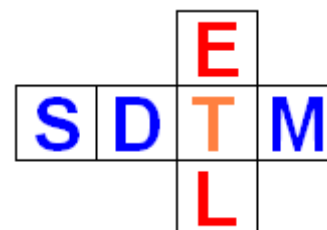


SDTM-ETL 5.1 **PREVIEW** : Summary of New Features

Author: Jozef Aerts, XML4Pharma

Last update: **2025-08-31**



Summary

This document contains a summary of the most important new features of SDTM-ETL 5.1 and bug fixes.

Table of Contents

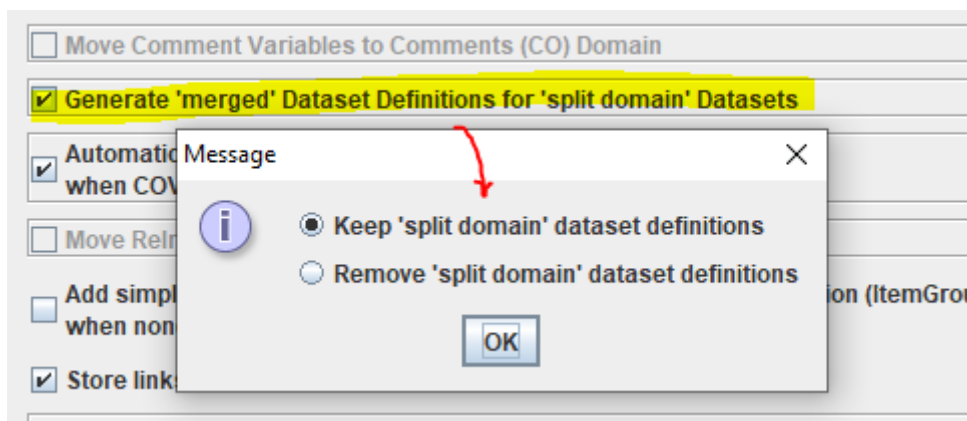
Save clean Define.xml.....	2
Insert "SUBJID" variable in other domains than DM.....	2
New mapping function "ucum2cdisc".....	4
New mapping function "studyday"	5
Improved handling of (very) large datasets.....	7
Removing several domain/dataset definitions at once	8
Extended CDISC CORE validation features - custom rules	10
CORE validation of existing datasets	16
New way of automated naming of different instances of the same study-specific domain	18
Improved dialogs when adapting the OID of the dataset definition in the case of "split domain" datasets.....	19
Improved handling of adding new inserted "non-standard Variables", "Variable for Comment" and "Variable for RELREC" in the case of "split-domain" datasets	21
Automated population of COREF in CO datasets	23
Automated population of COEVAL in CO datasets.....	24
New stylesheet for display of the define.xml (version 2.1) in the user's default browser	26
Removal of all Dataset-XML features	27
ODMSubjectRetriever.....	27
Other small changes and improvements.....	28
Color coding in the mapping script editor	28
Search button for "Insert - New SDTM/SEND Variable"	28
CORE validation Graphical User Interface improvements	30
"CDISC Notes" updates.....	31
Removing "COMMENT" and "RELREC" variables	31
New "Mapping script editor" features	32
Limitations to if-elsif-else structures.....	34
Bug fixes	35
QNAM-QLABEL match for Variables with more than 400 characters	35
QLABEL in batch execution when also requesting to merge "split domain" datasets	36
Merging define.xml-s with mappings with the option to only load the new dataset definitions that were not present yet.....	36
Fixes for Medical Devices domains DI and DT	36
Other small fixes.....	37
Other remarks.....	37

Save clean Define.xml

When using the menu "Save - Clean define.xml", there is now a new checkbox "Generate 'merged' Dataset Definitions for 'split domain' Datasets":

This checkbox becomes only available when the system detects that there are "split domain" dataset definitions for the study (like LBCH, LBBL, ...).

If the checkbox is checked, the "cleaned" define.xml file will then also contain a dataset definition for a 'merged' dataset (like LB). It is also asked whether the dataset definitions for the 'split domain' datasets must be kept or must be removed.



This will also take care that a "merged" dataset definition is generated for SUPPxx datasets, e.g. when both LBCH as LBBL have supplemental qualifiers, a dataset definition for an additional "SUPPLB" will be generated on top of a SUPPLBCH and SUPPLBBL dataset definition (when "Keep 'split domain' dataset definitions" is selected).

The user can then use the result cleaned define.xml as the basis for further "fine tuning" the define.xml for a regulatory submission (every regulatory authority has its own requirements for the define.xml) e.g. using our ["Define-XML Designer"](#).

Insert "SUBJID" variable in other domains than DM

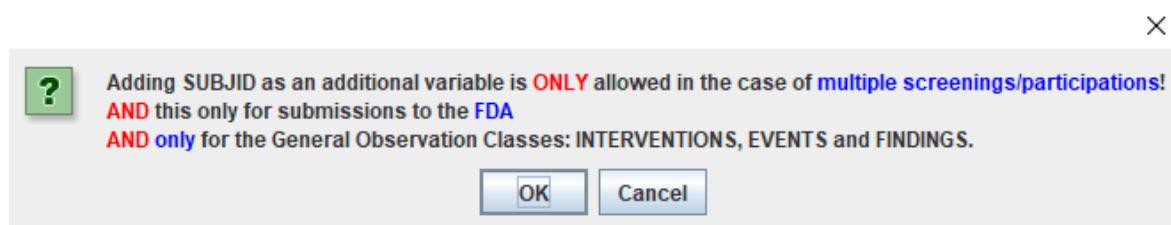
Although this is essentially crazy, some regulatory authorities, such as FDA, may require to have an additional "SUBJID" (the local subject identifier) in other domains than DM. The idea is that some reviewers want to have this information when there are multiple screenings or participations.

Essentially, having SUBJID in other domains is strictly forbidden by the SDTM-IG, but as one knows, some regulatory authorities deviate from the SDTM-IG rules ...

In order to add a "SUBJID" variable to other domains than DM, we added a new menu "Insert - SUBJID variable for multiple screenings/participations".

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 or External CodeList					
Create new SDTM CodeList from MeasurementUnits					
Create new ValueList from existing CodeList					
Create mapping formula					
Sponsor defined SDTM Domain					
Domain-specific SUPPQUAL					
Associated Persons Domain					
Associated Persons Related to Subjects (APREL SUB) domain					
Global Subject Variables Domain					
New SDTM Variable					
New non-standard SDTM Variable for SUPPQUAL					
SUBJID variable for multiple screenings/participations					
New SDTM Variable for COMMENT					
New SDTM Variable for RELREC					
Link to Annotated CRF					
Link to Supplemental Doc					
CRF Page Numbers to Variable Origin					

For example, when we already started the mapping work for DM and LB, and then select a cell in the LB row, and then use the menu, this will result in a dialog:



and when one then clicks "OK" in:

RELREC	STUDYID	RDOMAIN	USUBJID	IDVAR	define.xml information:
RELSPEC	STUDYID	USUBJID	REFID	SPEC	SDTM Name: SUBJID
RELSUB	STUDYID	USUBJID	POOLID	RSUBJID	OID: SUBJID
SUPPQUAL	STUDYID	RDOMAIN	USUBJID	IDVAR	Mandatory: No
CM4620-203:DM	STUDYID	DOMAIN	USUBJID	SUBJID	OrderNumber: 4
CM4620-203:LB	STUDYID	DOMAIN	USUBJID	SUBJID	Role: Identifier
					Data type: text
					Length: 80
					Description: Subject Identifier for the Study

with a "SUBJID" variable immediately after the "USUBJID" variable.
One can then map "SUBJID" in the same way as for DM.

Please remark that having SUBJID in other domains than DM will certainly lead to validation errors in most validation tools.

New mapping function "ucum2cdisc"

CDISC still stubbornly refuses to allow UCUM units in SDTM. Even worse, it mandates that for synonyms of units, only one of them may be used.

For example, CDISC does not allow the use of "mg/mL" but mandates the use of the synonymous "g/L". The reasoning looks to be that CDISC supposes that reviewers are incapable of recognizing that "mg/mL" and "g/L" are the same thing...

According to CDISC:

www.cdisc.org/kb/articles/ucum-and-cdisc-codelists

is that a computer can read a string of characters and determine whether that character string is a valid expression in the nomenclature as well as determine the meaning of valid expressions.

- A codelist is, as the name implies, a list of codes (terms) with definitions of what they mean. For example, "g/L" is a term in the CDISC "UNIT" codelist. In the "UNIT" codelist, as in all CDISC codelists, each term has its own definition, as well as an NCI thesaurus code (often referred to as a C-code), a CDISC submission value, and synonyms.

A string of characters that is a valid UCUM expression may not be present in the CDISC "UNIT" codelist for various reasons:

- CDISC has not received a request for the concept represented by the expression.
- Many unit expressions are mathematically synonymous, but the CDISC "UNIT" codelist includes only one of those synonymous expressions as a submission value. For example, the CDISC "UNIT" codelist includes the submission value "g/L" but does not include mathematically synonymous terms such as "mg/mL" or "ug/uL", as separate entries in the codelist, both of which are also valid UCUM expressions. **This restriction to a single submission value is intended to make life easier for reviewers, so that they always see the same expression, and don't have to mentally translate between mathematically synonymous expressions.**
- To make UCUM expressions unambiguous and to deal with a variety of units outside the SI system, UCUM uses symbols which are unfamiliar to most lay users. The CDISC codelist is intended for use by a broad audience and therefore uses expressions without these special and potentially confusing symbols. For example:
 - Most non-SI units, such as the "imperial" units used in the US, are represented by character strings which includes suffixes and are enclosed in square brackets. For example, the representation of "inch" is "[in_i]", where "_i" indicates the imperial system of measurement.
 - It is common to see, as part of the representation of a unit, text enclosed in curly brackets that UCUM considers annotations. For example, what is represented with a submission value of "ELISA unit/dose" in the CDISC "UNIT" codelist is represented as "[ELU]/{dose}" in UCUM.

Mapping between UCUM and CDISC codelist representations of units may be facilitated by the [Unit-UCUM Code table](#). This code table includes all CDISC unit of measure codelists along with UCUM representations for each unit.

The table below compares these unit code systems at a high-level.

UCUM	CDISC UNIT Codelists
Nomenclature for constructing machine-readable unit representations from a set of basic building blocks.	A codelist of unit representations
Contains mathematically equivalent representations of a unit (e.g., g/l, mg/ml)	Includes only one representation of each unit, (e.g. g/l, but not mg/ml)

In the age of computers, this is really crazy ...

But it is as it is, so we needed a function that e.g. takes care that when e.g. a concentration was collected in "mg/mL" units, this can automatically be converted to "g/L".

This function is named "ucum2cdisc()" and takes a single parameter which is the collected unit. It can be selected in the "mapping editor", e.g.:

The Transformation Script

```

5 $CODEDVALUE = $pdv(studyeventdata/codedata[@ucumunit='g/dL']/itemscroupdata[@itemscroupunit='10_uL_U/L']) // 10
6 if ($CODEDVALUE == 'g/dL') {
7   $NEWCODEDVALUE = 'g/dL';
8 } elseif ($CODEDVALUE == 'mg/dL') {
9   $NEWCODEDVALUE = 'mg/dL';
10 } elseif ($CODEDVALUE == '10*3/uL') {
11   $NEWCODEDVALUE = ucum2cdiscct($CODEDVALUE);
12 } else {
13   $NEWCODEDVALUE = 'NULL';
14 }
15 $LB.LBORRESU = $NEWCODEDVALUE;

```

Scripting Language Functions

contains	starts-with	ends-with	matches	not		
abs	sqrt	log	log10	exp	exp10	
min	max	avg	sum	count	is-a-number	ucum2cdiscct
ceiling	floor	round	modulus	number	string	provides the CDISC-CT unit for the provided UCUM unit
date	year	month-in-year	day-in-year	day-in-month	day-in-week	

AT QS.QSORRES QS.QSC
 RP.RPSCAT RP.RPC
 SC.SCORRES SC.SCC
 SS.SSORRES SS.SSS
 TU.TUORRES TU.TUS
 TR.TRTEST TR.TRO
 RS.RSTEST RS.RSC
 VS.VSPOS VS.VSOI
 FA.FASCAT FA.FAOF
 SR.SRCAT SR.SRS
 DTTC DM.DTHOTC DM.DTH
 LB.LBSCAT LB.LBOI
 LB.LBSCAT LB.LBOI

taking care that the collected value "10*3/uL" (UCUM notation) translated into 10⁹/L" (CDISC unit).

The function has been implemented in the "functions.xsl" file under "stylesheets" and can thus easily be extended:

```

<xsl:function name="sdtm-etl:ucum2cdiscct">
  <xsl:param name="ucumunit"/>
  <xsl:choose>
    <xsl:when test="$ucumunit = 'mg/mL'">g/L</xsl:when>
    <xsl:when test="$ucumunit = 'ng/mL'">ug/L</xsl:when>
    <xsl:when test="$ucumunit = 'pg/mL'">ng/L</xsl:when>
    <!-- Typical UCUM units for erythrocytes and leukocytes and basophils -->
    <xsl:when test="$ucumunit = '10*6/uL'">1012/L</xsl:when>
    <xsl:when test="$ucumunit = '10*3/uL'">109/L</xsl:when>
    <!-- UCUM uses "*" for exponent, whereas CDISC uses Excel (!) notation -->
    <xsl:when test="contains($ucumunit, '10*')">
      <xsl:value-of select="replace($ucumunit, '10*', '10^')"/>
    </xsl:when>
    <!-- nothing to change -->
    <xsl:otherwise><xsl:value-of select="$ucumunit"/></xsl:otherwise>
  </xsl:choose>
</xsl:function>

```

New mapping function "studyday"

Essentially, the calculation of the "study day" (-DY variables) is very simple: one uses the function "datediff" and then adds "1" in case the value is non-negative. E.g.:

Remark that there is no need in this script to also define the variable \$LB.LBDTC, as the latter comes before (the script for) LBDY, and can thus be reused.

The function "studyday" has been defined in such a way that in case one of the two arguments does not at least have a complete date part (e.g. "2024-12"), the empty string is returned.

Improved handling of (very) large datasets

In version 5.0, we added a new method to cope with very large datasets of clinical data, as this can lead to memory problems "out-of-memory error".

The software then calculates the ratio between available memory (which can be set using the "-Xmx" parameter in the SDTM-ETL.bat file) and the size of the file with clinical data. If this ratio drops between 10, then the file with clinical data is split by subject, and the single subject files (stored in the folder "tempWorkDir") are used to generate the SDTM (or SEND) files.

The generation of the single subject files with clinical data however also takes some time. Therefore, a customer asked to implement features to have more control over the behavior when the file with clinical data is split into single subject files or not.

This has now been realized in the following way:

- the parameter for the ratio between available memory and file size (default 10) can be set in the "properties.dat" file using the parameter "thresholdmemoryfilesizeratioforsubjectsplitsplit".

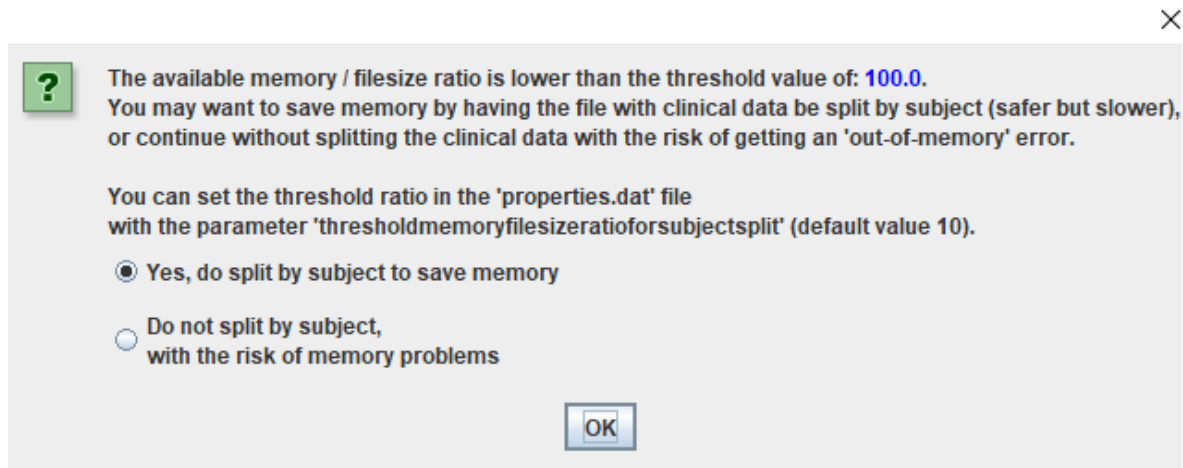
For example:

```
# memory/filesize ratio for splitting clinical data by subject
thresholdmemoryfilesizeratioforsubjectsplitsplit=1000
```

In order to have "splitting" for smaller files too, set the value for the parameter to a higher value than 10, e.g. "1000" or "100". In order to request not to split the file with clinical data even with (very) large datasets, set the value of the parameter lower than the default value of 10.

Remark that in the latter case, the transformation may fail with an "out-of-memory" error.

- At the start of the execution of the mappings, the system checks the available memory and the file size, and if the value of the ratio drops below the value of the "thresholdmemoryfilesizeratioforsubjectsplitsplit" parameter provided, it will ask whether splitting by subject must be done first, using a dialog. For example when the user has set a value of 100 for the parameter:



The user can then still decide ...

The optimal value of the parameter for splitting or not splitting by subject is very hard to determine as it depends on many factors, such as the number of domains, the number of subjects and amount of data per subject, and the properties of the computer, such as "memory caching". Therefore it may require some testing to find the optimal value for the parameter and thus the equilibrium between speed and memory usage.

Removing several domain/dataset definitions at once

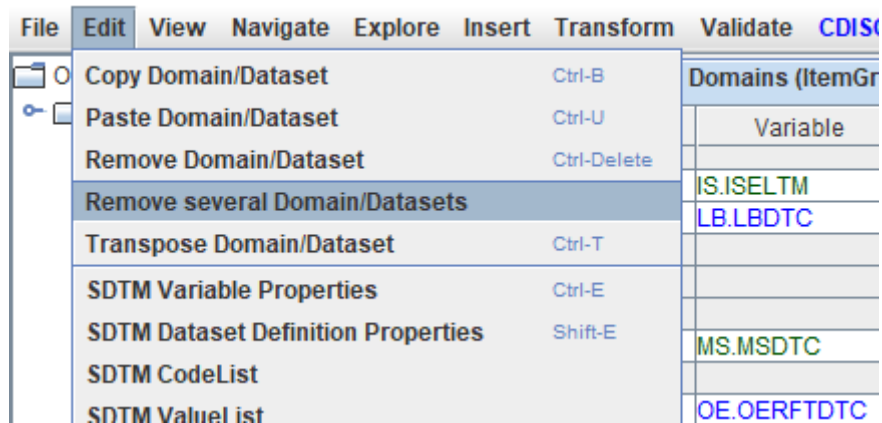
When needing to update mappings, it occurs that one has loaded a define.xml that already contains a lot of domain/dataset definitions, and that one wants to remove most of them in order to concentrate on just one or a few of them. The updated one can then later be re-merged with the existing ones.

The typical way to remove a domain/dataset definition is by doing a right-click in a cell of the applicable row, which leads to a confirmation dialog like:

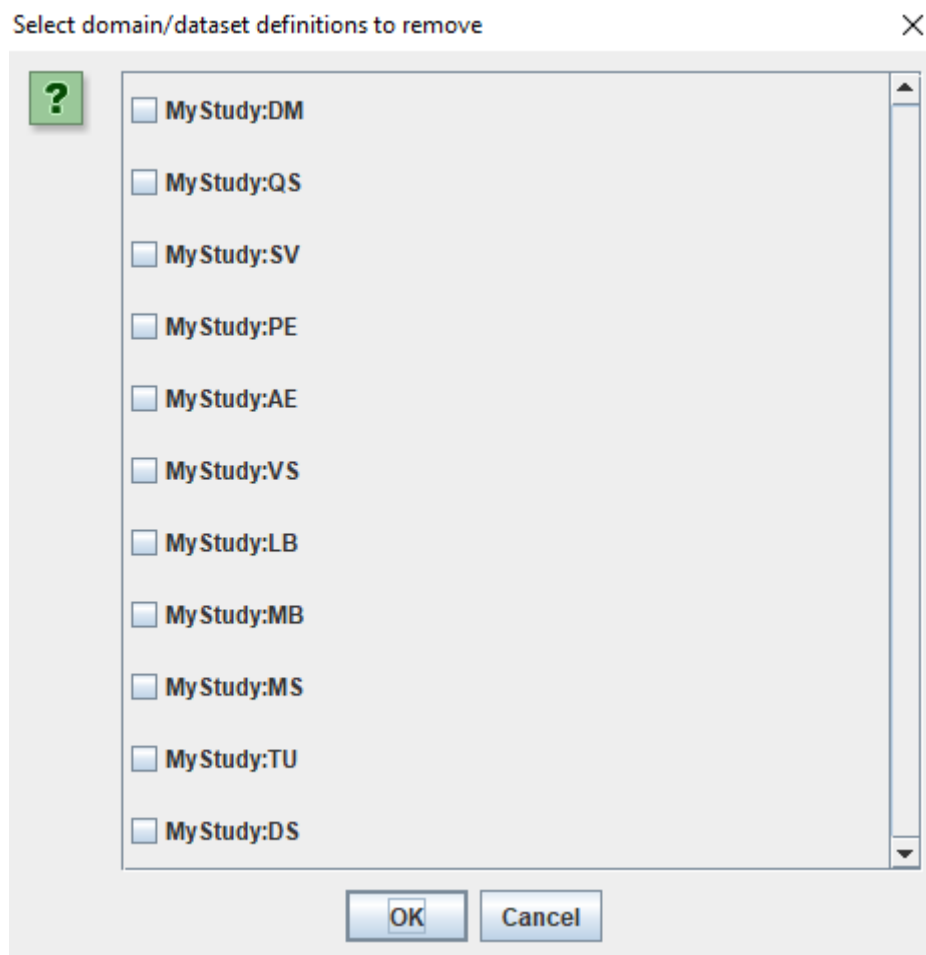


After clicking "Yes", the underlying define.xml structure is updated, which takes a little bit of time. If this has to be repeated for a larger number of domains, this may take considerable time. Therefore, one of our users asked to have a feature that allows to remove several domain/dataset definitions in one run.

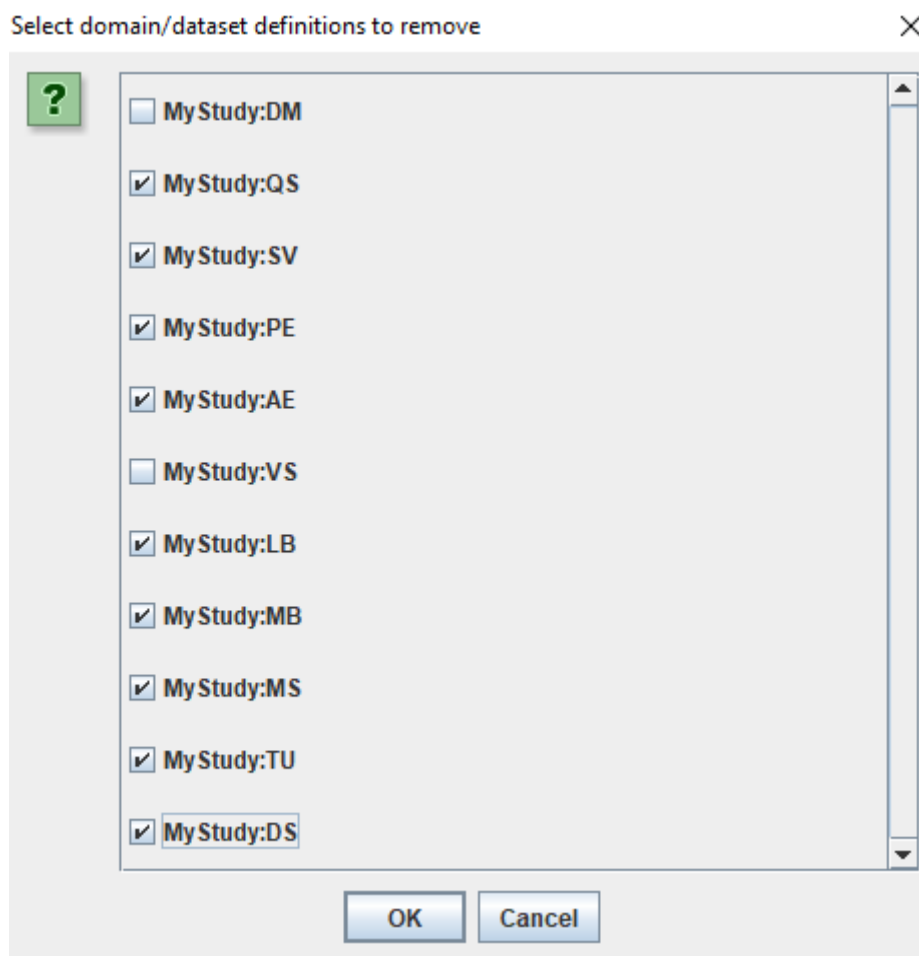
So, we added a new menu "Edit - Remove several Domain/Datasets":



The system then presents the list of all study-specific domain instances, like:



and when one e.g. then only wants to continue working on the mappings for DM and VS, one checks all the checkboxes except the ones for DM and VS:



All the "checked" domain/dataset definitions are then removed at once, and the underlying define.xml is then updated, leading to the result:

RS	STUDYID	DOMAIN	USUBJID	RS.RSSEQ	RS.RSGRPID
VS	STUDYID	DOMAIN	USUBJID	VS.VSSEQ	VS.VSGRPID
FA	STUDYID	DOMAIN	USUBJID	FA.FASEQ	FA.FAGRPID
SR	STUDYID	DOMAIN	USUBJID	SR.SRSEQ	SR.SRGRPID
RELREC	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL
SUPPQUAL	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL
My Study:DM	STUDYID	DOMAIN	USUBJID	SUBJID	DM.RFSTDTC
My Study:VS	STUDYID	DOMAIN	USUBJID	VS.VSSEQ	VS.VSGRPID

Extended CDISC CORE validation features - custom rules

CDISC CORE validation is rapidly replacing SDTM/SEND validation of Pinnacle21 (known for its many "false positives"). One of the reasons is that CORE allows sponsors and providers to develop their own set of rules, which is extremely interesting for quality assurance.

One such "CORE Rules Extensions" has been developed by us and is based on the "[CDISC Dataset Specializations](#)" which essentially are "Biomedical Concepts", but with additional information. These "Dataset specializations" are available as an [Excel worksheet](#), but can also

be retrieved from the ["CDISC Library" using the API](#). CDISC also developed a "browser" which is nice for inspection but worthless for use in any form of automation.

Essentially, these "Dataset Specializations" describe things like "when the test is 'Diastolic Blood Pressure', the expected unit (in VSORRESU/VSSTRESU) is 'mmHg' and the expected values for the (body) location of the measurement are 'BRACHIAL ARTERY', 'CAROTID ARTERY', 'DORSALIS PEDIS ARTERY', 'FEMORAL ARTERY', 'FINGER', 'PERIPHERAL ARTERY', or 'RADIAL ARTERY', and the expected values for laterality are 'LEFT' and 'RIGHT'".

So one can derive "rules" from these Dataset Specializations ... especially for quality assurance. Remark that the "normal" CORE rules nor Pinnacle21 can do this: for example, when one has VSTESTCD=DIABP and VSORRESU=cm, nor CORE nor Pinnacle21 will report this as a violation, as "cm" is a valid value of the codelist for VSORRESU.

We therefore developed some software (it essentially is just a small set of Java classes) to automatically retrieve all Dataset Specializations from the CDISC Library and automatically generate CORE rules from them. This led to 1789 different new rules which were then added to the CORE engine that comes with SDTM-ETL 5.1.

Also new is that one can now its own (e.g. company-specific) CORE rules. These rules are usually rules developed for quality assurance that goes beyond validation against the SDTM/SEND standard, and can be study- or sponsor-specific. For example, one may have a quality assurance rule that subjects may not be older than 65 years of age.

The rule, as developed in YAML, then e.g. is:

```

1  Authorities:
2  - Organization: XML4Pharma
3  Standards:
4  - Name: SDTMIG
5  References:
6  - Citations:
7    - Cited Guidance: Study protocol p. 20
8      Document: Study protocol
9      Origin: SDTM and SDTMIG Conformance Rules
10 Rule Identifier:
11   Id: XML4P1
12   Version: '1'
13   Version: '2.0'
14   Version: '3.3'
15 Check:
16 all:
17   # throw an error when AGE (in years) is higher than 65
18   - name: AGEU
19     operator: equal_to
20     value: YEARS
21   - name: AGE
22     operator: greater_than
23     value: 65
24
25 Core:
26   Id: XML4P1
27   Status: Draft
28   Version: '1'
29   Description: 'Age in years must be 65 or lower'
30   Executability: Fully Executable
31 Outcome:
32   Message: 'Age in years must be 65 or lower'
33   Rule Type: Record Data
34 Scope:
35   Classes:
36   Include:
37     - SPECIAL PURPOSE
38   Domains:
39   Include:
40     - DM
41   Sensitivity: Record

```

In this case limited to SDTMIG-3.3.

Such rules are easy to develop once one has understood the syntax, with all functions documented at: <https://cdisc-org.github.io/conformance-rules-editor/#/>.

If you do not have the necessary knowledge within your company, please [contact us](#) - we are also providing CORE consultancy services.

When then doing an SDTM generation, and check the checkbox "Perform CDISC CORE validation ...":

☐ Perform post-processing for assigning --LOBXFL
 ☐ Perform post-processing unscheduled VISITNUM
☒ Split records > 200 characters to SUPP-- records
☒ Move non-standard SDTM Variables to SUPP--
☐ Move Relrec Variables to Related Records (RELREC) domain
☐ Move Comment Variables to Comments (CO) Domain
☒ View Result SDTM tables
☐ Try to generate 1:N RELREC Relationships
☐ Generate 'NOT DONE' records for QS datasets
☒ Adapt Variable Length for longest result value
☒ Unique --SEQ values across 'split' domains
☐ Re-sort records using define.xml keys
☒ Save Result SDTM tables as:
☒ Perform CDISC CORE validation on generated SDTM files

☐ Dataset-JSON 1.1
 ☒ SAS-XPT
 ☐ UTF-8 encoded CSV
 ☐ SQL INSERT statements

SDTM export files directory:

D:\temp Browse...

☐ Add location of generated SDTM files to define.xml
 ☐ Store link as relative path

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

Messages and error messages:

When then all datasets are generated (either in SAS-XPT format or Dataset-JSON 1.1 format), a dialog is displayed:

☐ Generate 'NOT DONE' records for QS datasets
 ☐ Re-sort records using define.xml keys
☒ Unique --SEQ values across 'split' domains
 ☒ Perform CDISC CORE validation on generated SDTM files
 result SDTM tables as:
☐ Dataset-JSON 1.1
 ☒ SAS-XPT
 ☐ UTF-8 encoded CSV
 ☐ SQL INSERT statements
 port files directory:
 ation of generated SDTM files to define.xml
 nally generate a merged dataset for 'split' domain datasets
 and error messages:

id value for variable LBSPEC for LOINC code 704-7
 iTful web service to find LBSPEC value for LOINC code 704-7
 BSPEC for LOINC code 704-7 = BLOOD
 id value for variable LBMETHOD for LOINC code 704-7
 iTful web service to find LBMETHOD value for LOINC code 704-7
 BMETHOD for LOINC code 704-7 = AUTOMATED COUNT

CORE rulesets

☒ CDISC Ruleset

☐ CDISC Dataset Specializations Ruleset (custom rules)

☐ Own custom data quality rule(s)

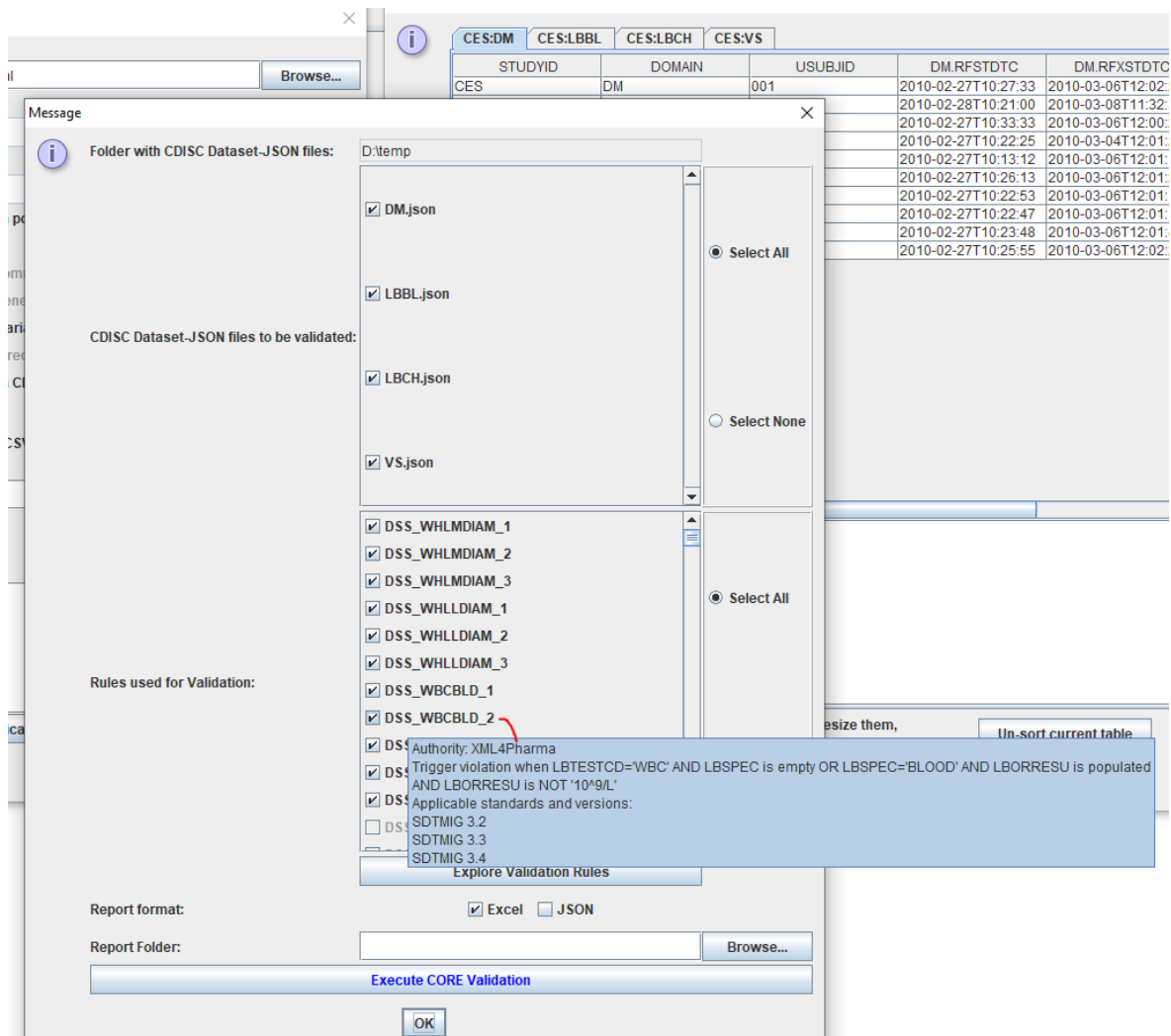
OK

asking whether one want to do "normal" CORE validation (using the CDISC ruleset) or do validation using the CORE rules generated from the "Dataset Specializations" or to apply oen or more of the companies own rules².

When one then selects "CDISC Dataset Specializations", a new dialog shows up allowing to select on which of the generated SDTM files one wants to perform the validation, and even select which rules need to be included and which must be excluded³:

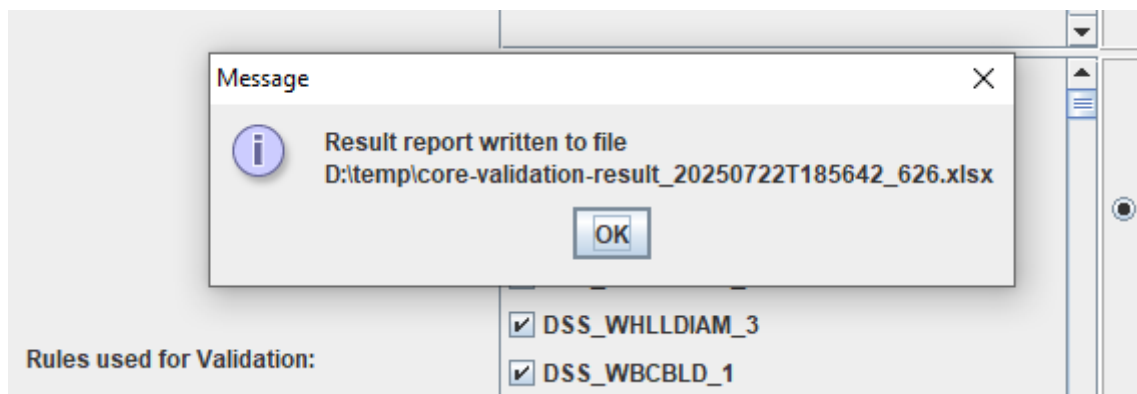
² At this moment, the CORE engine does not allow to do both or all three of them in one run, e.g. combined.

³ This is also that is not possible with Pinnacle21, there it always is "all or nothing"



Rules that are not applicable to the used SDTMIG version are automatically blended out and will then be skipped automatically.

When then one provides the folder where the report file (choice between Excel and JSON format) must be generated and click "Execute CORE Validation", the CORE validation process starts, and after about a minute, a message is provided that the validation has been finalized, e.g.:



When one then opens the Excel validation report file, one e.g. finds:

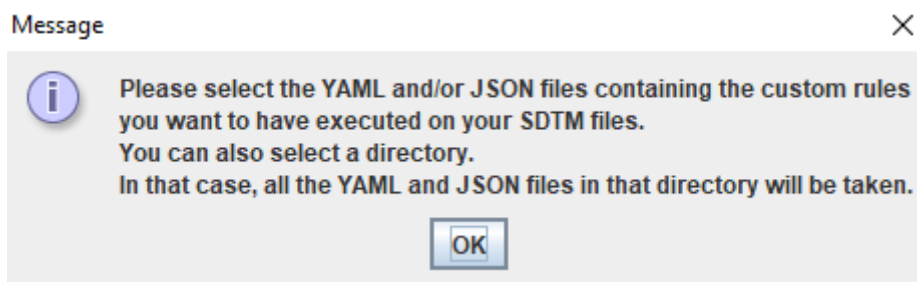
core-validation-result_20...									
A	B	C	D	E	F	G	H	I	
CORE-ID	Message	Executability	Dataset	USUBJID	Record	Sequence	Variable(s)	Value(s)	
2	DSS_WEIGHT_2 For VSTESTCD=WEIGHT, VSORRESU is expected to be one of: 'LB','g','kg'	fully executable	VS.json	001		1	VSORRESU, VSTESTCD	cm, WEIGHT	
3	DSS_WEIGHT_2 For VSTESTCD=WEIGHT, VSORRESU is expected to be one of: 'LB','g','kg'	fully executable	VS.json	002		12	VSORRESU, VSTESTCD	cm, WEIGHT	
4	DSS_WEIGHT_2 For VSTESTCD=WEIGHT, VSORRESU is expected to be one of: 'LB','g','kg'	fully executable	VS.json	003		23	VSORRESU, VSTESTCD	cm, WEIGHT	
5	DSS_WEIGHT_2 For VSTESTCD=WEIGHT, VSORRESU is expected to be one of: 'LB','g','kg'	fully executable	VS.json	004		34	VSORRESU, VSTESTCD	cm, WEIGHT	
6	DSS_WEIGHT_2 For VSTESTCD=WEIGHT, VSORRESU is expected to be one of: 'LB','g','kg'	fully executable	VS.json	005		45	VSORRESU, VSTESTCD	cm, WEIGHT	
7	DSS_WEIGHT_2 For VSTESTCD=WEIGHT, VSORRESU is expected to be one of: 'LB','g','kg'	fully executable	VS.json	006		56	VSORRESU, VSTESTCD	cm, WEIGHT	
8	DSS_WEIGHT_2 For VSTESTCD=WEIGHT, VSORRESU is expected to be one of: 'LB','g','kg'	fully executable	VS.json	007		67	VSORRESU, VSTESTCD	cm, WEIGHT	
9	DSS_WEIGHT_2 For VSTESTCD=WEIGHT, VSORRESU is expected to be one of: 'LB','g','kg'	fully executable	VS.json	008		78	VSORRESU, VSTESTCD	cm, WEIGHT	
10	DSS_WEIGHT_2 For VSTESTCD=WEIGHT, VSORRESU is expected to be one of: 'LB','g','kg'	fully executable	VS.json	009		89	VSORRESU, VSTESTCD	cm, WEIGHT	
11	DSS_WEIGHT_2 For VSTESTCD=WEIGHT, VSORRESU is expected to be one of: 'LB','g','kg'	fully executable	VS.json	010		100	VSORRESU, VSTESTCD	cm, WEIGHT	
12									
13									
14									

stating that for some records with VSTESTCD=WEIGHT, an incorrect value "cm" for VSORRESU has been assigned.

This kind of mapping errors is otherwise not so easy to detect ...

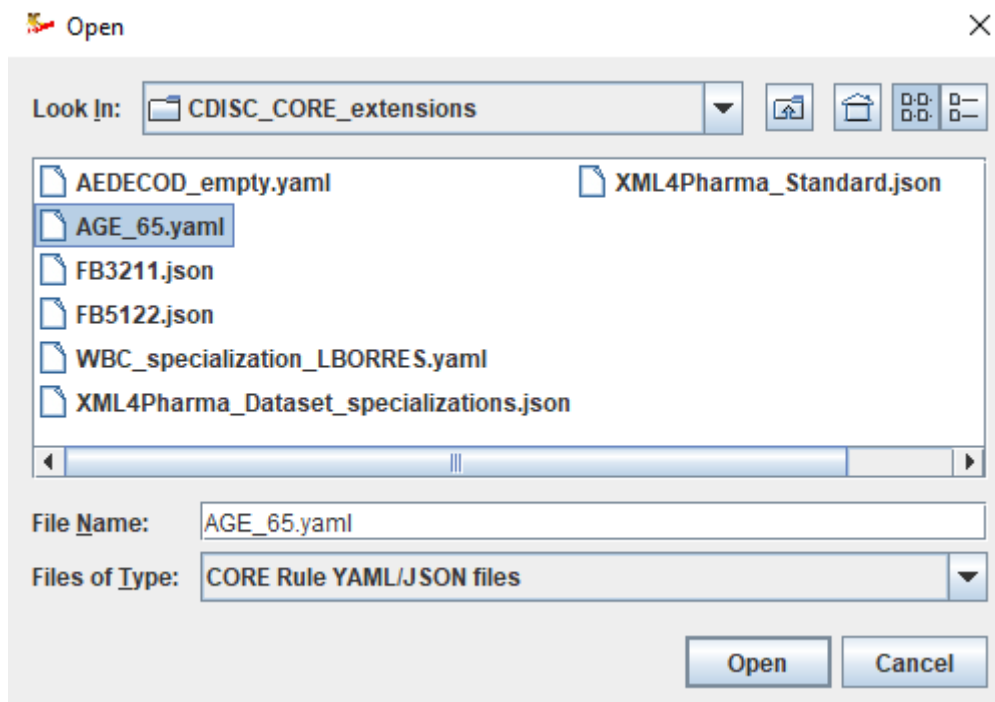
Remark that the development of "Dataset Specialization" CORE rules is still experimental and under further development. We will further improve and extend these "custom" rules in the next months.

When one selects "Own custom data quality rule(s)", one is requested to provide either a directory where you have your custom rules, or one or more YAML (or JSON⁴) files in such a directory.



For example:

⁴ In most cases, it will be easier to develop such core rules in YAML format.

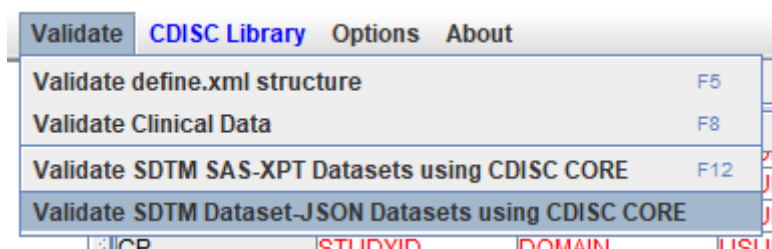


In case a directory is selected, or more than one rule is selected, it is later than still possible to include or exclude specific rules.

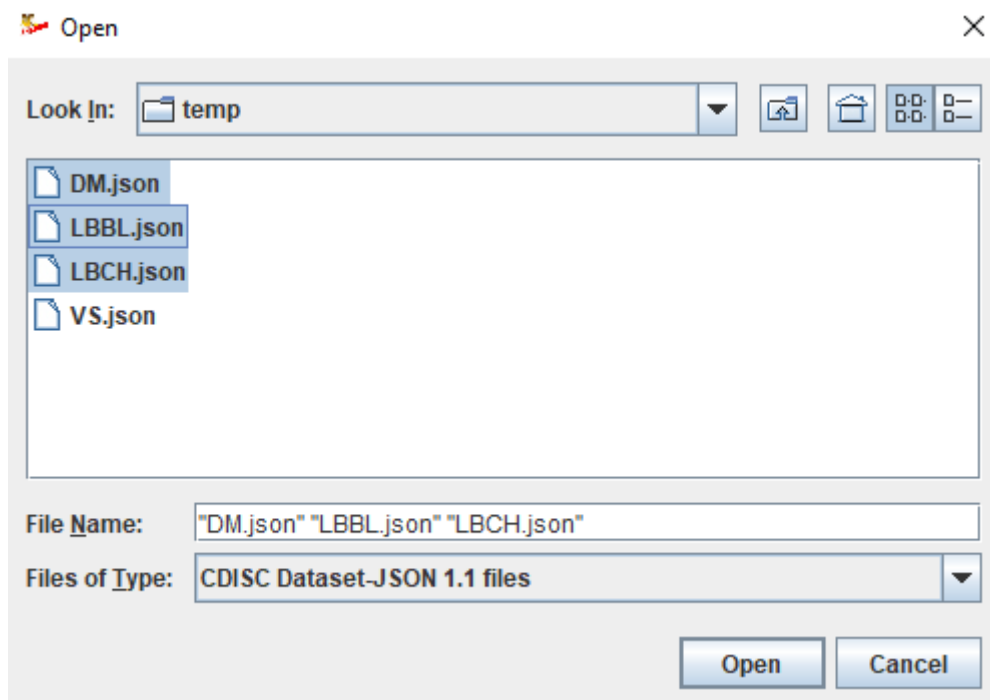
CORE validation of existing datasets

Usually one will do CORE validation immediately after the generation of the datasets, by checking the checkbox "Perform CDISC CORE validation on generated SDTM files" in the dialog for the execution of the mappings.

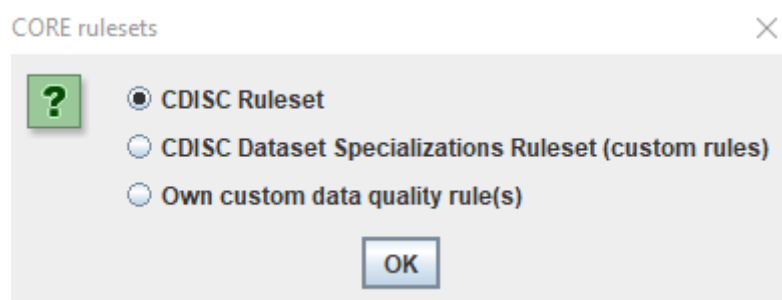
New is that one can also do CORE validation on existing datasets using the menus "Validate - SDTM/SEND SAS-XPT Datasets using CDISC CORE" and "Validate - SDTM/SEND Dataset-JSON Datasets using CDISC CORE":



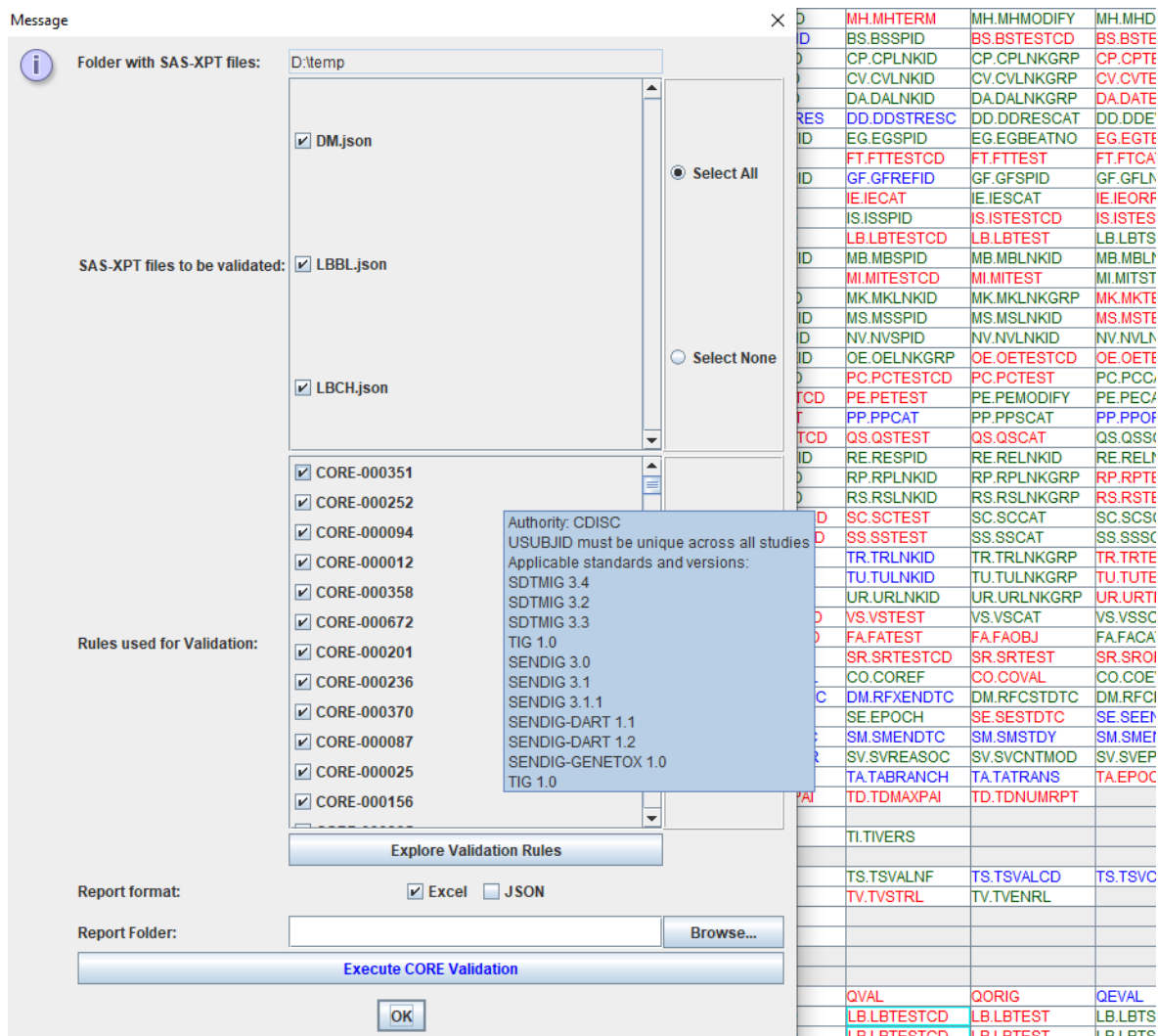
A "file chooser" is then presented to the user, where one can select one or more files, e.g.



Just like when CORE validation is included in execution of the mappings, one can choose between using the classic CDISC ruleset, or the new Dataset Specialization (custom) ruleset:



and the typical "CORE Validation" dialog is started, e.g.:



where one can then (if desired) make a selection of the rules to be applied, using the "Explore Validation Rules" button.

Limitations: CORE validation is currently only available on Windows. We are working on adding an implementation for Linux.

New way of automated naming of different instances of the same study-specific domain

When generating a second, third, ... instance of the same domain, the OID (identifier) and Name of the dataset definition assigned so far typically was a number. For example, when a second study-specific instance of LB is created (either using drag-and-drop or using the menu "Edit - Copy" and "Edit - Paste") the assigned name is "LB.1", the third one "LB.2" etc.. The user can then assign new, more logic names, like "LBCH" for "Laboratory - Chemistry" or LBUR for "Laboratory - Urinalysis".

On request of our users, we have now changed the mechanism for assigning the OID and the Name: for example, when a second instance of LB is created, the assigned name is "LBAA", and when a third one is created, "LBAB" is assigned. For the OID, the assignment is still done based on the identifier of the study, followed by a colon and the name of the domain instance, such as "CES:LBAA".

CES:LB	STUDYID	DOMAIN	USUBJID
CES:LBAA	STUDYID	DOMAIN	USUBJID
CES:LBBB	STUDYID	DOMAIN	USUBJID

When already a logical name was assigned, like LBCH (for "Laboratory - Chemistry") and a copy-paste is done from it, the following character in the alphabet will be used to form the new name, in this case "LBCI".

The user should then of course take care that more logic names are assigned, by selecting the first cell in the row, and then use the menu "Edit - SDTM/SEND Dataset Definition Properties".

Improved dialogs when adapting the OID of the dataset definition in the case of "split domain" datasets

When there are different instances of the same domain, the SDTM standard expects that the "Name" of each instance consists of 4 characters, and is unique. So, in some cases, one may want to change the name (and also the associated OID identifier) of such a dataset definition. In the above case, where we have instances "LB", "LBAA" and "LBBB", we will probably want to change the name of "LB", as the two-character name suggests this is the "overall" (i.e. non-split or "merged") dataset definition. In our case, we do however want to have 3 different instances, so we want e.g. to change "LB" into "LBCC".

In order to do so, we click the first cell "CES:LB" and change the "Name" from "LB" to "LBCC":

Edit properties for SDTM dataset/domainLB with OID CE

?

Name :

LB

OID :

CES:LB

Domain:

LB

SAS Dataset Name:

LB

Purpose :

Tabulation

Comment:

Edit properties for SDTM dataset/domainLB with OID CE

?

Name :

LBCC

OID :

CES:LBCC

Domain:

LB

SAS Dataset Name:

LBCC

Purpose :

Tabulation

Comment:

When then clicking "OK", a new (improved) dialog comes up:



You changed the dataset name from **LB** to **LBCC**,
Do you want to adapt the OIDs of the variables accordingly?
This may be useful/necessary in case you have different instances of the domain
and you want variables with the same name have different properties in the different instances,
such as is usually the case for **AP** (Associated Persons) or **QS** (Questionnaires) datasets.

There is **no need** to adapt the OIDs of the variables
when you do **NOT expect to have different instances** of the same domain.

In case you later want to be able to **merge** the different instances of the same domain,
you **should not adapt** the OIDs of the variables now!
In such a case, you can still differentiate on properties of the variables within the same domain
by using **ValueLists** in the define.xml

- ☐ Adapt OIDs of the variables in the define.xml
(typical best practice for QSxx and APxx where no merging is expected).

- ☒ Do not adapt OIDs of the variables in the define.xml
(typical when one later wants to also generate a 'merged' dataset).
P.S. You can then still differentiate on properties of the variables in the same domain
by using **ValueLists** in the (final) define.xml

OK

The user can choose between having "instance-specific" variable identifiers (OIDs) (first option) or "shared" OIDs between the different instance of the domains (second option). In the case of "AP" ("Associated Persons") domains or "QS" ("Questionnaires") domains, one will usually want to have "instance-specific" OIDs (first option), as for these, one will usually not want to generate a "merged" dataset later anyway. For "AP" this is often even not possible, as e.g. APMH (Medical History of Associated Persons) and APVS (Vital Signs of Associated Persons) will have a total difference structure.

In the case of other domains (often "Findings" domains), one will however often want to later be able to generate a "merged" dataset. In that case, the different instances will need to have the same variable properties for the standard variables.

So, for non-AP, non-QS datasets

So for "AP" and "QS" domains, the system will pre-select the first radiobutton, whereas for other domains, the system will pre-select the second radiobutton, which are just suggestions. The user can however decide otherwise, and e.g. have "shared" OIDs (meaning shared variable properties) for different instances of QS, e.g. as it was decided to later also generate a "merged" QS dataset.

In our case, accepting the suggestion, the result after clicking "OK" is:

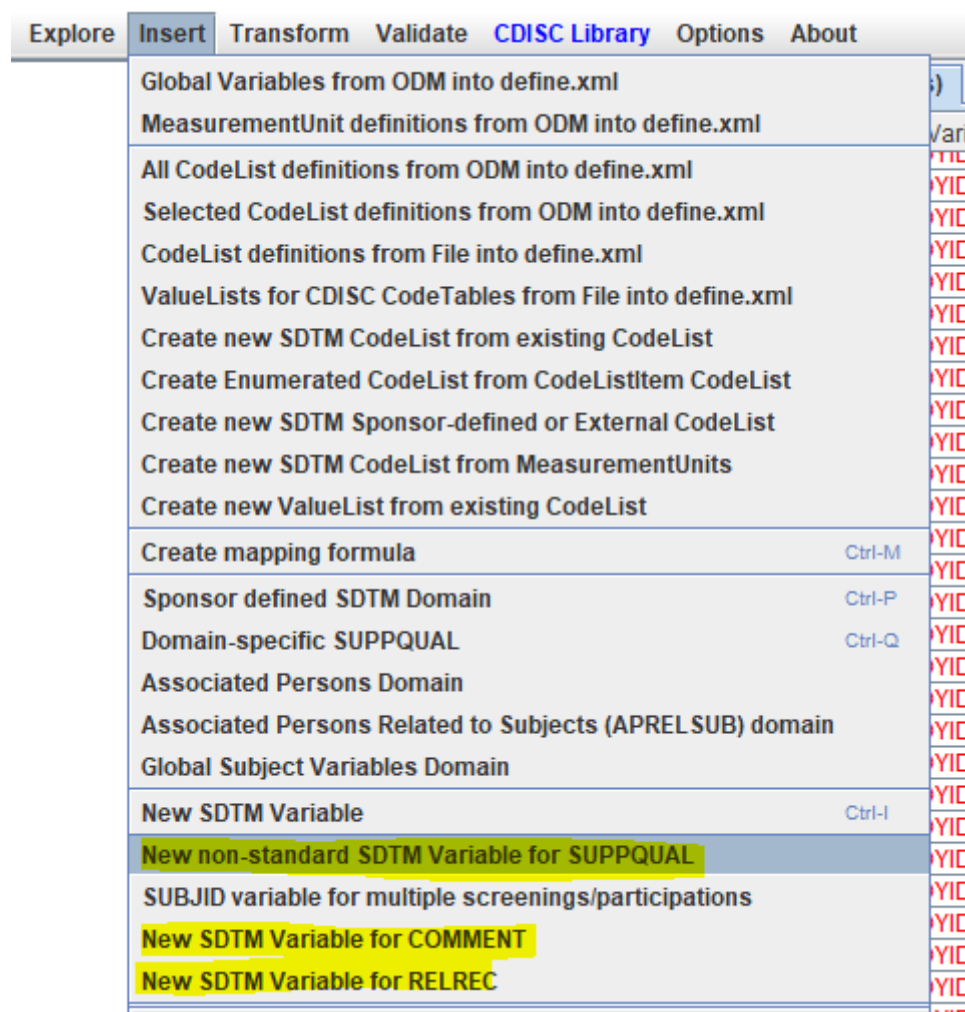
TI	STUDYID	DOMAIN	TI.IETESTCD	TI.IETEST	TI.IECAT	TI.IESCAT	TI.TIRL	TI.TIVERS		
TS	STUDYID	DOMAIN	TS.TSSEQ	TS.TSGRPID	TS.TSPARMCD	TS.TSPARM	TS.TSVAL	TS.TSVALNF	TS.TSVALCD	TS.TSVCDREF
RELREC	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	RELTYPE	RELID			
SUPPQUAL	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	QNAM	QLABEL	QVAL	QORIG	QEQAL
RELSUB	STUDYID	USUBJID	POOLID	RSUBJID	SREL					
OI	STUDYID	DOMAIN	OI.OIHOID	OI.OISEQ	OI.OIPARMCD	OI.OIPARM	OI.OIVAL			
CES.LBCC	STUDYID	DOMAIN	USUBJID	LB.LBSEQ	LB.LBGRPID	LB.LBREFID	LB.LBSPID	LB.LBTESTCD	LB.LBTEST	LB.LBCAT
CES.LBAA	STUDYID	DOMAIN	USUBJID	LB.LBSEQ	LB.LBGRPID	LB.LBREFID	LB.LBSPID	LB.LBTESTCD	LB.LBTEST	LB.LBCAT
CES.LBBB	STUDYID	DOMAIN	USUBJID	LB.LBSEQ	LB.LBGRPID	LB.LBREFID	LB.LBSPID	LB.LBTESTCD	LB.LBTEST	LB.LBCAT

with "shared" OIDs for all instances of LB, meaning that the variables for these will have the same properties (datatype, length, codelists ...) independent of the instance.

In such a case, one can later further differentiate between instance properties by using "ValueLists" in the define.xml.

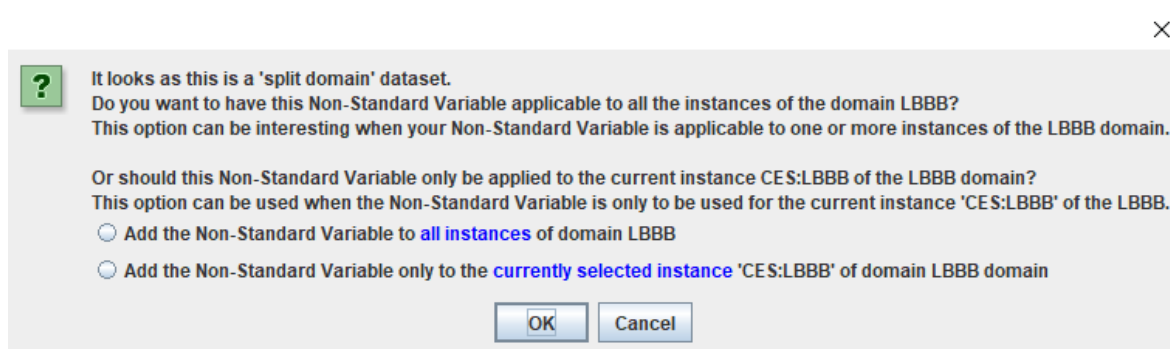
Improved handling of adding new inserted "non-standard Variables", "Variable for Comment" and "Variable for RELREC" in the case of "split-domain" datasets

SDTM-ETL allows to treat "non-standard variables" (NSVs) that usually later go into a SUPPXX dataset, as just normal variables, which makes developing the mappings considerably easier. The same applies for "comment variables" (will usually later go into a "CO" dataset) and "RELREC variables" (that will go into a RELREC dataset):



This may however become a bit complicated when there are different instances of the same domain. For example, one will want to have a NSV that is applicable to all instances of the same domain, or only to one instance. Just as an example, when having an LBUR (Urinalysis Laboratory) variable, and one want to add NSVs that are used to provide the datetime of the first collection and last collection of the urine when all samples are combined, this will of course not apply to an instance of LBHE (Hematology Laboratory), so we want to add these NSVs to LBUR only. But in many cases, one will want to have the NSV be added to each instance of the domain.

If we go back to our example with LBAA, LBBB and LBCC, when selecting a cell in the LBBB row, and then use the menu "Insert - New non-standard SDTM Variable for SUPPQUAL", the following dialog is displayed:



If we select "Add the Non-Standard Variable to all instances", and then set up the NSV variable e.g. with the name "LBNSV", the result will be:

	LB.LBPTREF	LB.LBRFTDTC	LB.LBPTFL	LB.LBPDUR	LB.LBNSV
	LB.LBPTREF	LB.LBRFTDTC	LB.LBPTFL	LB.LBPDUR	LB.LBNSV
	LB.LBPTREF	LB.LBRFTDTC	LB.LBPTFL	LB.LBPDUR	LB.LBNSV

and the NSV is added to all three instances and share the same properties.

If "Add the Non-Standard Variable only to the currently selected instance ...", the result will be:

ELTM	LB.LBPTREF	LB.LBRFTDTC	LB.LBPTFL	LB.LBPDUR	
ELTM	LB.LBPTREF	LB.LBRFTDTC	LB.LBPTFL	LB.LBPDUR	LB.LBNSV
ELTM	LB.LBPTREF	LB.LBRFTDTC	LB.LBPTFL	LB.LBPDUR	

making the NSV applicable to LBBB only.

The same applies to when inserting a "Variable for Comment" or "Variable for RELREC". This will e.g. lead to (just as an example) somewhat complicate combination like:

LBPTFL	LB.LBPDUR	LB.NSV2		
LBPTFL	LB.LBPDUR	LB.LBNSV	LB.NSV2	LBBB.LBBBREL...
LBPTFL	LB.LBPDUR	LB.NSV2	LBCC.LBCCCO...	

where the first NSV (LBNSV) is only applicable to LBBB, a second NSV is applicable to all

three instances of the LB domain, a "Comment" variable has been added to LBCC and a "Variable for RELREC" has been added to LBBB only.

Remark that the order of such variables is completely unimportant.

Also remark that one can always remove such "special" variables by a right-click, and confirm the action in the following dialog.

Automated population of COREF in CO datasets

In CO datasets, COREF ("Comment Reference") is often populated with the CRF page numbers, allowing the reviewer to trace back from where the comment comes from in the CRF.

As of version 5.1, and currently only for Define-XML 2.1, it is possible to have COREF automatically populated when having used "New Variable for COMMENT" from the "Insert" menu.

In order to do so, select the "Comment Variable" after having created it, and use the menu "Edit - SDTM Variable" (or use Ctrl-E), then select the checkbox "Edit Origin/Source", then click the button "Edit", leading to:

The screenshot shows a software interface with a list of variables on the left and a configuration dialog on the right. The list includes 'DM.DMCOMM', 'DMCOMM', 'text', '200', '200', '-1', 'COMMENT', 'COMMENT', 'CL.C141657.TENMW1T', 'NONE DEFINED YET', 'NO CODELIST ASSIGNED', 'Comments for domain D', 'NO VALUELIST ASSIGNED', and 'NO VALUELIST'. The dialog box, titled 'Designing/Updating Origin for Item: DMCOMM', contains the following elements:

- Origin type:** Radio buttons for Not Available, Assigned (selected), Protocol, Derived, Predecessor, and Collected.
- Source type:** Radio buttons for Subject, Investigator, Vendor, and Sponsor.
- Origin description:** A large text area.
- Document (leaf) ID:** A dropdown menu showing 'Location.CO'.
- Page details:** Radio buttons for 'No page details', 'Page list (physical reference)', and 'Named destinations'.
- Page list / List of named destinations:** A text input field.
- Page range:** Radio button for 'Page range: first page - last page'.
- First page:** A text input field.
- Last page:** A text input field.
- Title:** A text input field.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom.

One can now add the CRF page number or page range, by clicking "Collected" and the

appropriate radiobutton for "Source". A single page number or a list of them can then be added by selecting "Page List ..." and adding them as a blank-separated list, or selecting "Page range ..." and then assigning a (positive) integer for both "First page" and "Last page". For example, using "Page List":

Origin type:

☐ Not Available

☐ Assigned

☐ Protocol

☐ Derived

☐ Predecessor

☒ Collected

Source type:

☐ Subject

☒ Investigator

☐ Vendor

☐ Sponsor

Origin description

Document (leaf) ID:

Location.CO

☐ No page details

☒ Page list (physical reference)

☐ Named destinations

Page list / List of named destinations

3 7 22

When then executing the mappings, and requesting to "Move Comment Variables to Comments (CO Domain) this leads to:

CES:DM	CES:CO								
STUDYID	DOMAIN	RDOMAIN	USUBJID	CO.COSEQ	CO.IDVAR	CO.IDVARVAL	CO.COREF	CO.COVAL	
CES	CO	DM	001	1			CRF Pages 3,7,22	comment for subject 001	
CES	CO	DM	002	1			CRF Pages 3,7,22	comment for subject 002	
CES	CO	DM	003	1			CRF Pages 3,7,22	comment for subject 003	
CES	CO	DM	004	1			CRF Pages 3,7,22	comment for subject 004	
CES	CO	DM	005	1			CRF Pages 3,7,22	comment for subject 005	
CES	CO	DM	006	1			CRF Pages 3,7,22	comment for subject 006	
CES	CO	DM	007	1			CRF Pages 3,7,22	comment for subject 007	
CES	CO	DM	008	1			CRF Pages 3,7,22	comment for subject 008	
CES	CO	DM	009	1			CRF Pages 3,7,22	comment for subject 009	
CES	CO	DM	010	1			CRF Pages 3,7,22	comment for subject 010	


Remark that for Comments from the DM domain, IDVAR and IDVARVAL are not populated.

Automated population of COEVAL in CO datasets

When using Define-XML 2.1, it is now also possible to populate COEVAL (Evaluator) when using a "Variable for Comment". Remark that COEVAL is only to be used for comments that essentially are results of evaluations, which excludes (objective) measurements.

When inserting a "Variable for Comment", an extended dialog is now shown:

Add new Variable (for COMMENTS) to domain DM X

 **New Variable for COMMENT: DM.DM**

Data type:

Length:

Origin:

Role:

Comment:

Description:

COMM

text

200

COMMENT

Comments for domain DM

Select how COEVAL must be populated

☒ No COEVAL

☐ COEVAL from list:

ADJUDICATION COMMITTEE
▼

Other COEVAL:

☐ COEVAL from Define-xml Origin/Source

Validate

OK

Cancel

The middle part being new in SDTM-ETL 5.1.

The user now has the choice between "No COVAL", which is for the case that the comment is not an evaluation", selecting COEVAL from the CDISC list (as it is under Controlled Terminology), or taking it from the source and origin assigned to this variable, and which is stored in the define.xml. The latter will be seldom the case though, as it requires that the "Origin-type" is "Assigned".

When "COEVAL from list" is selected, a choice list is displayed, inviting the user to select a value from the CDISC Controlled Terminology, like:

YID	DOMAIN	USUBJID	TS.TSSEQ	TS.TSGRPID	TS.TSREFID
YID					REFID
YID					GRPI
YID					GRPI
YID					REFID
YID					REFI
YID					REFID
YID					REFI
YID					GRPII
YID					GRPI
YID					REFI
YID					GRPI
YID					REFI
YID					REFI
YID					EST
YID					SPID
YID					EFID
YID					SPID
YID					REFI
YID					REFI
YID					PID
YID					REFI
YID					D
YID					NRL
YID					CD
YID	DOMAIN				ID.TD.GTP
YID	DOMAIN				
YID	DOMAIN				TI.IESCAT
YID	DOMAIN	TS.TSSEQ	TS.TSGRPID	TS.TSPARMCD	TS.TSPARM
YID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	RELTYPE

Add new Variable (for COMMENTS) to domain DM

New Variable for COMMENT: DM.DM

Data type:

Length:

Origin:

Role:

Comment:

Description:

COMM

text

200

COMMENT

Comments for domain DM

Select how COEVAL must be populated

☐ No COEVAL

☒ COEVAL from list:

ADJUDICATION COMMITTEE

ADJUDICATION COMMITTEE

ADJUDICATOR

AUDIOLOGIST

CARDIOLOGIST

CAREGIVER

CERTIFIED ASSESSOR

CHILD

CLINICAL PATHOLOGIST

CLINICAL RESEARCH ASSOCIATE

CLINICAL RESEARCH COORDINATOR

This can then e.g. lead to a CO record like:

AL	COREF	COVAL	COEVAL	COEVALID
		Suspicion of structural heart disease	CARDIOLOGIST	

Once again, COEVAL is to be used solely in the case of comments that are evaluations!

New stylesheet for display of the define.xml (version 2.1) in the user's default browser

The older stylesheet from 2019 for the display of the define.xml in the user's own browser has been replaced by a newer one from 2023, taken from Lex Jansen's Github website at <https://github.com/lexjansen/define-xml-2.1-stylesheets/tree/master/localization/stylesheet>.

The 2019 stylesheet has been renamed into "define2-1-0_browser_2019.xsl" in the folder "stylesheet".

The new stylesheet has considerably more parameters than the old one. We however chose not

to develop a dialog for all these parameters for the simple reason that one will often want to use the same stylesheet for the regulatory submission. In that case, and when the user is not satisfied with the default parameter settings in the stylesheet file "define2-1-0_browser.xml", the user is encouraged to edit the stylesheet file "define2-1-0_browser.xml" and provide other values for the parameters.

Also remark that the stylesheet delivered with the regulatory submission is the responsibility of the sponsor, and not of CDISC nor of the mapping software provider!

Removal of all Dataset-XML features

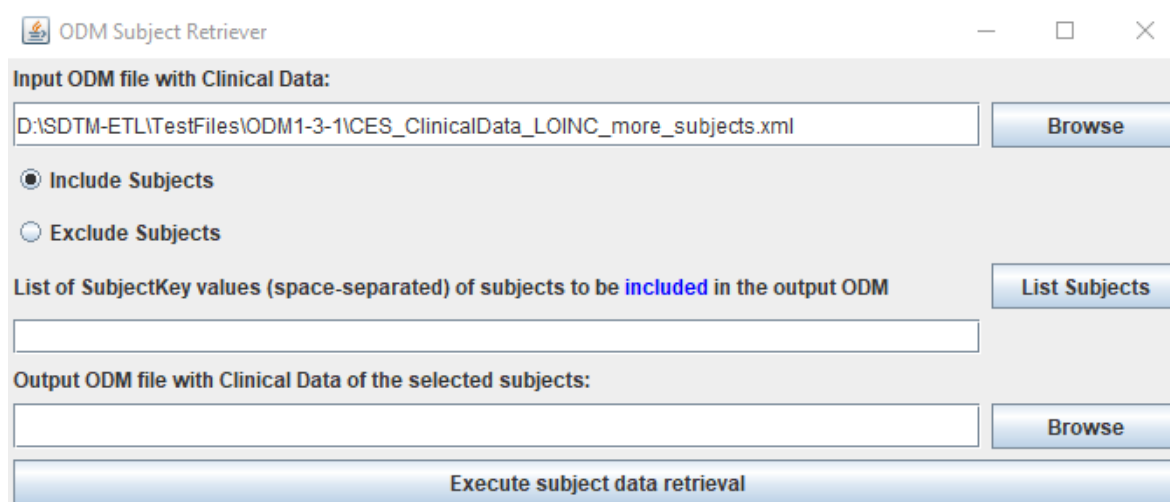
Now that it is almost certain that FDA (and later, also other regulatory authorities) will move to Dataset-JSON as a submission format (with a transition period also still supporting SAS Transport 5), it is also clear that the newer Dataset-XML format is obsolete. Therefore we removed all features that use Dataset-XML, such as the menu "Validate - Validate SDTM Dataset-XML Records against define.xml".

ODMSubjectRetriever

The separate program "ODMSubjectRetriever" (started by double click on "ODMSubjectRetriever.bat") has been further improved.

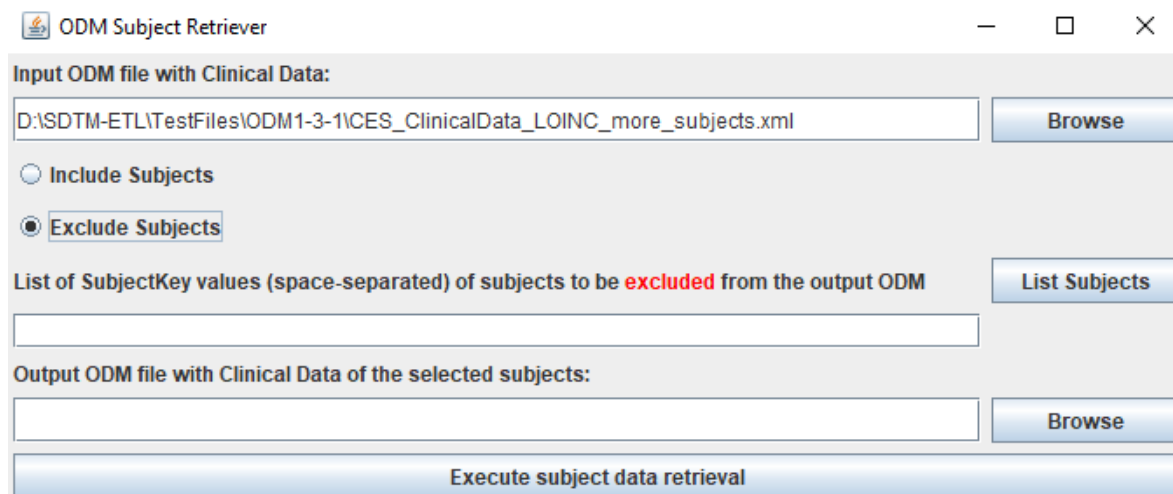
In the earlier versions, one could only "include" subjects, which was a bit cumbersome when one just wanted to exclude a few subjects, e.g. because the data for them is incomplete or does not meet quality requirements.

In the new version, one can choose between "including" or "excluding" subjects:

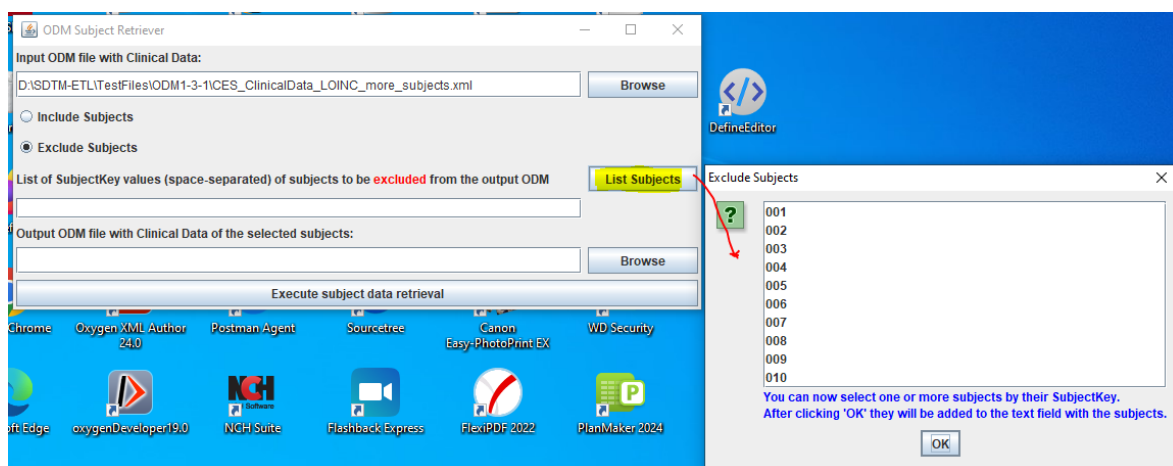


The screenshot shows the "ODM Subject Retriever" dialog box. It has a title bar with the text "ODM Subject Retriever" and standard window controls. The main area is divided into several sections:

- Input ODM file with Clinical Data:** A text box containing the path "D:\SDTM-ETL\TestFiles\ODM1-3-1\CES_ClinicalData_LOINC_more_subjects.xml" and a "Browse" button to its right.
- Selection Options:** Two radio buttons: "Include Subjects" (which is selected) and "Exclude Subjects".
- List of SubjectKey values:** A text box for entering subject keys, with a "List Subjects" button to its right. The text "List of SubjectKey values (space-separated) of subjects to be included in the output ODM" is above the box.
- Output ODM file with Clinical Data of the selected subjects:** A text box for the output path, with a "Browse" button to its right.
- Execute button:** A large blue button at the bottom labeled "Execute subject data retrieval".



Clicking "List Subjects" then generates a list of all subjects in the file with clinical data (by SubjectKey):



When one then e.g. selects "009" and "010" (use "Ctrl" to do multiple selection with the mouse), these two subjects will be excluded and the resulting ODM file will only contain the clinical data for the subjects 001 to 008.

Other small changes and improvements

Color coding in the mapping script editor

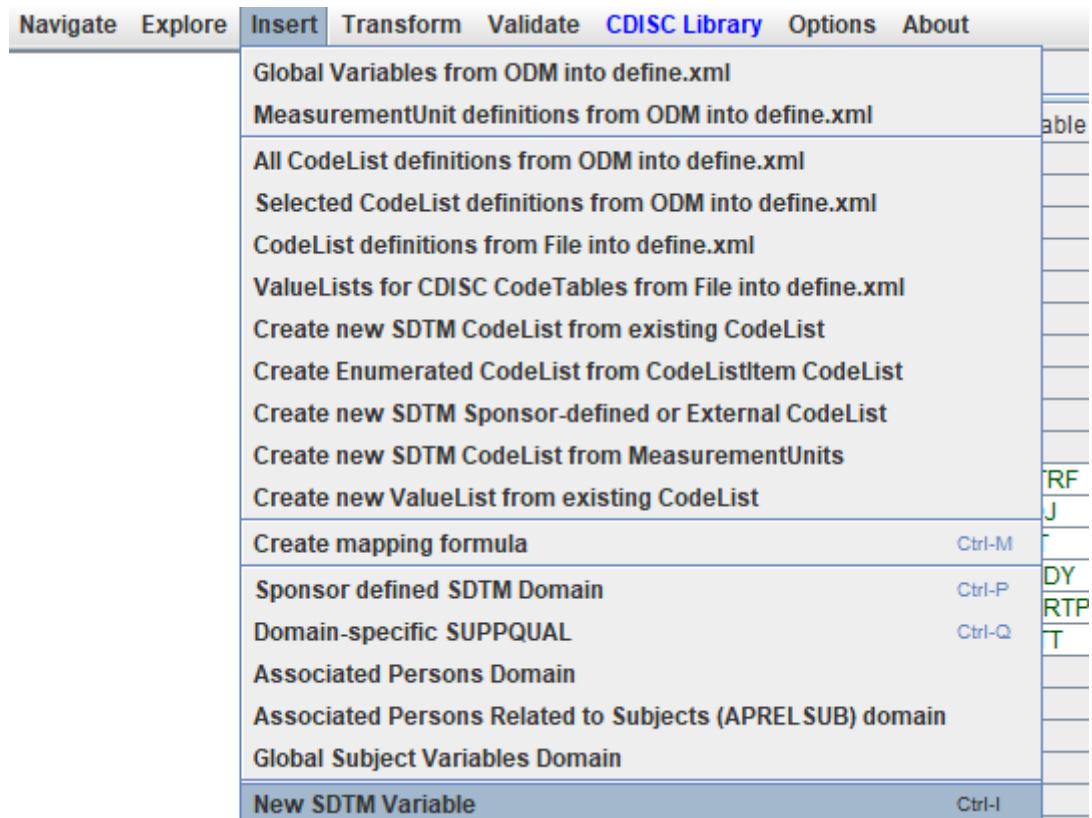
As one may already have noticed, the "\$" character that defines the start of a variable is now being colored. This makes it somewhat easier to not forget to add it when manually typing (parts of) a mapping script.

Search button for "Insert - New SDTM/SEND Variable"

With each new version of SDTM and SEND, the number of variables increases as well for the IG (Implementation Guide) as for the underlying SDTM model.

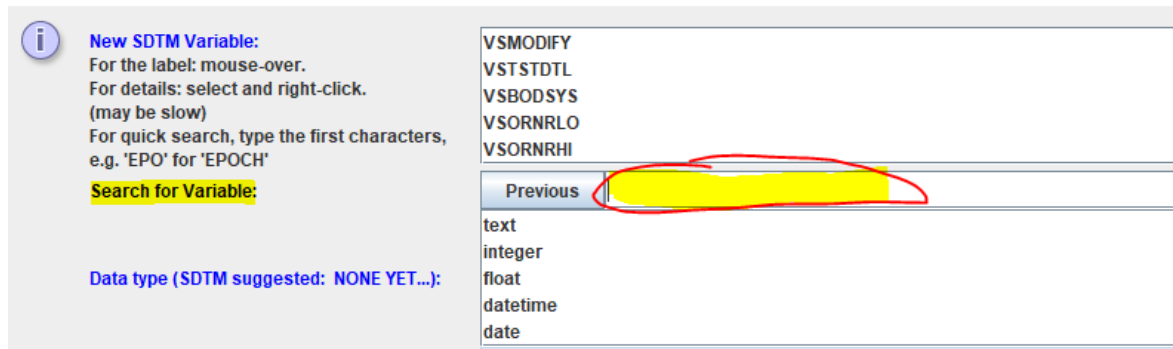
This also means that when one wants to add an SDTM Model variable, using the menu "Insert - New SDTM/SEND Variable" it becomes more difficult to find the desired one, especially when one is not so accustomed to SDTM or SEND. Therefore, we added a "Search button" to

the dialog:

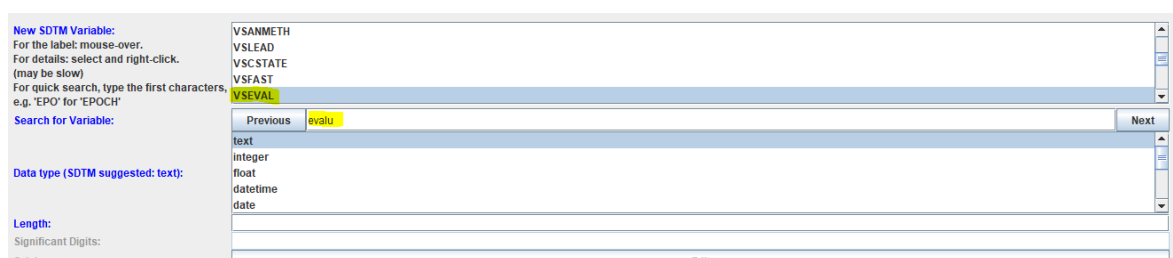


followed by:

Add new SDTM Variable to dataset/domain VS



When starting typing in the search field on the right, the system will start searching through all the proposed variables, and select a candidate. For example, when one needs a variable that has to do with "evaluator" or something similar:



By holding the mouse over the selected candidate variable, one can then check whether this is

what one wants, e.g.:

Add new SDTM Variable to dataset/domain VS

New SDTM Variable:
 For the label: mouse-over.
 For details: select and right-click.
 (may be slow)
 For quick search, type the first characters,
 e.g. "EPO" for "EPOCH"

Search for Variable:

Data type (SDTM suggested: text):

VSANMETH
 VSLEAD
 VSCSTATE
 VSFASST
 VSEVAL

Evaluator evalu

text
 integer
 float
 datetime

Next

If one then wants to find the next "hit", click the "Next" button.

In our case, we are satisfied with the choice, so we provide a (maximal) length, and already provide the "Origin", and/or assign a codelist. This can however be done at a later time (but should not be forgotten ...). We can check whether we have provided the minimum amount of information by using the "Validate" button, and when all fine, click the "OK" button, leading to:

Message

Inserted SDTM variable with OID VS.VSEVAL at position 24 for dataset VS (domain VS)

OK

VS.VSEVAL	VS.VSEVAL	VS.VSEVAL	VS.VSEVAL	VS.VSEVAL	VS.VSEVAL
-----------	-----------	-----------	-----------	-----------	-----------

The system then automatically inserts the new variable "VSEVAL" at the correct position according to the order provided by the SDTM model.

CORE validation Graphical User Interface improvements

The graphical interface for "CORE Rules Exploration" has been further improved: rules that are not applicable to the used standard and IG version get a gray color, and have obtained a tooltip containing the message that the rule is not applicable. For example:

<input type="checkbox"/>	CORE-000180	CDISC	Dataset	SDTMIG 3.4 SDTMIG 3.2 SDTMIG 3.3 TIG 1.0		RELATIONSHIP	
<input type="checkbox"/>	CORE-000724	FDA	Record	SDTMIG 3.2 SDTMIG 3.3 SDTMIG 3.4	EVENTS		AE
<input type="checkbox"/>	CORE-000105	CDISC	Record	SDTMIG 3.4 SDTMIG 3.3	FINDINGS		ALL
	Not applicable to SDTMIG version 3.2! CORE-000105			SDTMIG 3.4 SDTMIG 3.3	ALL		ALL
<input type="checkbox"/>	CORE-000193	CDISC	Dataset	SDTMIG 3.4 SDTMIG 3.3			

In such a case, the checkbox on the left, for selecting the rule to be included, cannot be checked either.

"CDISC Notes" updates

The file with "CDISC Notes" has been updated for SENDIG-3.1 and one for SENDIG-3.1.1 has been added (folder "CDISC_Notes").

SDTM model variables for SEND only

SDTM model version 2.0 contains a number of variables that are "only to be used in SEND", this although none of the SENDIG versions is based on SDTM model 2.0. This is rather strange, but it may be that when creating the SDTM model 2.0, one was thinking that later also a SENDIG version would be based on the same model. This however has never taken place.

In order to take care that such variables are not presented when (at least for SDTM dataset creation) the menu "Insert - New SDTM Variable" is used, we added these variables as "SDTMIG-forbidden" to the file "AllowedVariables_v2.0.xml" in the folder "CDISC_AllowedVariables". This folder contains all the model variables for the different versions of the SDTM/SEND models.

When then using SDTMIG-3.4, and using "Insert - New SDTM Variable", these variables are filtered out, so that they cannot be added. Typical examples of such variables are "-DETECT", "--EXCLFL", "FETUSID" and "RPHASE".

Removing "COMMENT" and "RELREC" variables

"COMMENT" and "RELREC" variables can now also be removed by selecting the cell and using a right-click. This feature was already in place for "Non-Standard" Variables (NSVs).

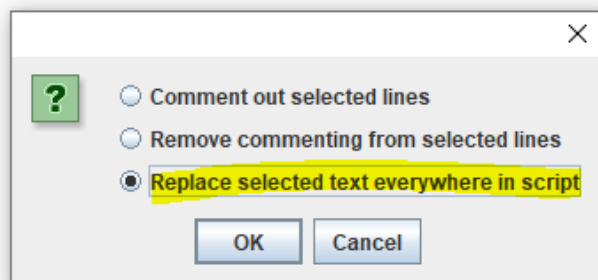
- In the mapping script editor, when having selected a word, or a set of words, and do a right-click, a new feature has been added. For example, when selecting the word "\$CODEDVALUE" in the script:

```
The Transformation Script

1 # Mapping using ODM element ItemData with ItemOID I_HEIGHT - value from attribute ItemOID
2 # Generalized for all StudyEvents
3 # Generalized for all Forms within the StudyEvent
4 # Generalized for all ItemGroups within the Form
5 # Generalized for all Items within the ItemGroup
6 # Mapping for ODM Items [I_DIABP, I_DIZZY, I_HEIGHT, I_SYSBP, I_WEIGHT] to SDTM CodeList VS.VSTESTCD
7 # with CodeList OID 'CL.C66741.VSTESTCD.SUBSET'
8 $CODEDVALUE = xpath(/StudyEventData[@StudyEventOID='BASELINE' or @StudyEventOID='WEEK_1' or @StudyEventOID='WEE
9 if ($CODEDVALUE == 'I_DIABP') {
10   $NEWCODEDVALUE = 'DIABP';
11 } elseif ($CODEDVALUE == 'I_DIZZY') {
12   $NEWCODEDVALUE = 'DIZZYNES';
13 } elseif ($CODEDVALUE == 'I_HEIGHT') {
14   $NEWCODEDVALUE = 'WEIGHT';
15 } elseif ($CODEDVALUE == 'I_SYSBP') {
16   $NEWCODEDVALUE = 'SYSBP';
17 } elseif ($CODEDVALUE == 'I_WEIGHT') {
18   $NEWCODEDVALUE = 'WEIGHT';
19 } elseif ($CODEDVALUE == '') {
20   $NEWCODEDVALUE = '';
21 } else {
22   $NEWCODEDVALUE = 'NULL';
23 }
24 $VS.VSTESTCD = $NEWCODEDVALUE;
```

and then do a right-click with the mouse, the following dialog is shown:

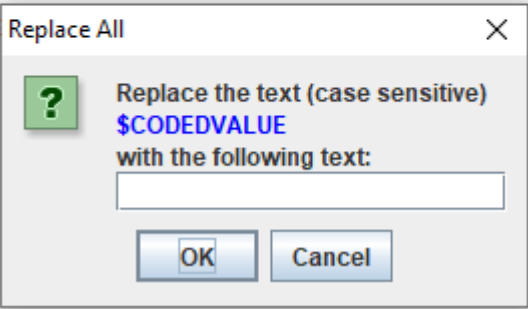
```
6 # Mapping for ODM Items [I_DIABP, I_DIZZY, I_HEIGHT, I_SYSBP, I_WEIGHT] to SDTM CodeList
7 # with CodeList OID 'CL.C66741.VSTESTCD.SUBSET'
8 $CODEDVALUE = xpath(/StudyEventData[@StudyEventOID='BASELINE' or @StudyEventOID='WEEK_1'
9 if ($CODEDVALUE == 'I_DIABP') {
10   $NEWCODEDVALUE = 'DIABP';
11 } elseif ($CODEDVALUE == 'I_DIZZY') {
12   $NEWCODEDVALUE = 'DIZZYNES';
13 } elseif ($CODEDVALUE == 'I_HEIGHT') {
14   $NEWCODEDVALUE = 'WEIGHT';
15 } elseif ($CODEDVALUE == 'I_SYSBP') {
16   $NEWCODEDVALUE = 'SYSBP';
17 } elseif ($CODEDVALUE == 'I_WEIGHT') {
18   $NEWCODEDVALUE = 'WEIGHT';
19 } elseif ($CODEDVALUE == '') {
```



New "Mapping script editor" features

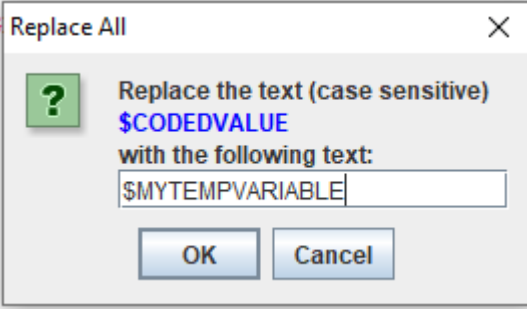
New is the item "Replace selected text everywhere in the script". When selected, and then using "OK", this leads to:

```
8 $CODEDVALUE = xpath(/StudyEventData[@StudyEventOID='BASELINE' or @StudyE
9 if ($CODEDVALUE == 'I_DIABP') {
10   $NEWCODEDVALUE = 'DIABP';
11 } elseif ($CODEDVALUE == 'I_DIZZY') {
12   $NEWCODEDVALUE = 'DIZZYNES';
13 } elseif ($CODEDVALUE == 'I_HEIGHT') {
14   $NEWCODEDVALUE = 'WEIGHT';
15 } elseif ($CODEDVALUE == 'I_SYSBP') {
16   $NEWCODEDVALUE = 'SYSBP';
17 } elseif ($CODEDVALUE == 'I_WEIGHT') {
18   $NEWCODEDVALUE = 'WEIGHT';
19 } elseif ($CODEDVALUE == '') {
20   $NEWCODEDVALUE = '';
21 } else {
22   $NEWCODEDVALUE = 'NULL';
23 }
24 $VS.VSTESTCD = $NEWCODEDVALUE;
```

A "Replace All" dialog box with a green question mark icon. The text inside says "Replace the text (case sensitive) \$CODEDVALUE with the following text:" followed by an empty text input field. At the bottom are "OK" and "Cancel" buttons.

If one then e.g. wants to have "\$MYTEMPVARIABLE" instead of "\$CODEDVALUE":

```
8 $CODEDVALUE = xpath(/StudyEventData[@StudyEventOID='BASELINE' or @StudyE
9 if ($CODEDVALUE == 'I_DIABP') {
10   $NEWCODEDVALUE = 'DIABP';
11 } elseif ($CODEDVALUE == 'I_DIZZY') {
12   $NEWCODEDVALUE = 'DIZZYNES';
13 } elseif ($CODEDVALUE == 'I_HEIGHT') {
14   $NEWCODEDVALUE = 'WEIGHT';
15 } elseif ($CODEDVALUE == 'I_SYSBP') {
16   $NEWCODEDVALUE = 'SYSBP';
17 } elseif ($CODEDVALUE == 'I_WEIGHT') {
18   $NEWCODEDVALUE = 'WEIGHT';
19 } elseif ($CODEDVALUE == '') {
20   $NEWCODEDVALUE = '';
21 } else {
22   $NEWCODEDVALUE = 'NULL';
23 }
24 $VS.VSTESTCD = $NEWCODEDVALUE;
```

A "Replace All" dialog box with a green question mark icon. The text inside says "Replace the text (case sensitive) \$CODEDVALUE with the following text:" followed by a text input field containing "\$MYTEMPVARIABLE". At the bottom are "OK" and "Cancel" buttons.

and then clicking "OK", this will replace all occurrences of "\$CODEVALUE" in the script by "\$MYTEMPVARIABLE", leading to:

```

7 # with CodeList OID 'CL.C66741.VSTESTCD.SUBSET'
8 $MYTEMPVARIABLE = xpath(/StudyEventData[@StudyEventOID='BASELINE' or @St
9 if ($MYTEMPVARIABLE == 'I_DIABP') {
10   $NEWMYTEMPVARIABLE = 'DIABP';
11 } elseif ($MYTEMPVARIABLE == 'I_DIZZY') {
12   $NEWMYTEMPVARIABLE = 'DIZZYNES';
13 } elseif ($MYTEMPVARIABLE == 'I_HEIGHT') {
14   $NEWMYTEMPVARIABLE = 'WEIGHT';
15 } elseif ($MYTEMPVARIABLE == 'I_SYSBP') {
16   $NEWMYTEMPVARIABLE = 'SYSBP';
17 } elseif ($MYTEMPVARIABLE == 'I_WEIGHT') {
18   $NEWMYTEMPVARIABLE = 'WEIGHT';
19 } elseif ($MYTEMPVARIABLE == '') {
20   $NEWMYTEMPVARIABLE = '';
21 } else {
22   $NEWMYTEMPVARIABLE = 'NULL';
23 }
24 $VS.VSTESTCD = $NEWMYTEMPVARIABLE;|

```

The following "editing" keyboards could were already be used in the mapping script editor: Ctrl-C ("Copy"), Ctrl-V ("Paste"), Ctrl-X ("Delete"), Ctrl-F ("Search"). In the latter case, this opens a new dialog where the user can add the search term. We have now also added "Ctrl-Z" ("Undo"),

We are planning to add some more such features for the "mapping script editor" in future. Remark that when using drag-and-drop from the ODM tree and following the wizards, in about 70-80% of the cases, one will not need to change anything in the mapping script.

Limitations to if-elseif-else structures

Before version 5.1, there was a limitation of the size of "if-elseif-...-else" structures which was hardcoded in the software. Reason is that such structures are translated into "<xsl:choose>" structures, where the translation can go wrong when there are errors in the script.

In the code, the limitation was 500 "snippets" (or tokens) which correspond to approximately 150 "elseif" statements with the "if-elseif-else" structure.

On request of one of our customers, we made this a parameter, with a default value of 2000, corresponding to about 600 "elseif" statements within the "if-elseif-else" structure.

The parameter can now also be set in the "properties.dat" file, using

"maxsnippetsinxslchoose=" and providing the maximum number of "snippets", The maximum number of "elseif" statements is than approximately 1 third of this value. For example:

```

postponeodmtreenoderecalculation=true
# set number of minutes between define.xml autosave
numminutesforautosave=15
# As of SDTM-ETL 5.1: max-number of "snippets" in if-elseif-else structures
# to be translated to <xsl:choose> (default: 2000)
maxsnippetsinxslchoose=3000
#

```

would lead to a maximum number of approximately 1000 "elseif" statements within each "if-elseif-else" structure.

Remark: the new default value of 2000 (about 600 "elsif") statements will be sufficient in >99% of the cases, so change this value (by editing the "properties.dat" file) only when absolutely necessary.

Bug fixes

QNAME-QLABEL match for Variables with more than 400 characters

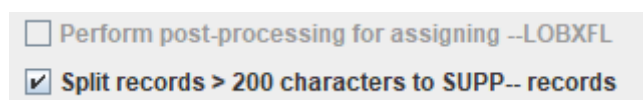
One of the major disadvantages of SAS Transport 5 is its limitation to 200 characters for the variable value. This means that when generating SAS-XPT files, and a variable value has more than 200 characters, the first 200 will go into the regular dataset, and the remaining⁵ (when less than another 200 characters) go into a "Supplemental Qualifier" dataset, typically with QNAME the name of the variable plus a sequence number. For example, when MHTERM has more than 200 characters, a SUPPMH dataset will be created with QNAME=MHTERM1 and QLABEL e.g. "Reported Term for the Medical History", or "Reported Term for the Medical History 1".

For the case that the number of characters is even larger (typically larger than about 400), we had a bug in the assignment of QLABEL, all subsequent QLABELs for the next QNAMEs having the same value for the QLABEL variable. This violates the SDTM standard stating that there must be a 1:1 relationship between QNAME and QLABEL.

Suppose we have a MHTERM with over 600 characters, then the structure of SUPPMH will become like:

QNAME	QLABEL	QVAL
MHTERM1	Reported Term for the Medical History 1	second "patch" of 200 characters ...
MHTERM2	Reported Term for the Medical History 2	third "patch" of 200 characters ...
MHTERM3	Reported Term for the Medical History 3	fourth "patch" of 200 characters ...

Remark that this does not require any programming effort from the user, one only needs to ensure that in the "SDTM Execution" dialog, the checkbox "Split records > 200 characters to SUPP-- records" is checked, which is done anyway automatically when one selected to have the datasets generated in SAS-XPT format.



Remark that when generating the datasets in Dataset-JSON or CSV format, or as "SQL Insert" statements, nothing of this applies, as these formats do not have any character length restriction.

Yet another reason to get rid of SAS-XPT format as soon as possible ...

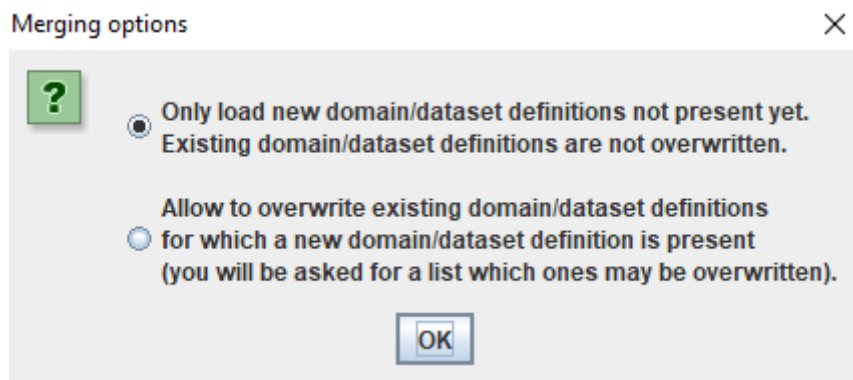
⁵ The matter is even a bit more complicated, as it is not allowed to "split" in the middle of a word ...

QLABEL in batch execution when also requesting to merge "split domain" datasets

When generating datasets using batch execution with automated SUPPxx and generation and requesting to also generate a "merged" dataset for "split domains", QLABEL was not correctly populated in the generate SUPPxx datasets.
This has been fixed.

Merging define.xml-s with mappings with the option to only load the new dataset definitions that were not present yet

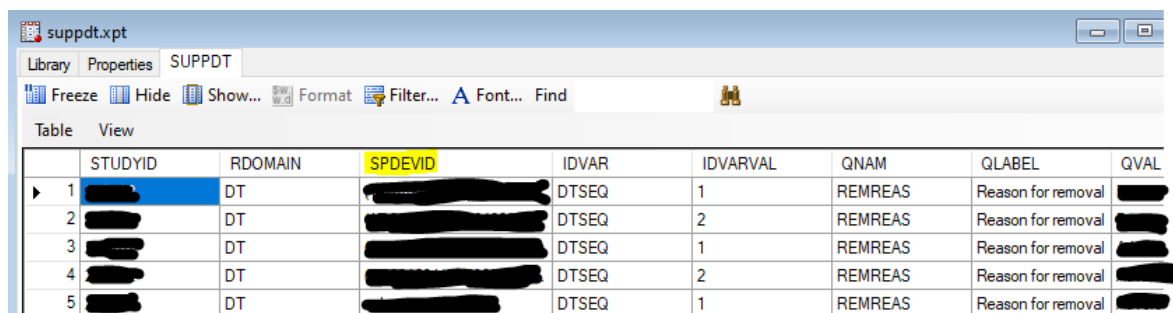
When loading additional dataset definitions from a define.xml with mappings ("merging"), the options presented are:



When the first option is selected, asking to only load dataset definitions that are not present yet, this not always went well, so that it could happen that the result was to have two instances of the same domain/dataset definition.
This has now been fixed.

Fixes for Medical Devices domains DI and DT

Two of the "[Medical Devices](#)" domains, DI (Device Identifiers) and DT (Device Tracking and Disposition), do not have USUBJID as the major identifier, but have SPDEVID (Sponsor Device Identifier). This gave problems in the automated assignment of --SEQ. This has been fixed now. It also gave problems when a non-standard variable (NSV) has been added which is then "split off" into SUPPDI or SUPPDT. When one now generates one of these two, then USUBJID is replaced by SPDEVID. For example:



	STUDYID	RDOMAIN	SPDEVID	IDVAR	IDVARVAL	QNAM	QLABEL	QVAL
1	[REDACTED]	DT	[REDACTED]	DTSEQ	1	REMREAS	Reason for removal	[REDACTED]
2	[REDACTED]	DT	[REDACTED]	DTSEQ	2	REMREAS	Reason for removal	[REDACTED]
3	[REDACTED]	DT	[REDACTED]	DTSEQ	1	REMREAS	Reason for removal	[REDACTED]
4	[REDACTED]	DT	[REDACTED]	DTSEQ	2	REMREAS	Reason for removal	[REDACTED]
5	[REDACTED]	DT	[REDACTED]	DTSEQ	1	REMREAS	Reason for removal	[REDACTED]

Remark that Pinnacle21 will throw two false positives for each row in SUPPDT, one for USUBJID not being present, and one for SPDEVID not being allowed.

Other small fixes

- when generating a "cleaned" define.xml, and selecting that the define.xml is "in the context of a regulatory submission", it is ensured that the attribute "SASFieldName" is filled with the same value as from the "Name" attribute⁶. Remark that this requirement essentially only applies to the case of a submission in SAS-XPT format in the case of Define-XML v.2.1.
- when having "non-standard variables" (NSVs) in "split domain" dataset definitions, and SUPPxx datasets were generated, the value of RDOMAIN in the latter was not always correctly assigned. This has been fixed.
- using the menu "Insert - Sponsor-defined SDTM domain" was very slow. The reason was that the file with "CDISC Notes" was read more than necessary. This has now been fixed. As a lot of information needs to be read in (essentially the whole SDTM Model), the response time is however still about 15 seconds.
- when using "View - SDTM/SEND CDISC Notes" (Ctrl-H), this did not always provide the desired results when the selected variable was part of a "split domain" dataset definition. This has now been fixed.
- when generating SUPPxx datasets, a NullPointerException occurred when trying to populate QEVAL, but no "source type" was provided under "Origin" for the non-standard variable. This has been fixed. In such a case a default value of "INVESTIGATOR" is used.

Other remarks

These new features have only partially been tested with other operating systems than Windows, such as on Linux. This is work in progress.

Several of the new features have not been tested yet with batch execution.

⁶ Essentially, this requirement is superfluous, as the value in the "Name" attribute and in "SASFieldName" will always be the same. It seems to be one of these FDA requirements due the primitiveness of their tools to work with define.xml. We also kept it in order to satisfy users who still use Pinnacle21 for validation of define.xml.