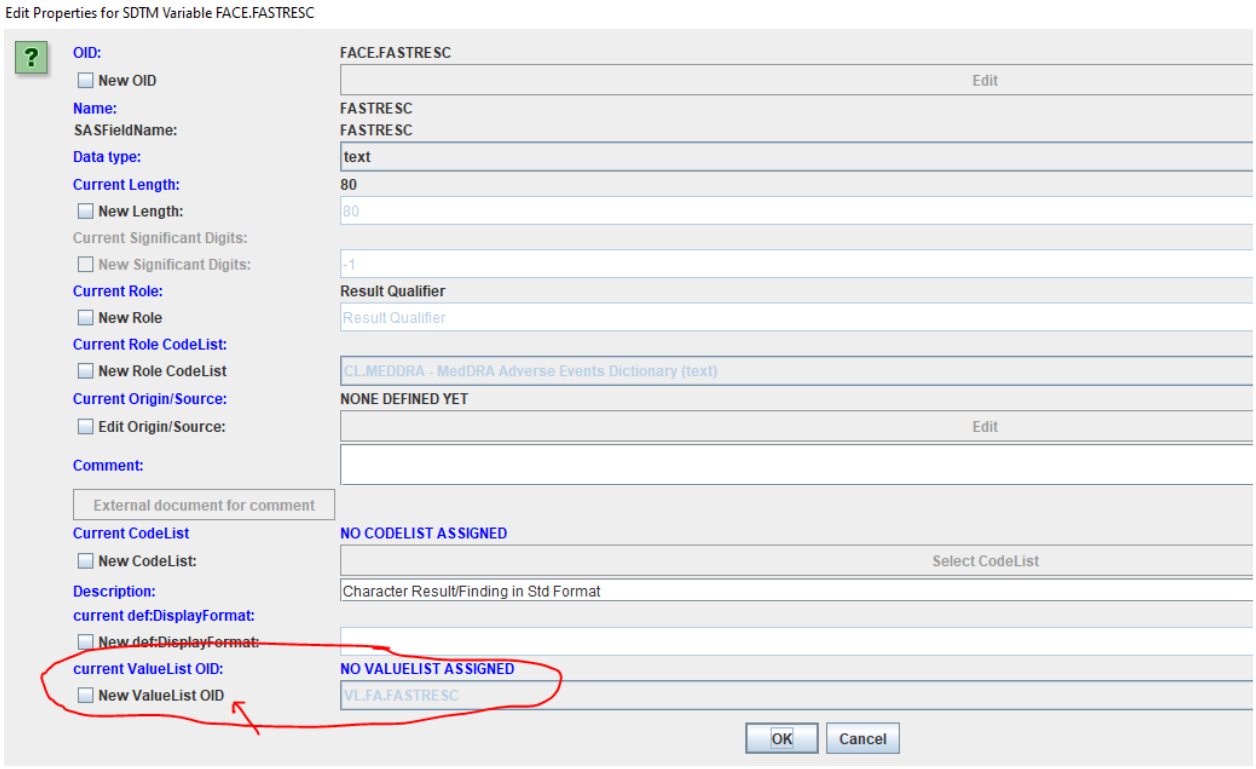
We can get the same information by just hovering the mouse over the "Where Clause" cell:



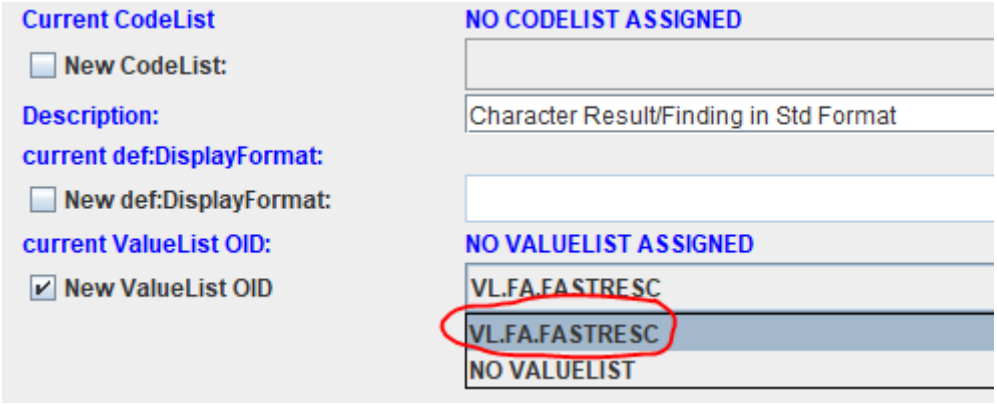
When we have adapted the ValueList for our purposes, we still need to assign this ValueList to the FASTRESC variable:



To do so, select it, and use the menu "Edit - SDTM Variable" (or use CTRL-E), leading to:



We now check the checkbox "New ValueList OID" and select the just created one from the dropdown:



After our "OK", we can have a look at the visualization of the define.xml, e.g. using the menu "View - View define.xml in browser", leading to:

FACE (Finding About Clinical Events) - FINDINGS ABOUT [SDTMIG 3.3]

Variable	Where Condition	Label / Description	Type	Role	Length or Display Format	Controlled Terms or ISO Format
STUDYID		Study Identifier	text	Identifier	80	
DOMAIN		Domain Abbreviation	text	Identifier	8	
USUBJID		Unique Subject Identifier	text	Identifier	60	
FASEQ		Sequence Number	integer	Identifier	80	
FAGRPID		Group ID	text	Identifier	80	
FASPID		Sponsor-Defined Identifier	text	Identifier	80	
FATESTCD		Findings About Test Short Name	text	Topic	8	Cardiovascular Findings About Test Code [20 Terms]
FATEST		Findings About Test Name	text	Synonym Qualifier	40	Cardiovascular Findings About Test Name [20 Terms]
FAOBJ		Object of the Observation	text	Record Qualifier	80	
FACAT		Category for Findings About	text	Grouping Qualifier	80	
FASCAT		Subcategory for Findings About	text	Grouping Qualifier	80	
FAORRES		Result or Finding in Original Units	text	Result Qualifier	80	
FAORRESU		Original Units	text	Variable Qualifier	80	Unit [886 Terms]
FASTRESC VLM		Character Result/Finding in Std Format	text	Result Qualifier	80	
	FATESTCD = "ACMITYPE"	Item Definition for CVFARS for the case that CVTESTCD = ACMITYPE	text		29	CodeList for CVFARS for the case that CVTESTCD = ACMITYPE [7 Terms]
	FATESTCD = "CATMTHID"	Item Definition for CVFARS for the case that CVTESTCD = CATMTHID	text		30	CodeList for CVFARS for the case that CVTESTCD = CATMTHID • "ANGIOGRAPHY" • "AUTOPSY"
	FATESTCD = "HETYPE"	Item Definition for CVFARS for the case that CVTESTCD = HETYPE	text		29	CodeList for CVFARS for the case that CVTESTCD = HETYPE • "HEART FAILURE HOSPITALIZATION" • "URGENT HEART FAILURE VISIT"