SDTM-ETL 4.4: Summary of New Features

Author: Jozef Aerts, XML4Pharma

Last update: **2024-02-29**

Summary

This document contains a summary of the most important new features of SDTM-ETL 4.4 and bug fixes.
There are many minor improvements and new features that are not described in this document, but that can be found in other manuals / tutorials of SDTM-ETL 4.4.

Table of Contents

Working with multiple codelists (further development of CDISC "CT-Relations")

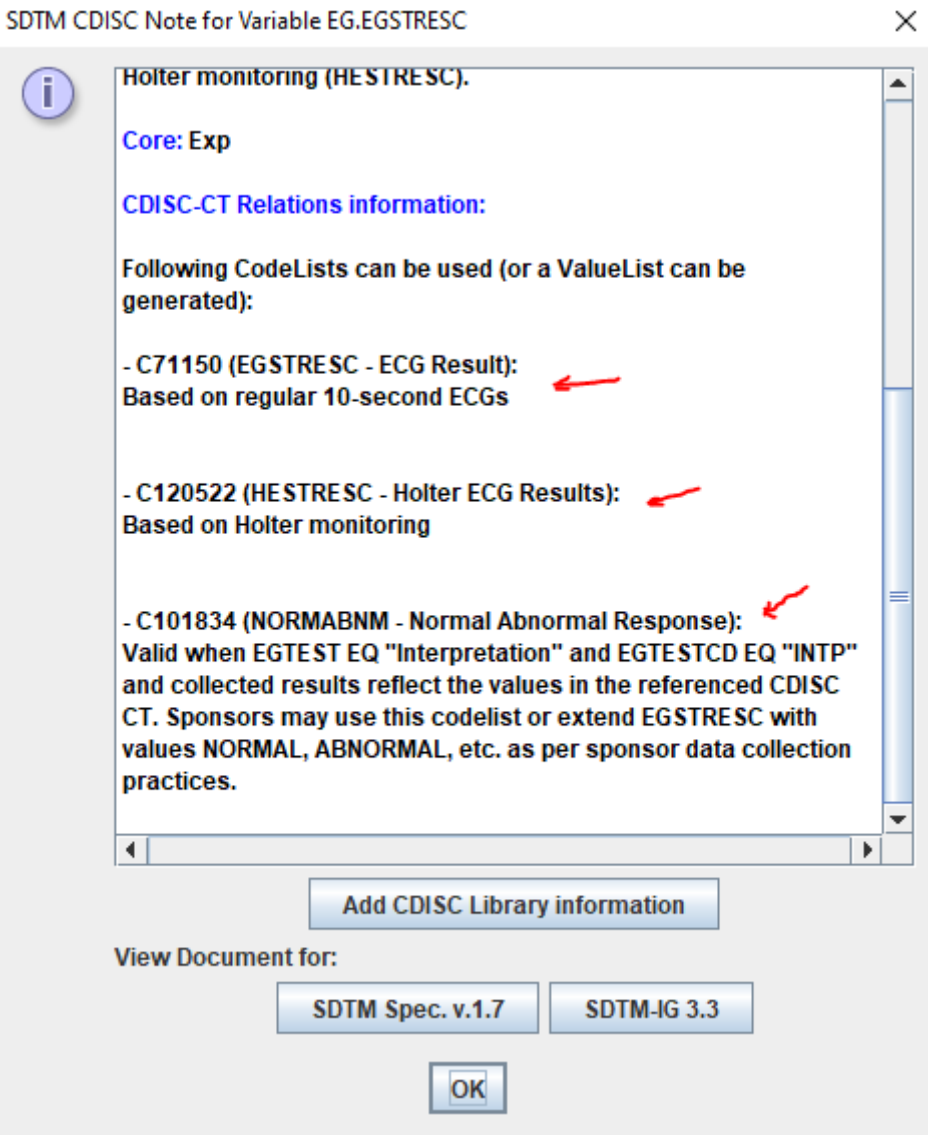
Some SDTM variables have different controlled terminology (i.e. associated codelists) depending on the use case. Examples are EGSTRESC, FASTRESC, RSCAT, etc..
Whether a variable has more than one possible codelist, can easily be seen in the "CDISC Library Browser", for example:

12	RSCAT	Category for Assessment	Used to define a category of related records across subjects. Examples: "RECIST 1.1", "CHILD-PUGH CLASSIFICATION". There are separate codelists used for RSCAT where the choice depends on whether the related records are about an oncology response criterion or another clinical classification. RSCAT is required for clinical classifications other than oncology response criteria.	Char	Grouping Qualifier	Exp	C118971; C124298

It doesn't however state which codelist must be used when.

CDISC has however published this information as "Codetable Mapping Files", unfortunately only in the form of Excel files, so barely usable in real applications.
Essentially, such "codetables" correspond to "ValueLists" in define.xml.
Therefore, we have transformed the CDISC "codetables" into files with Define-XML "ValueLists", so that they can immediately be used in mapping software.
We also generated a file with all use cases, from CDISC-Library API calls.

So, when selecting a variable for which there are multiple codelists, and asking for the "CDISC Notes", one also obtains information about the different use cases. For example:



Also, when "instantiating" FA (Findings About), and selecting a domain for which the "about" is, a list will be presented with possible codelists for FATESTCD, as this is dependent on the domain and/or the use case of the FA dataset.

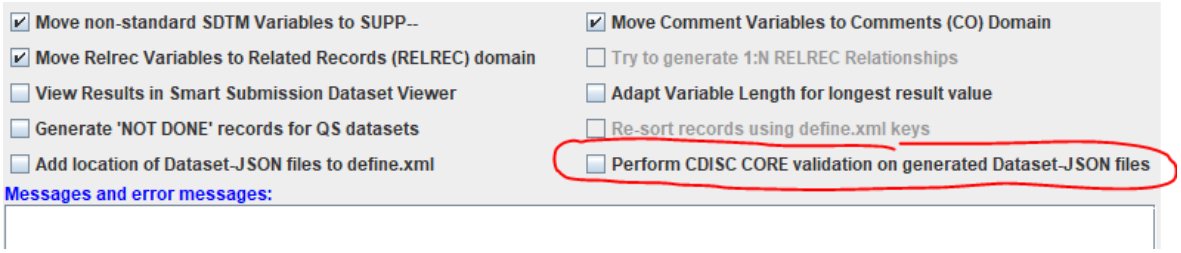
We have also tried to develop something similar for the CDISC "Therapeutic Area User Guides" (TAUGs), but these are unfortunately not available in an electronic form.

A separate tutorial "Handling multiple Codelists: CDISC Controlled Terminology Relationships" can be found on our website, containing all the details. This new feature and the "ready-to-go" ValueLists can save many many hours when developing mappings.

New CORE Validation Engine

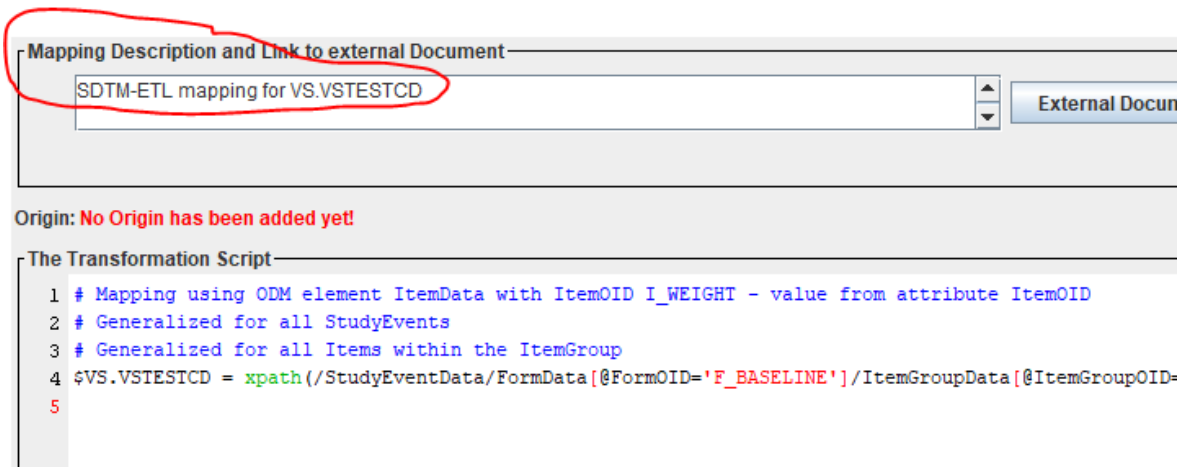
SDTM-ETL v.4.4 now comes with the CDISC CORE Engine generated from the main branch on 2023-11-15, which also supports Dataset-JSON as submission format. The implementation is however in such a way that when a new CORE version becomes available, it can just be replaced by the new one, without an update of the SDTM-ETL software. Exception is when the CORE command parameters to start CORE have been changed.
If this happens, we will make a new version of SDTM-ETL readily available.

This new CORE engine also means that CORE can be executed not only for the outdated SAS-XPT format, but also for the modern CDISC Dataset-JSON format.

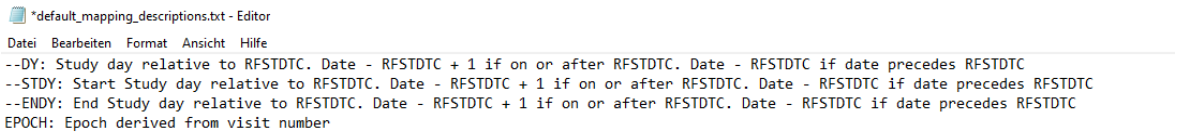


Default mapping descriptions

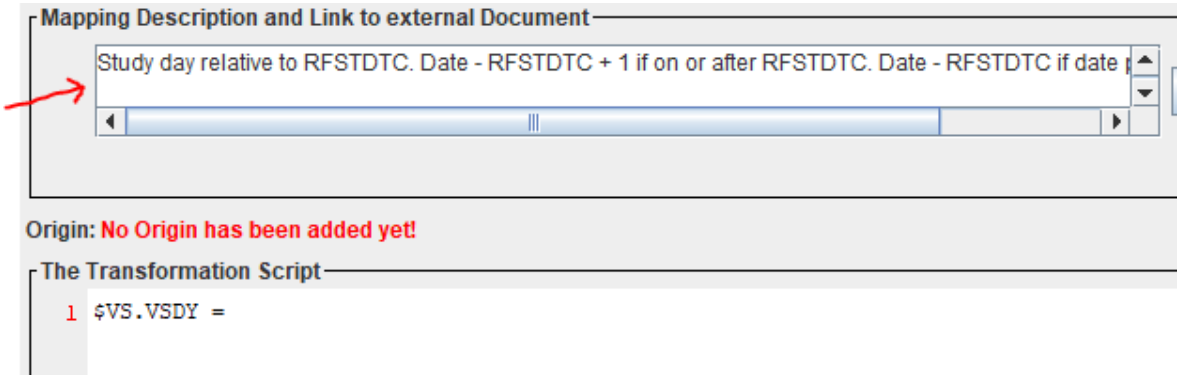
For each mapping, the user is expected to provide a short description:



For some variables, the mapping description will be extremely similar, even between studies. To avoid repetition, one can now provide such "standardized" descriptions in the file "default_mapping_descriptions.txt" which resides in main folder where the software is installed. The content in this file that comes with the software is:



Users can extend this file with their own mapping descriptions.
If then, for example, a mapping is started for VSDY, the description from the file is automatically added:

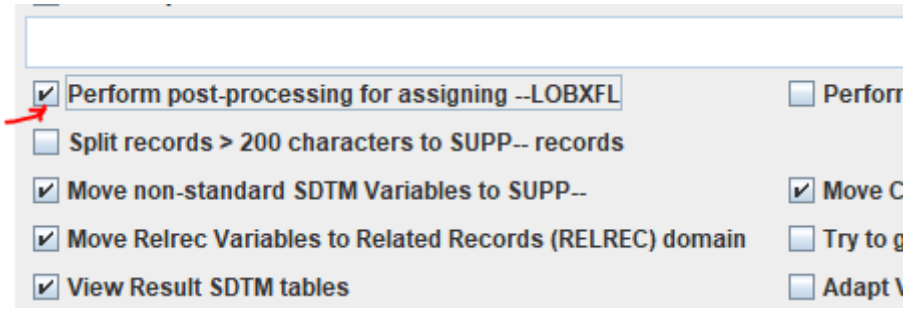


Also this new feature can save large amounts of time, and takes care that the descriptions, that later flow into the define.xml, especially when the variable is "derived", are consistent.

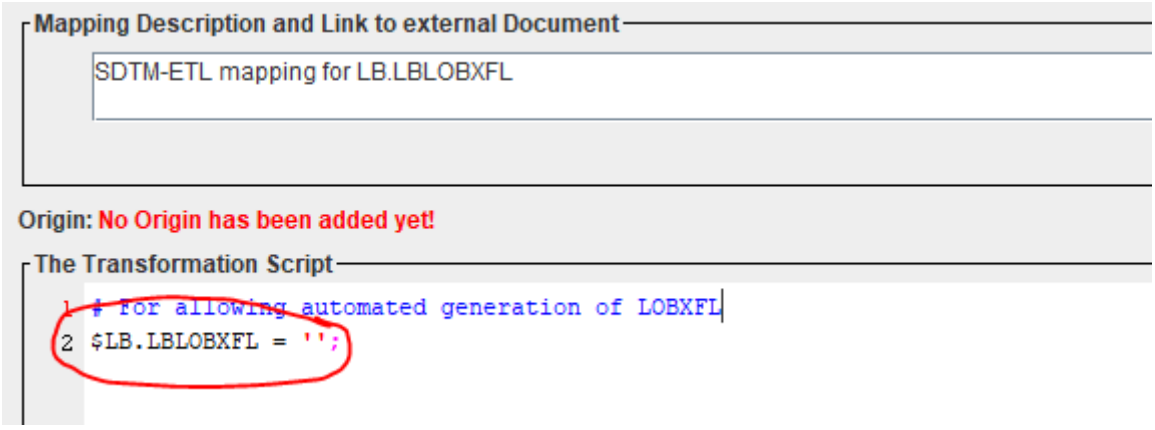
Automated (post-processing) assignment of –LOBXFL flags - using LOINC

Until v.3.2, the post-processing assignment of –LOBXFL (Last Observation Before First Exposure Flag) was solely based on the value of –TESTCD. In most cases (e.g. for vital signs) this usually is correct, but is not entirely correct for lab tests, as for lab tests, the value of LBTESTCD is not the unique identifier of the test (see e.g. here). A simple example is "GLUC" (Glucose). One can have a study where glucose is measured as well in blood as in urine. Essentially, these are two different tests, and one would then ideally have two baseline LBLOBXFL values for each subject.
Now, very often, one will have two datasets anyway, one for hematology, and one for urinalysis, so having separate baseline flags anyway, which of course remain when merging the different LB datasets into a single "super" one.

However, when there are only a few lab tests, users may prefer to generate only one LB dataset, which could then lead to only one baseline flag per subject and –TESTCD instead of several in case the assignment is automated. The latter is the case when the checkbox "Perform post-processing for assigning –LOBXFL" is checked in the last stage of the datasets generation:



In this case, one will use a "placeholder" mapping for –LOBXFL, e.g.



One can of course always provide ones own mapping script for –LOBXFL assignment, and then not use the post-processing mechanism.

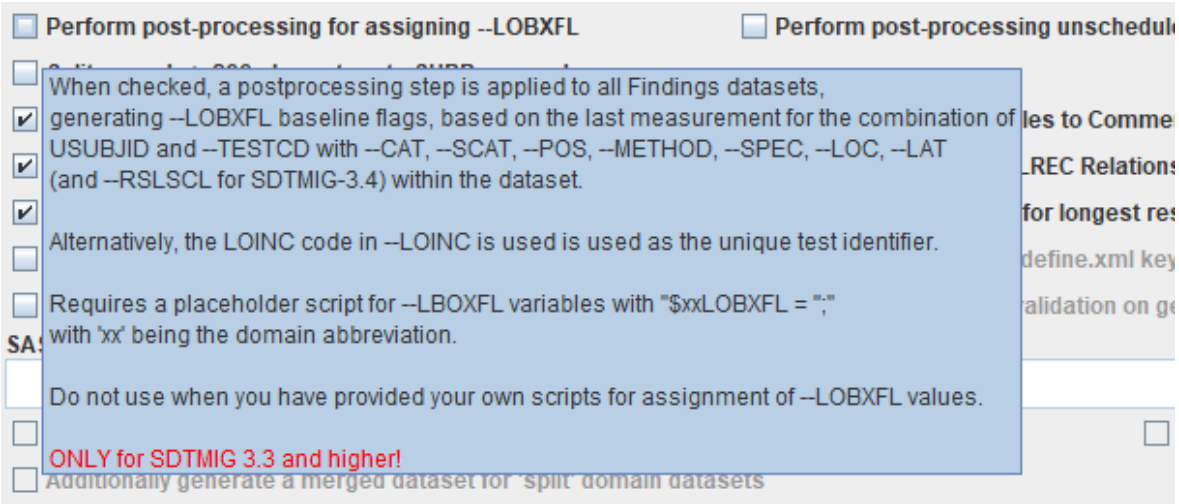
As said, until SDTM-ETL v.3.2, the assignment of --LOBXFL was based on the assumption that the value of –TESTCD defines the unique test.
As of SDTM-ETL v.3.3, this was changed to have more accurate baseline flags, by basing the "test uniqueness" on the combination of –TESTCD, –CAT, –SCAT, –POS, –METHOD, –SPEC, –LOC, –LAT, and (when using SDTMIG v.3.4) –RSLSCL (Result Scale[1]), of course when present and populated. As "–SPEC" is in this list, this will already allow to differentiate between "glucose in blood" and "glucose in urine", and assign different baseline flags for each separately.

Essentially however, the <u>only</u> unique identifier of the test in all Findings domains is the LOINC code. This as well for LB, MB, VS, QS, GF, …
Unfortunately, CDISC still refuses to recognize this, trying to "keep LOINC out of the door" as much as possible, due to "not-invented-here" …

In SDTM-ETL 4.4, we refined the algorithm for the automated assignment of –LOBXFL, now also making it available for the Dataset-JSON[2], Dataset-XML and CSV formats, with an extra new feature, using the LOINC value as the unique identifier for the test.
This can also be seen when keeping the mouse over the "Perform post-processing for assigning –LOBXFL" checkbox.

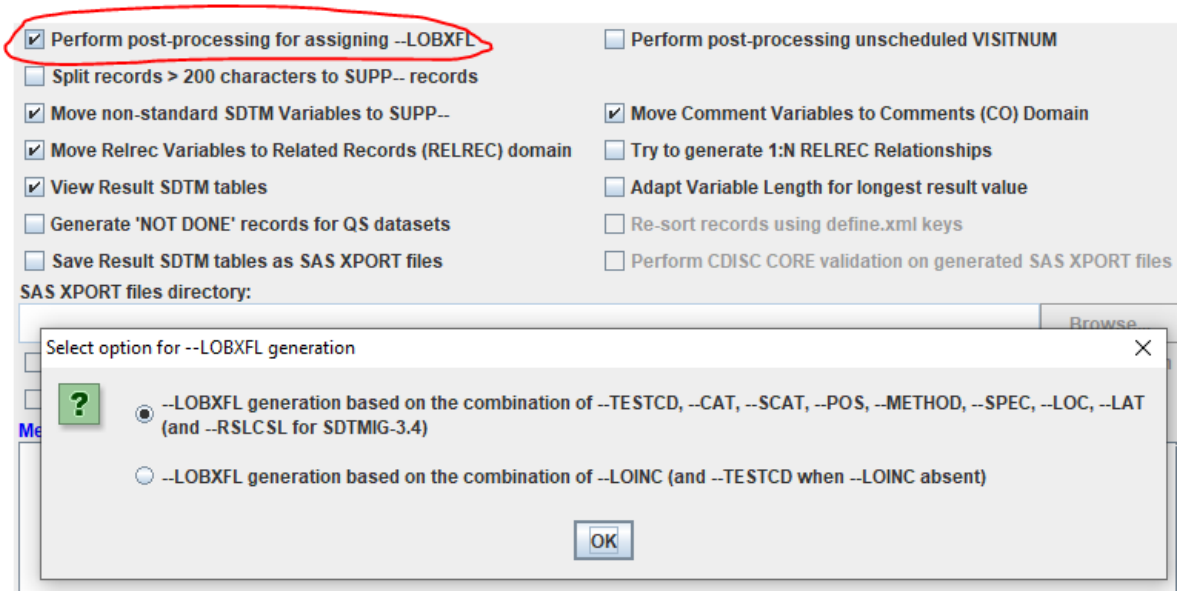
---

[1] This e.g. allows to differentiate between quantitative and qualitative tests.
[2] Support for –LOBXFL for Dataset-JSON is important, as we expect FDA to start accepting submissions in Dataset-JSON in 2024 or 2025.



When also the LOINC code (the real unique test identifier) is provided (e.g. In LBLOINC, VSLOINC, EGLOINC…), one can select it to be used as the unique test identifier for the algorithm, which essentially is the better choice.

When one checks the checkbox "Perform post-processing for assigning –LOBXFL", a dialog pops up:



allowing the choice between basing the "unique test" on a combination of SDTM variables (still the default), and the LOINC code (from –LOINC)
When then the radiobutton "–LOBXFL generation based … on LOINC …" is selected, the system will use the LOINC code for defining what a unique test it, and for those tests for which no LOINC code is provided, will base it on the value of –TESTCD (which is the old mechanism). The latter is important e.g. for the case there is no LOINC code (yet) for the test, or e.g. that the lab didn't provide it.

TS Generation: Use "FDA-desired list of TSPARMCD/TSPARM values"

The "Trial Summary" is, as its name states, a domain/dataset containing summarized information about the study, as well as for "planned" as for "actual"[3].

The FDA handles lists of the minimum parameters with their values it wants to obtain as part of a submission. These lists are now available in the files "FDA_TS_Codelist_clinical.xml" (for SDTM) and "FDA_TS_Codelist_nonclinical.xml" (for SEND) in the "CDISC_CT" folder. For example, for the former, the content contains:

FDA_TS_Codelist_clinical.xml

```xml
<CodeList xmlns="http://www.cdisc.org/ns/odm/v1.3" OID="CL.FDA.TSPARMCD"
    Name="CodeList for Trial Design dataset TS, variable TSPARMCD" DataType="text">
    <CodeListItem CodedValue="ACTSUB">
        <Decode>
            <TranslatedText>Actual Number of Subjects</TranslatedText>
        </Decode>
    </CodeListItem>
    <CodeListItem CodedValue="ADAPT">
        <Decode>
            <TranslatedText>Adaptive Design</TranslatedText>
        </Decode>
    </CodeListItem>
    <CodeListItem CodedValue="ADDON">
        <Decode>
            <TranslatedText>Added on to Existing Treatments</TranslatedText>
        </Decode>
    </CodeListItem>
    <CodeListItem CodedValue="AGEMAX">
        <Decode>
            <TranslatedText>Planned Maximum Age of Subjects</TranslatedText>
        </Decode>
    </CodeListItem>
    <CodeListItem CodedValue="AGEMIN">
        <Decode>
            <TranslatedText>Planned Minimum Age of Subjects</TranslatedText>
        </Decode>
    </CodeListItem>
    <CodeListItem CodedValue="COMPTRT">
        <Decode>
            <TranslatedText>Comparative Treatment Name</TranslatedText>
        </Decode>
    </CodeListItem>
    <CodeListItem CodedValue="CRMDUR">
        <Decode>
            <TranslatedText>Confirmed Response Minimum Duration</TranslatedText>
        </Decode>
    </CodeListItem>
    <CodeListItem CodedValue="CTAUG">
        <Decode>
            <TranslatedText>CDISC Therapeutic Area User Guide</TranslatedText>
        </Decode>
```

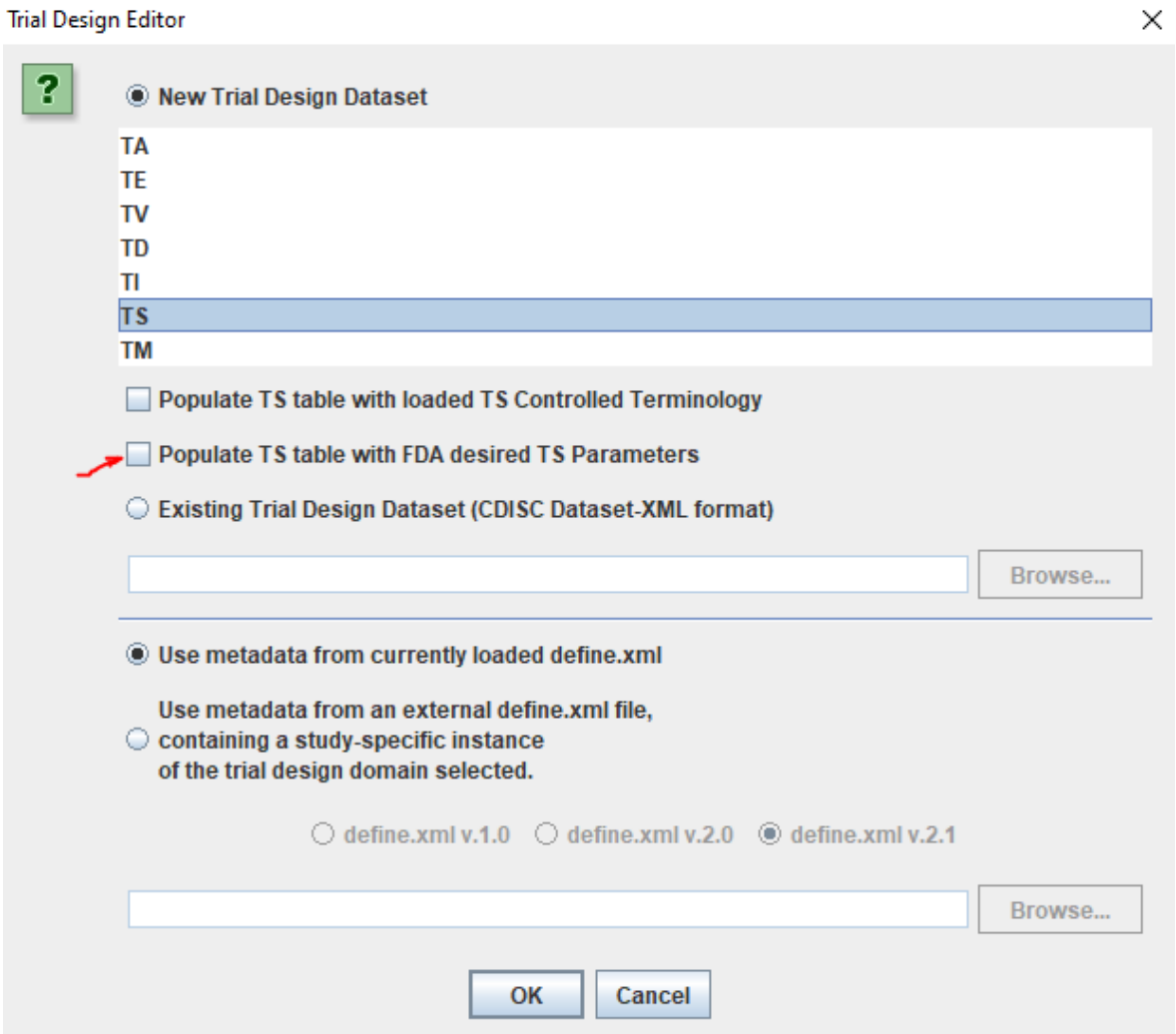
so, essentially as a codelist.

<u>IMPORTANT REMARK</u>: the content of these two files come without any guarantee of completeness or correctness. It is the duty of the user to keep these files up to date, e.g. when new requirements are published by the FDA.

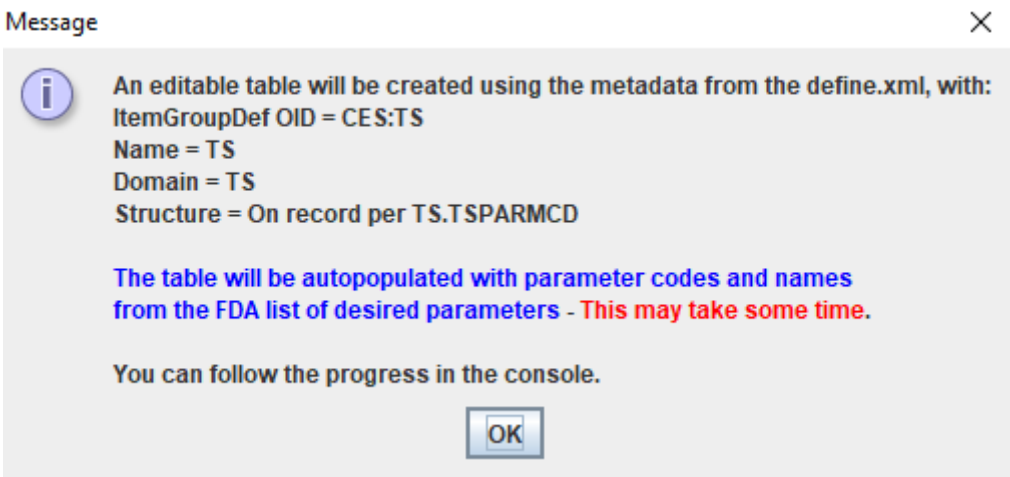
When now creating a new TS dataset, using the menu "Edit - Trial Design Dataset", and then selecting "New Trial Design Dataset", and selecting "TS" for the list, one will see that a new checkbox "Populate TS table with FDA desired TS Parameters" becomes available.

---

[3] I consider this bad design: personally, I would prefer separate domains for "planned" and for "actual".



When one then check it, and clicks "OK" (and one has already generated a "study-specific" instance of TS), an information message is shown:

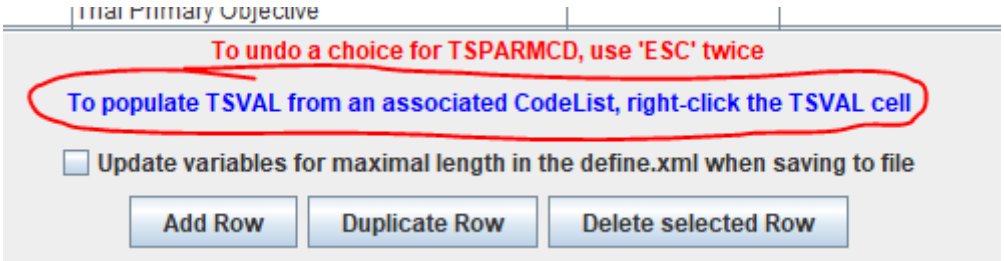


And after clicking "OK", the table is created and populated:

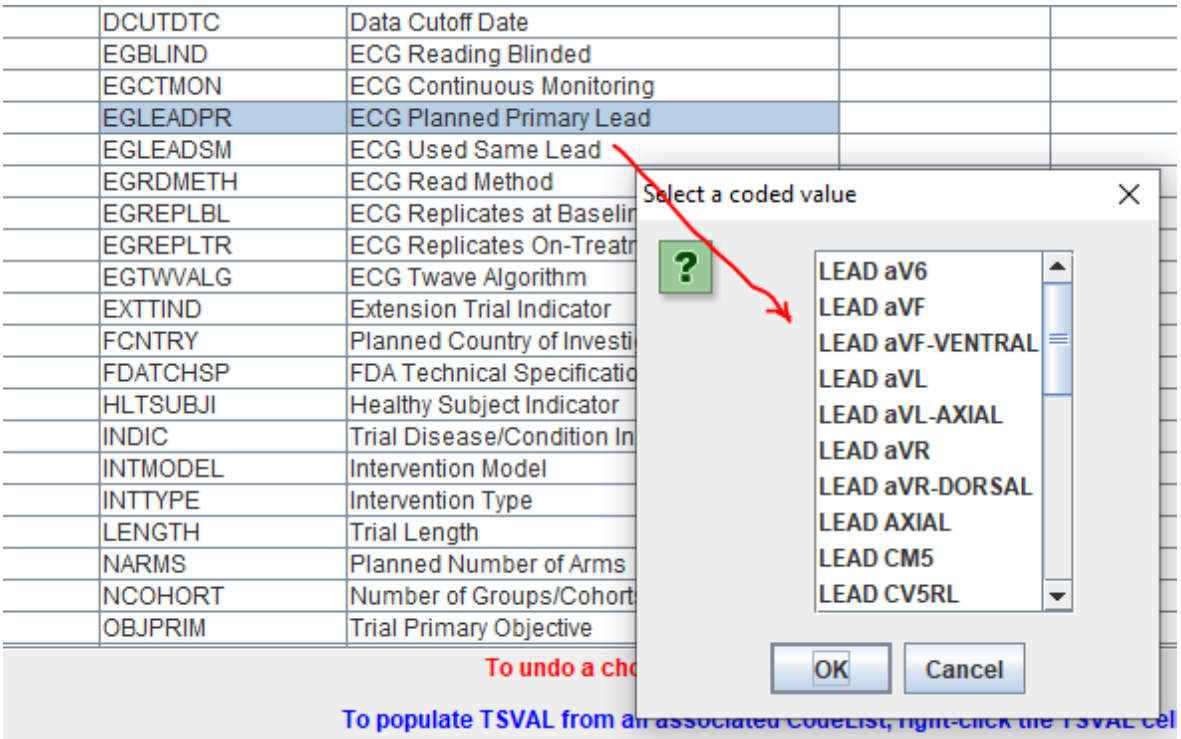
STUDYID	DOMAIN	TSSEQ	TSGRPID	TSPARMCD	TSPARM	TSVAL	TSVALNF	TSVALCD	TSVCDREF	TSVCDVER
CES	TS	1		ACTSUB	Actual Number of Subjects					
CES	TS	2		ADAPT	Adaptive Design					
CES	TS	3		ADDON	Added on to Existing Treatments					
CES	TS	4		AGEMAX	Planned Maximum Age of Subjects					
CES	TS	5		AGEMIN	Planned Minimum Age of Subjects					
CES	TS	6		COMPTRT	Comparative Treatment Name					
CES	TS	7		CRMDUR	Confirmed Response Minimum Duration					
CES	TS	8		CTAUG	CDISC Therapeutic Area User Guide					
CES	TS	9		CURTRT	Current Therapy or Treatment					
CES	TS	10		DCUTDESC	Data Cutoff Description					
CES	TS	11		DCUTDTC	Data Cutoff Date					
CES	TS	12		EGBLIND	ECG Reading Blinded					
CES	TS	13		EGCTMON	ECG Continuous Monitoring					
CES	TS	14		EGLEADPR	ECG Planned Primary Lead					
CES	TS	15		EGLEADSM	ECG Used Same Lead					
CES	TS	16		EGRDMETH	ECG Read Method					
CES	TS	17		EGREPLBL	ECG Replicates at Baseline					
CES	TS	18		EGREPLTR	ECG Replicates On-Treatment					
CES	TS	19		EGTWVALG	ECG Twave Algorithm					
CES	TS	20		EXTTIND	Extension Trial Indicator					
CES	TS	21		FCNTRY	Planned Country of Investigational Sites					
CES	TS	22		FDATCHSP	FDA Technical Specification					
CES	TS	23		HLTSUBJI	Healthy Subject Indicator					
CES	TS	24		INDIC	Trial Disease/Condition Indication					
CES	TS	25		INTMODEL	Intervention Model					
CES	TS	26		INTTYPE	Intervention Type					
CES	TS	27		LENGTH	Trial Length					
CES	TS	28		NARMS	Planned Number of Arms					
CES	TS	29		NCOHORT	Number of Groups/Cohorts					
CES	TS	30		OBJPRIM	Trial Primary Objective					

One can now start populating the table, add rows, delete rows, duplicate rows (when a parameter has more than one value) etc..

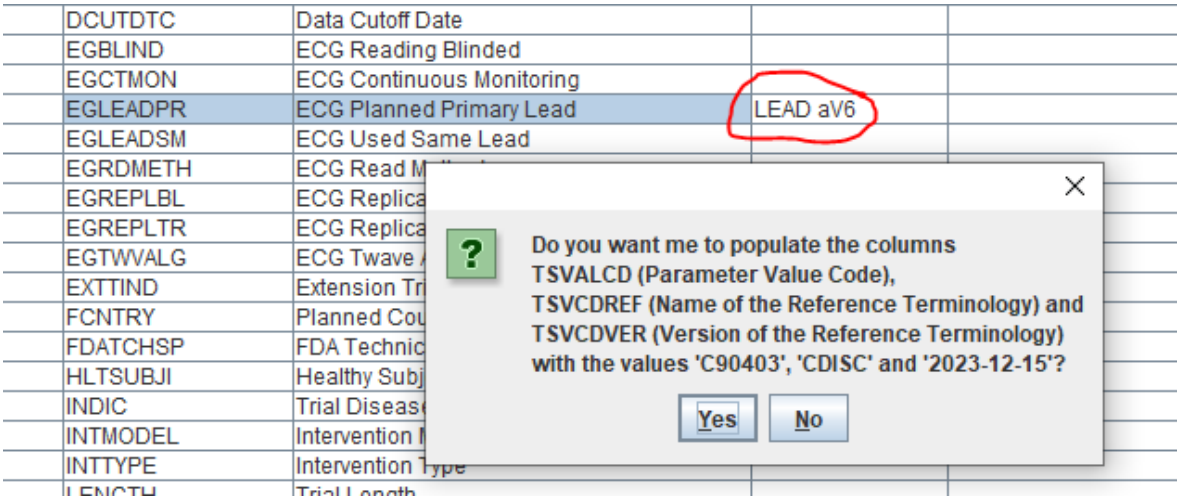
Important is also the information:



E.g. when one right-clicks the TSVAL cell for TSPARAMCD="ECG Planned Primary Lead", a list is presented containing all possible values of the planned primary ECG lead from the CDISC controlled terminology:



And when one has selected on e.g. "Lead aV6", a new dialog is presented:

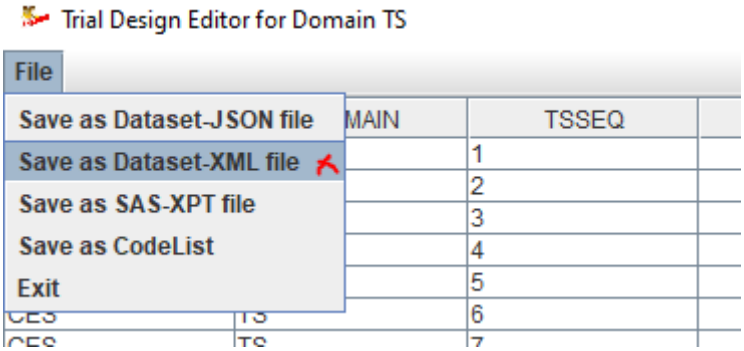


Inviting you that the system also auto-populates TSVALCD, TSVCDREF and TSVCDVER. These are then taken from the CDISC controlled terminology version selected when starting the mappings. In the above case, when clicking "Yes", the result is:

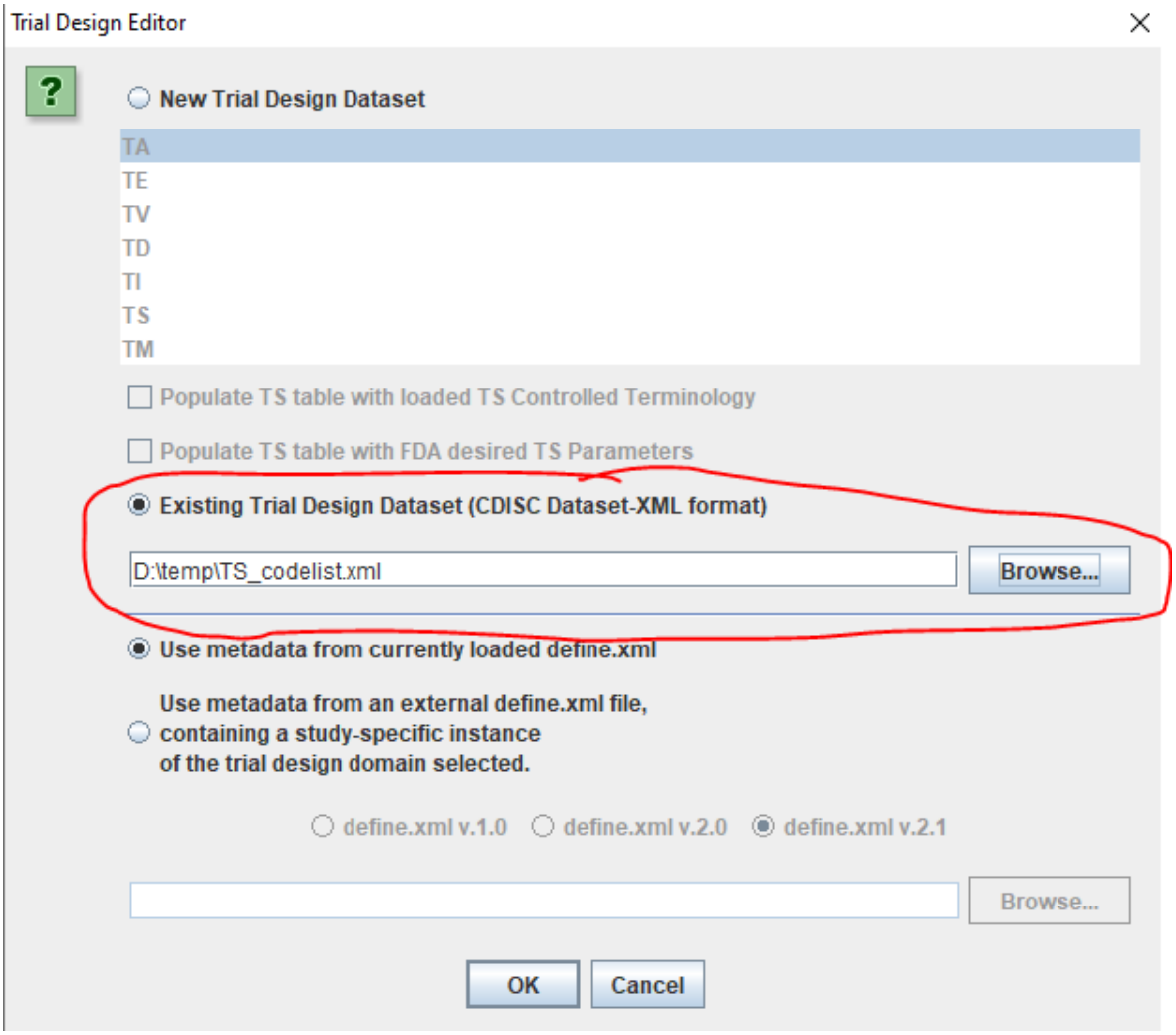
TSSEQ	TSGRPID	TSPARMCD	TSPARM	TSVAL	TSVALNF	TSVALCD	TSVCDREF	TSVCDVER
1		ACTSUB	Actual Number of Subjects					
2		ADAPT	Adaptive Design					
3		ADDON	Added on to Existing Treatments					
4		AGEMAX	Planned Maximum Age of Subjects					
5		AGEMIN	Planned Minimum Age of Subjects					
6		COMPTRT	Comparative Treatment Name					
7		CRMDUR	Confirmed Response Minimum Duration					
8		CTAUG	CDISC Therapeutic Area User Guide					
9		CURTRT	Current Therapy or Treatment					
10		DCUTDESC	Data Cutoff Description					
11		DCUTDTC	Data Cutoff Date					
12		EGBLIND	ECG Reading Blinded					
13		EGCTMON	ECG Continuous Monitoring					
14		EGLEADPR	ECG Planned Primary Lead	LEAD aV6		C90403	CDISC	2023-12-15
15		EGLEADSM	ECG Used Same Lead					
16		EGRDMETH	ECG Read Method					
17		EGREPLBL	ECG Replicates at Baseline					

Remark that it is not required to fill in all parameters at once. One can always save the table

to file as XML, and then reload later for further editing later. To do so, use "File - Save as Dataset-XML file":



When one then later wants to continue working on the TS dataset, in the first step, select "Existing Trial Design Dataset (CDISC Dataset-XML format), and then select the file one has saved before.

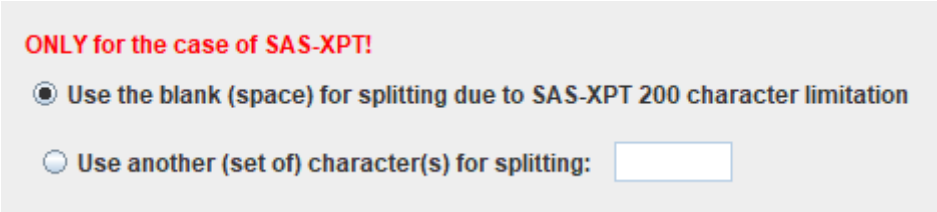


P.S. As soon as FDA will start accepting Dataset-JSON format instead of SAS-XPT, we will move to Dataset-JSON instead of Dataset-XML for intermediate storing of TS.

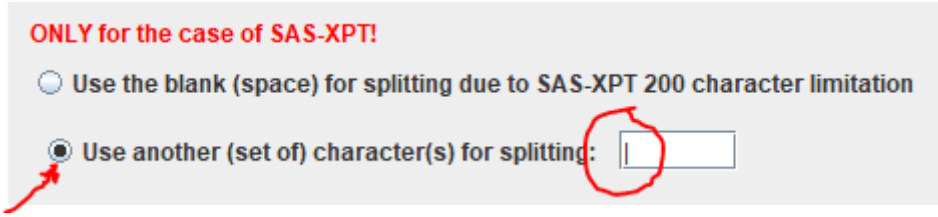
Split text in maximum 200 character pieces - split character

Unfortunately, FDA and other regulatory agencies still force us to submit datasets in the ancient SAS Transport 5 format, which is essentially a "digital punch card format". This format (from the time of IBM mainframes) has a limit of 200 (ASCII-only) characters for text values.
In case a value exceeds the 200-character limit, the SDTMIG requires us to store the 200 first characters in the normal way, and then put the (sets of) next 200 characters into the corresponding Supplemental Qualifier dataset[4], however, in such a way that words are not split somewhere in the middle.
This usually works well (and in SDTM-ETL in an automated way) when the "blank" character is used to "split" between words.
In SEND however, there are some variables (like EXTRT) where it is expected to use another character to separate different entries that are combined into a single variable. In such a case, there sometimes is no blank character, and the "splitting" will cause problems.

For the very seldom cases that one wants to use another character to "split" between words, there is now an option to indicate this. For using it, use the menu "Options - Settings", and then look for the section "Only for the case of SAS-XPT":



The default is to use the blank character for splitting between words. If one wants to use another character (or set of characters) for splitting, select the radiobutton "Use another (set of) character(s) for splitting, and fill in the desired character(s) in the text field, e.g.:



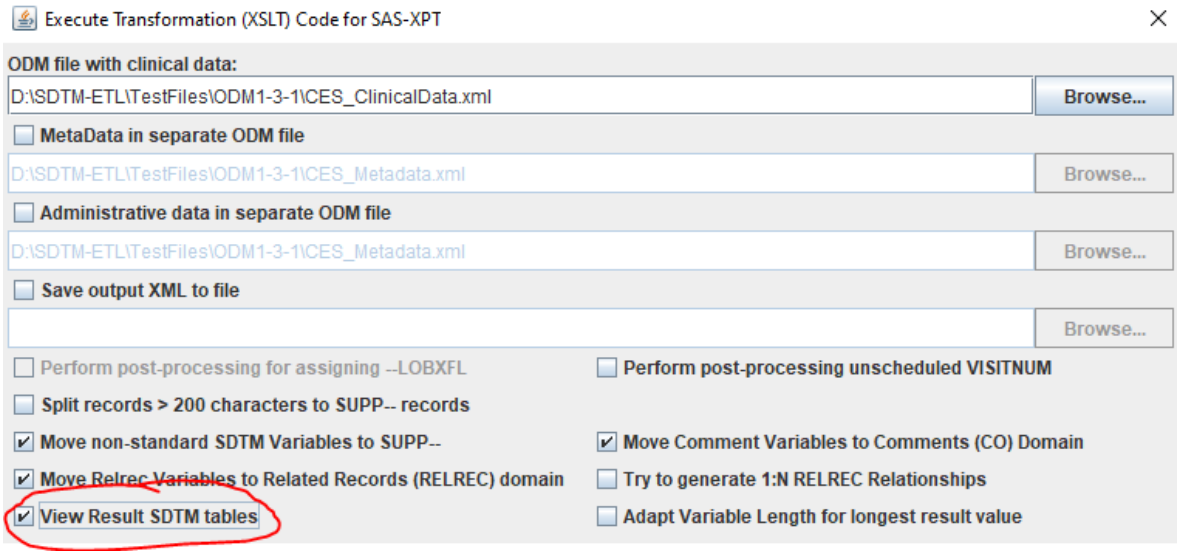
Where the vertical bar is selected as the "split character" when the value exceeds 200 characters.

IMPORTANT REMARK: This is only for the case that SAS Transport is used! For modern formats like Dataset-JSON, there is no such 200-character (nor any other length) limitation, and "banning" parts of submission values should not be done.

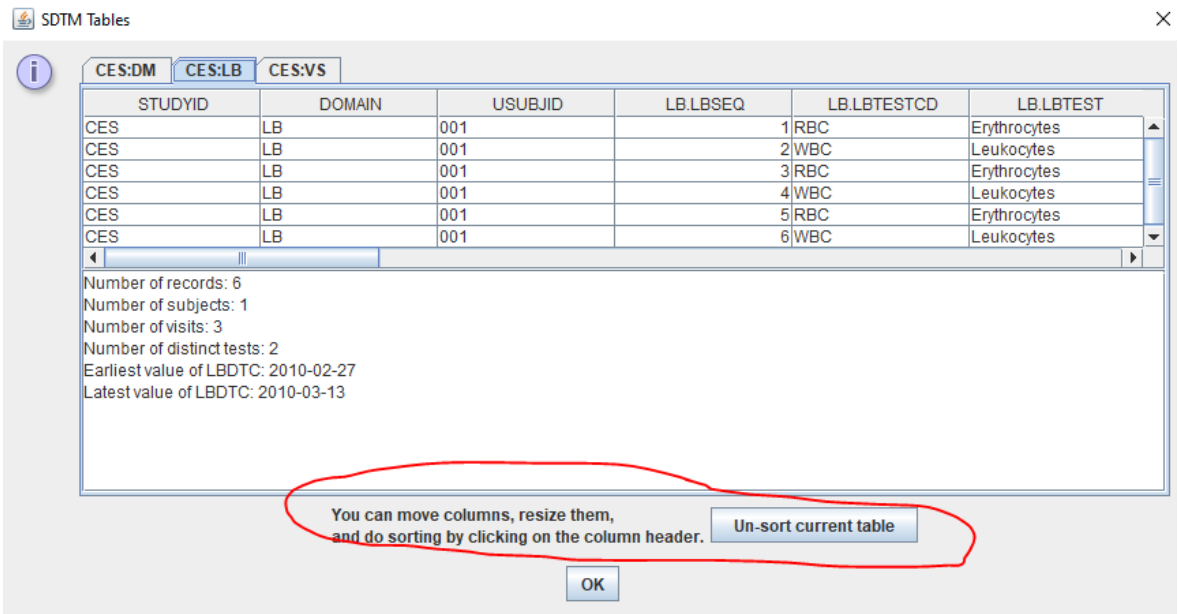
Results View in SDTM-ETL: new features

[4] See section 4.5.3.2 "Text Strings Greater than 200 Characters in Other Variables" in the SDTMIG-4.3.

When still developing the mappings, in most cases, one does not want to generate SAS-XPT files during testing all the time, as for visualization, this would require to start a "SAS Viewer" outside the application. Instead one wants to visualize the results within the SDTM-ETL application itself.
This is done by checking the checkbox "View Result SDTM Tables":



And when then clicking "Execute Transformation …", the results are visualized within the SDTM-ETL application itself:

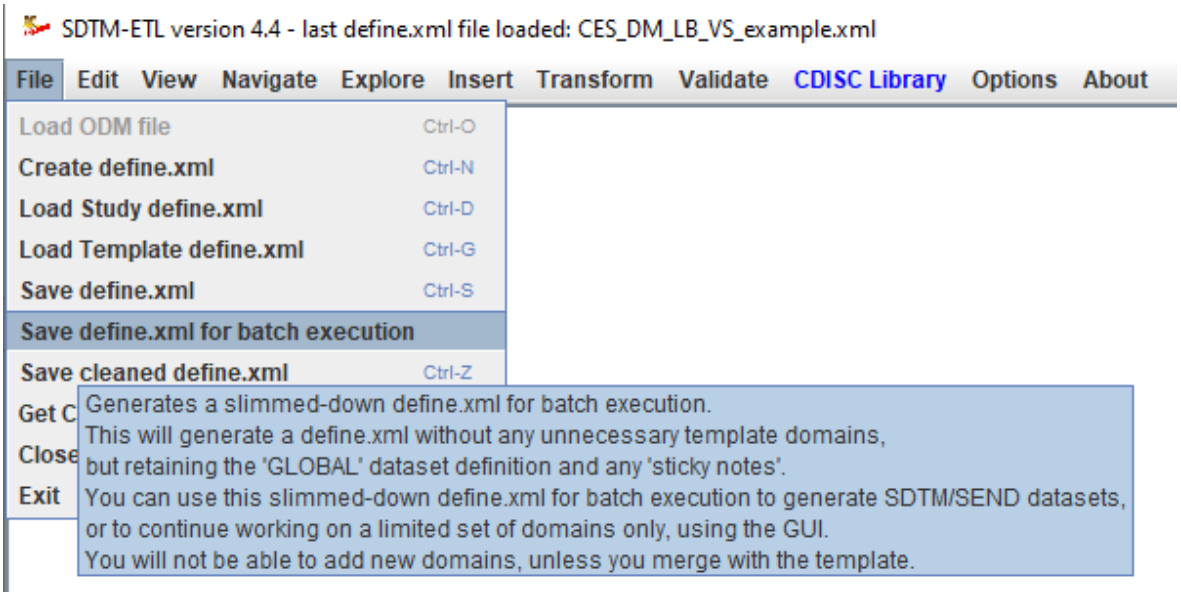


New in SDTM-ETL v.4.4 is that one can now move columns, and sort rows just by clicking on a column header. Going back to the original view (unsorted) can then be established by clicking the button "Un-sort current table".

Save define.xml for batch execution

Once the mappings are in good shape or even final, one will often want to execute them in "batch mode, i.e. without the use of the graphical user interface (GUI). See the tutorial "'SDTM-ETL Light' and running in batch execution mode". When doing so, the define.xml is loaded, including the "template rows" which are however not used by the batch execution engine. This may lead to slow execution behavior, especially when several define.xml files with embedded mappings are used.
Therefore, we developed a new feature to "slim down" the define.xml files with mappings, removing the "template rows", i.e. only the "study-specific" dataset definitions are retained.

Such a "slimmed down for batch execution" define.xml can be generated using the menu "File - Save define.xml for batch execution":



For more details, see the tutorial "Save define.xml for batch execution"

Additional filtering on "looping" variables

When developing mappings, one will usually first provide the mapping for the so-called "looping variable" which usually is the "–TESTCD" variable in the case of a Findings domain, "–TERM" in the case of an Events domain and "–TRT" in the case of an Interventions domain.

Essentially, when developing the mapping for the "looping variable", one selects which data points in the source (the ODM) are used for generating the dataset. Usually, this is done using the wizards after "drag-and-drop", using the "Generalize for …" with "Only for … " and "Except for …" filter buttons (see several of the tutorials on our website).

The selection then results in an "xpath(…)" statement in the mapping script, which, under circumstances, can become pretty complicated.
So, some of our users asked us whether one can do this in steps …

As of SDTM-ETL v.4.4 this is now possible, using the "xpathfiler()" function. For example:

```
8  # with CodeList OID 'CL.C66741.VSTESTCD'
9  $CODEDVALUE = xpath(/StudyEventData/FormData[@FormOID='F_BASELINE' or @For
10 $CODEDVALUE = xpathfilter($CODEDVALUE,"[not(@Value='M')]");
11 if ($CODEDVALUE == 'I_HEIGHT') {
12   $NEWCODEDVALUE = 'HEIGHT';
13 } elsif ($CODEDVALUE == 'I_WEIGHT') {
14   $NEWCODEDVALUE = 'WEIGHT';
15 } elsif ($CODEDVALUE == 'I_SYSBP') {
16   $NEWCODEDVALUE = 'SYSBP';
17 } elsif ($CODEDVALUE == 'I_DIABP') {
```

Where line 9 filters out those records for which the ODM value is "M".

Much more is possible, for further details please see the separate tutorial "Additional filtering on 'looping' variables"

SUPP-- datasets: QORIG

For "Supplemental Qualifier" (SUPP–) datasets, the QORIG variable is "Required". Essentially, this is nonsense, as "Origin" is metadata, which must go into the define.xml. For SUPP– this can easily be accomplished by define.xml "ValueLists".
It looks as the developers of SDTM have little of no knowledge about define.xml, otherwise they would have not come to the (i.m.o. stupid) idea of making QORIG "Required". Or it must be that this is again one of these crazy requests of the FDA, to make life of the reviewers "easier", allowing to ignore the define.xml.

However, such stupidities cannot be undone, so, for the case of "automatically generated SUPP–" datasets, either by "moving non-standard variables to SUPP–" or due to splitting of text values longer than 200 characters, we needed to do something.
For the case of "Non-standard variables" (NSVs), the "Origin" from the define.xml is taken, and copied to QORIG. If Define-XML 2.1 is used, QEVAL is then taken from "Source".
For example:

Edit Properties for SDTM Variable VS.VSNSV

OID:	VS.VSNSV
New OID	Edit
Name:	VSNSV
SASFieldName:	
Data type:	text
Current Length:	20
New Length:	20
Current Significant Digits:	
New Significant Digits:	-1
Current Role:	SUPPQUAL
New Role	SUPPQUAL
Current Role CodeList:	
New Role CodeList	CL.C66742.Y - No Yes Response (Yes only) (text)
Current Origin/Source:	Assigned/Sponsor
Edit Origin/Source:	Edit
Comment:	
External document for comment	
Current CodeList	NO CODELIST ASSIGNED
New CodeList:	Select CodeList
Description:	Example Non-Standard Variable

Leading to, for QORIG in SUPPVS:

SDTM Tables

CES:DM | CES:LB | CES:VS | CES:SUPPVS

AL	QNAM	QLABEL	QVAL	QORIG	QEVAL
	VSNSV	Example Non-Standard Variable	test	ASSIGNED	SPONSOR

When no source/origin is provided from the NSV definition, or the SUPP– record is due to the "200 character splitting" then QORIG will be populated with "CRF".
However, the value in the define.xml (sometimes through "ValueList") is much more important.

Extended features for "Mapping Completeness"

Even more than software validation is what we call "result validation". This is especially the case for SDTM-ETL, as it is software to categorize data, combine data, and sometimes derive data, i.e. a typical ETL (Extract, Transform and Load) process. This means that even with a perfect software, when the user makes the wrong mapping decisions, "garbage" will be produced.

An important aspect of this is "mapping completeness". Mappers must always ask themselves:
- Did I include all (types of) datapoints that need to be included[5]?
- Did I include all the visits?
- Did I at least have mappings for all "required" and "expected" variables?
- Did I include all tests for this domain?
- Have (coded) values from the source been mapped to the applicable CDISC Controlled terminology when this is required?

SDTM-ETL already provide a lot of features for checking all these. For example, the SDTM

[5] Rememer that answers for some questions like "Did any adverse events occur" will not appear in the SDTM.

"table" in the GUI has cells that are color-coded: red for "required", blue for "expected" and green for "permissible" variables.
As explained in other tutorials, earlier versions already allowed to quickly find out which ODM "items" are used in which mappings, and to generate a "mapping completeness report", showing for each ODM item, in which SDTM/SEND variables it has been used, and how.
See the website for further details and tutorials.

In version 4.4, we have further extended these features. When one now generates SDTM datasets, and visualizes within the application (checkbox "View Result SDTM tables" or View Result SEND tables"), not only the result tables themselves will be shown, but also some summary information about the contents:
- Number of records
- Number of subjects
- Number of visits covered
- Number of distinct tests (in the case of Findings domains), treatments (in the case of Interventions domains) or number of distinct terms (in the case of Events domains)
- Earliest (start) date
- Latest (start) date
- Earliest (end) date
- Latest (end) date.

For example:

SDTM Tables

My Study:DM | My Study:QS | My Study:SV | My Study:PE | My Study:AE | My Study:VS

STUDYID	DOMAIN	USUBJID	QS.QSSEQ
MyStudy	QS	001	1
MyStudy	QS	001	2
MyStudy	QS	001	3
MyStudy	QS	001	4
MyStudy	QS	001	5
MyStudy	QS	001	6
MyStudy	QS	001	7
MyStudy	QS	001	8
MyStudy	QS	001	9
MyStudy	QS	001	10
MyStudy	QS	001	11
MyStudy	QS	001	12
MyStudy	QS	001	13
MyStudy	QS	001	14
MyStudy	QS	001	15
MyStudy	QS	001	16
MyStudy	QS	001	17
MyStudy	QS	001	18
MyStudy	QS	001	19
MyStudy	QS	001	20
MyStudy	QS	001	21
MyStudy	QS	001	22
MyStudy	QS	001	23
MyStudy	QS	001	24
MyStudy	QS	001	25
MyStudy	QS	001	26
MyStudy	QS	001	27

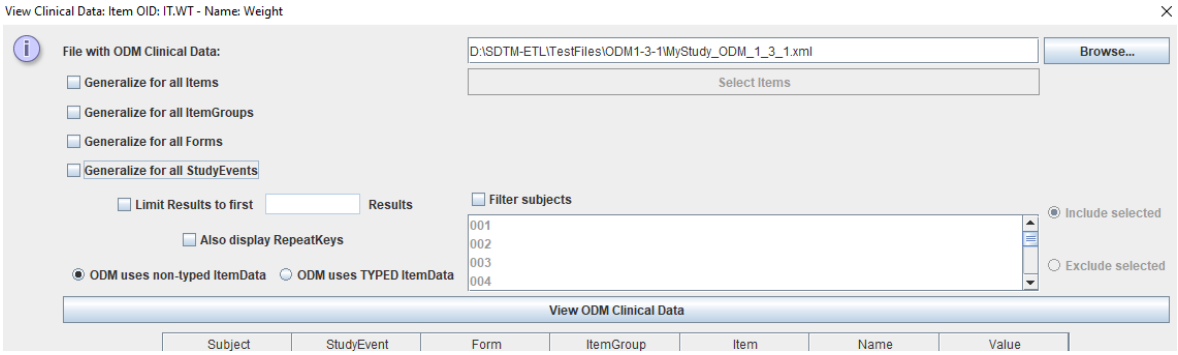
Number of records: 3143
Number of subjects: 12
Number of visits: 2
Number of distinct tests: 85
Earliest value of QSDTC: 2006-04-01
Latest value of QSDTC: 2006-05-12

Especially important than is to check the number of subjects ("did I cover all subjects?"), number of visits ("Did I cover all visits?"), and the number of distinct tests (Did I include all tests for this domain?"). Also earliest and latest dates give an indication about whether everything within the study period has been covered.

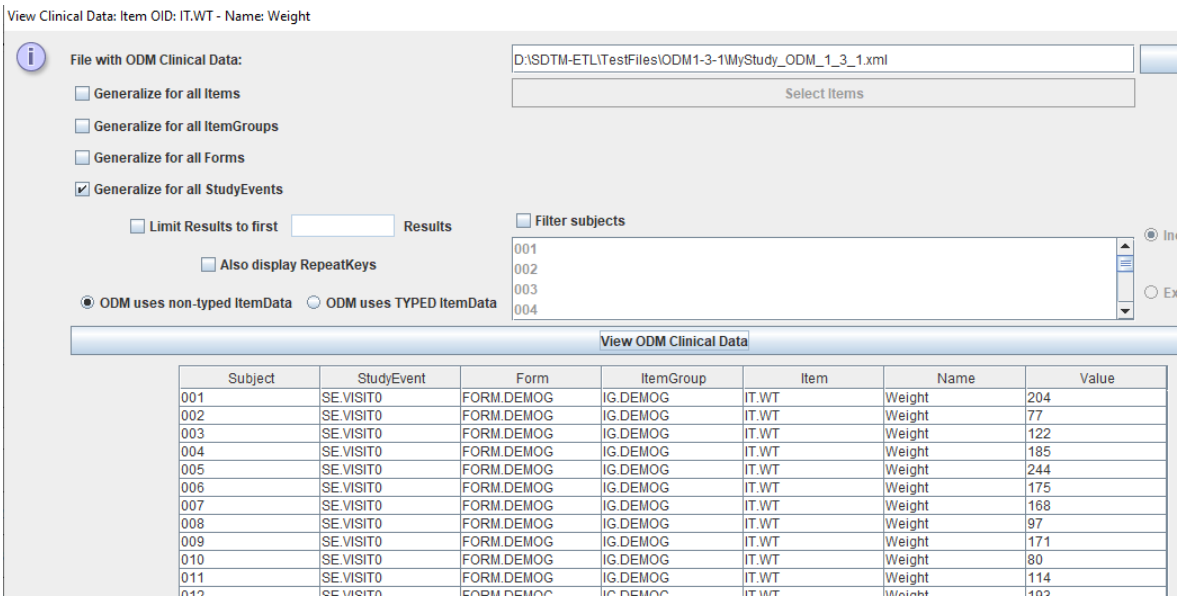
Visualization of collected data: choice of items

One of the highly appreciated features of SDTM-ETL by our users is the ability to check the data from the ODM "ClinicalData" part from within the application. This very often allows them to better understand what the data is about, whether it is coded or not, etc..

When, after selecting an Item from the ODM tree", using the menu "View - ODM ClinicalData is used", the following dialog is displayed:



In this case, for inspecting the "Weight" values from the ODM.
Normally, this would then be limited to the currently selected visit, but one can "generalize" this for all the visits by checking the "Generalize for all StudyEvents" checkbox.
When then clicking "View ODM Clinical Data", one e.g. obtains:



Subject	StudyEvent	Form	ItemGroup	Item	Name	Value
001	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	204
002	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	77
003	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	122
004	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	185
005	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	244
006	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	175
007	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	168
008	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	97
009	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	171
010	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	80
011	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	114

One can then also obtain the values from all other items in the same group by checking the checkbox "Generalize for all Items", e.g. leading to:

View Clinical Data: Item OID: IT.WT - Name: Weight

File with ODM Clinical Data: D:\SDTM-ETL\TestFiles\ODM1-3-1\MyStudy_ODM_1_3_1.xml

Select Items

- [x] Generalize for all Items
- [ ] Generalize for all ItemGroups
- [ ] Generalize for all Forms
- [x] Generalize for all StudyEvents
- [ ] Limit Results to first Results
- [ ] Also display RepeatKeys
- [ ] Filter subjects

001
002
003
004

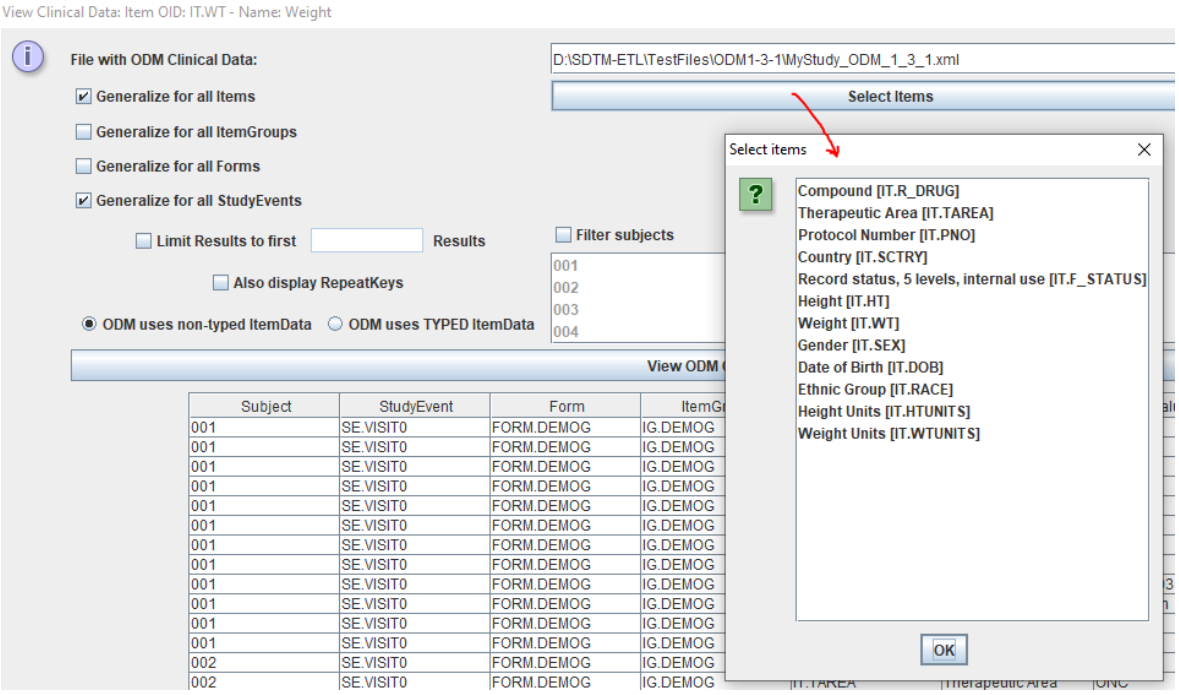
ODM uses non-typed ItemData / ODM uses TYPED ItemData

View ODM Clinical Data

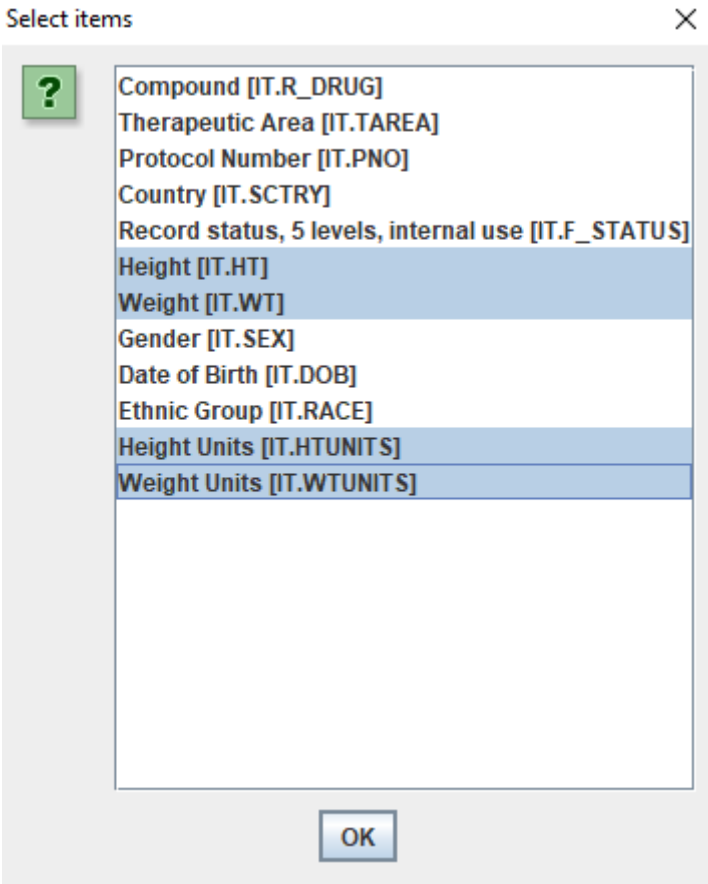
Subject	StudyEvent	Form	ItemGroup	Item	Name	Value
001	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.R_DRUG	Compound	SDP
001	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.TAREA	Therapeutic Area	ONC
001	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.PNO	Protocol Number	143-02
001	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.SCTRY	Country	USA
001	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.F_STATUS	Record status, 5 lev...	V
001	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.HT	Height	73
001	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	204
001	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.SEX	Gender	3
001	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.DOB	Date of Birth	1960-04-03
001	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.RACE	Ethnic Group	Caucasian
001	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.HTUNITS	Height Units	in
001	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WTUNITS	Weight Units	lb
002	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.R_DRUG	Compound	SDP
002	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.TAREA	Therapeutic Area	ONC
002	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.PNO	Protocol Number	143-02
002	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.SCTRY	Country	USA
002	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.F_STATUS	Record status, 5 lev...	S

Which however may be "overkill" as also e.g. the value for "Compound" is provided.

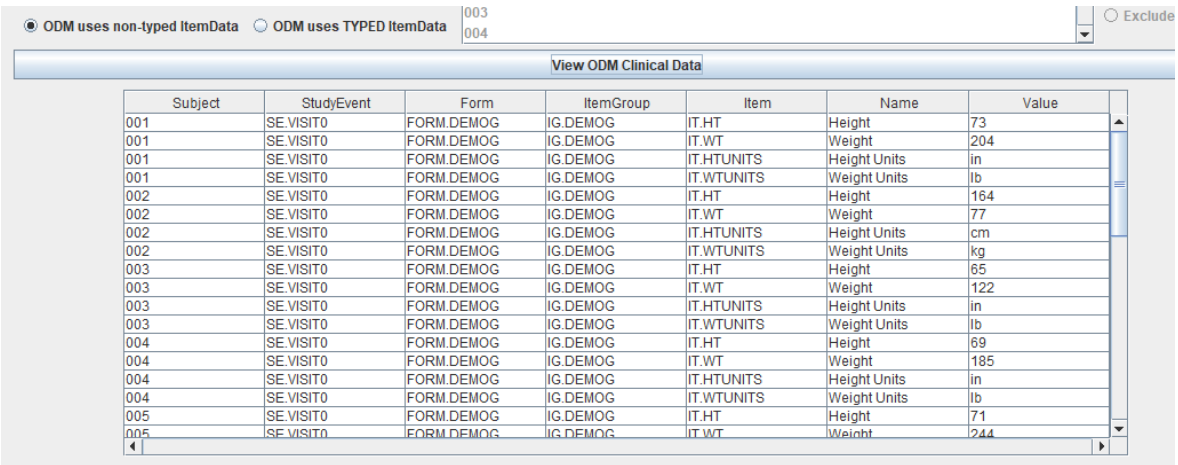
New in SDTM-ETL 4.4 is that one can now select <u>which</u> items in the group of the clinical should be displayed. This can be accomplished by clicking the new button "Select Items", e.g. leading to:



And selecting the one of interest for the user, e.g.:



leading to:



Subject	StudyEvent	Form	ItemGroup	Item	Name	Value
001	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.HT	Height	73
001	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	204
001	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.HTUNITS	Height Units	in
001	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WTUNITS	Weight Units	lb
002	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.HT	Height	164
002	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	77
002	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.HTUNITS	Height Units	cm
002	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WTUNITS	Weight Units	kg
003	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.HT	Height	65
003	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	122
003	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.HTUNITS	Height Units	in
003	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WTUNITS	Weight Units	lb
004	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.HT	Height	69
004	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	185
004	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.HTUNITS	Height Units	in
004	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WTUNITS	Weight Units	lb
005	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.HT	Height	71
005	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	244

Making it clear that some of the height were captures with units of inches, other in cm, and for weight some in pounds, others in kg.
This then makes the user aware that some unit conversions will be necessary for VSSTRESC / VSSTRESN.

Further refined treatment of "Unscheduled Visits"

A hot topic is always the treatment of "unscheduled visits", i.e. visits that take place between

two "planned" visits. Also these "unscheduled" visits require to obtain a "VISITNUM" which will howewever not appear in the TV "Trial Visits" (Trial Design) datasets. For VISITNUM, the SDTMIG has some rules:

Clinical encounters are described by the CDISC visit variables. For planned visits, values of VISIT, VISITNUM, and VISITDY must be those defined in the Trial Visits (TV) dataset (see Section 7.3.1, Trial Visits). For planned visits:

- Values of VISITNUM are used for sorting and should, wherever possible, match the planned chronological order of visits. Occasionally, a protocol will define a planned visit whose timing is unpredictable (e.g., planned in response to an adverse event, a threshold test value, or a disease event), and completely chronological values of VISITNUM may not be possible in such cases.
- There should be a one-to-one relationship between values of VISIT and VISITNUM.
- For visits that may last more than 1 calendar day, VISITDY should be the planned day of the start of the visit.

For "unplanned/unscheduled" visits, it also provides some information about possible approaches:

Sponsor practices for populating visit variables for unplanned visits may vary.

- VISITNUM should generally be populated, even for unplanned visits, as it is expected in many Findings domains, as described above. The easiest method of populating VISITNUM for unplanned visits is to assign the same value (e.g., 99) to all unplanned visits, although this method provides no differentiation between the unplanned visits and does not provide chronological sorting. Methods that provide a one-to-one relationship between visits and values of VISITNUM, that are consistent across domains, and that assign VISITNUM values that sort chronologically require more work and must be applied after all of a subject's unplanned visits are known.
- VISIT may be left null or may be populated with a generic value (e.g., "Unscheduled") for all unplanned visits, or individual values may be assigned to different unplanned visits.
- VISITDY must not be populated for unplanned visits; VISITDY is, by definition, the planned study day of visit. The actual study day of an unplanned visit belongs in a --DY variable.

Interesting is the wording (a bit hilarious …) "may vary" …

Essentially, VISITNUM is only present in SDTM/SEND, as it looks as reviewers are uncapable to sort data based on the visit name and the start- and end-date information in the SV (Subject Visits) datasets[6].

One of the approaches is to assign VISITNUM by sorting the SDTM/SEND data chronologically, and then, for the "unscheduled visits" assign a VISITNUM value as a decimal number, with a value between the integer numbers of the prior planned visit number (an integer) and the next planned visit number (also an integer).

For example, when the prior visit is "VISIT 2" with VISITNUM=2, and the next planned visit is "VISIT 3" with VISITNUM=3, then unplanned visits will get VISITNUM=2.1, VISITNUM=2.2 etc..

A set of new algorithms for making this possible in a "post-processing" step has now been implemented in SDTM-ETL 4.4. It requires that the data is in chronological order (which is mandated by the ODM specification, but sometimes violated), or that the the –DTC variable (or –SDTDTC) is assigned as one of the "key variables" in the define.xml, which can easily be achieved using the menu "Edit - SDTM/SEND Variable Properties" (CTRL-E).

---

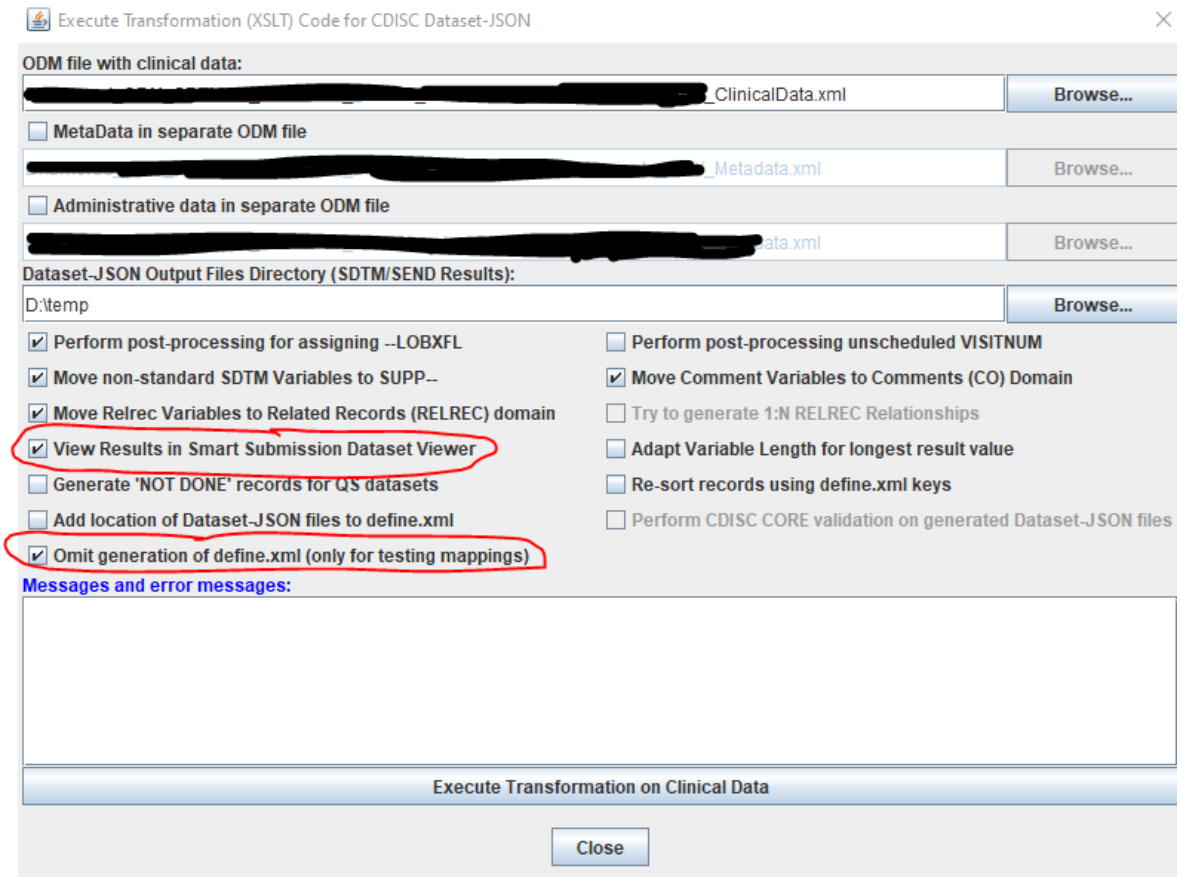
[6] Sometimes I have the impression that reviewers cannot combine information from different datasets anyway, explaining the (ever growing) data redundancy in SDTM and SEND.

All the possibilities and options for using this new feature are described in the separate tutorial "Handling unscheduled visits".
REMARK: The user is always free to use its own method of assigning VISITNUM for unscheduled visits by providing a mapping. There is no obligation at all to use this new feature.

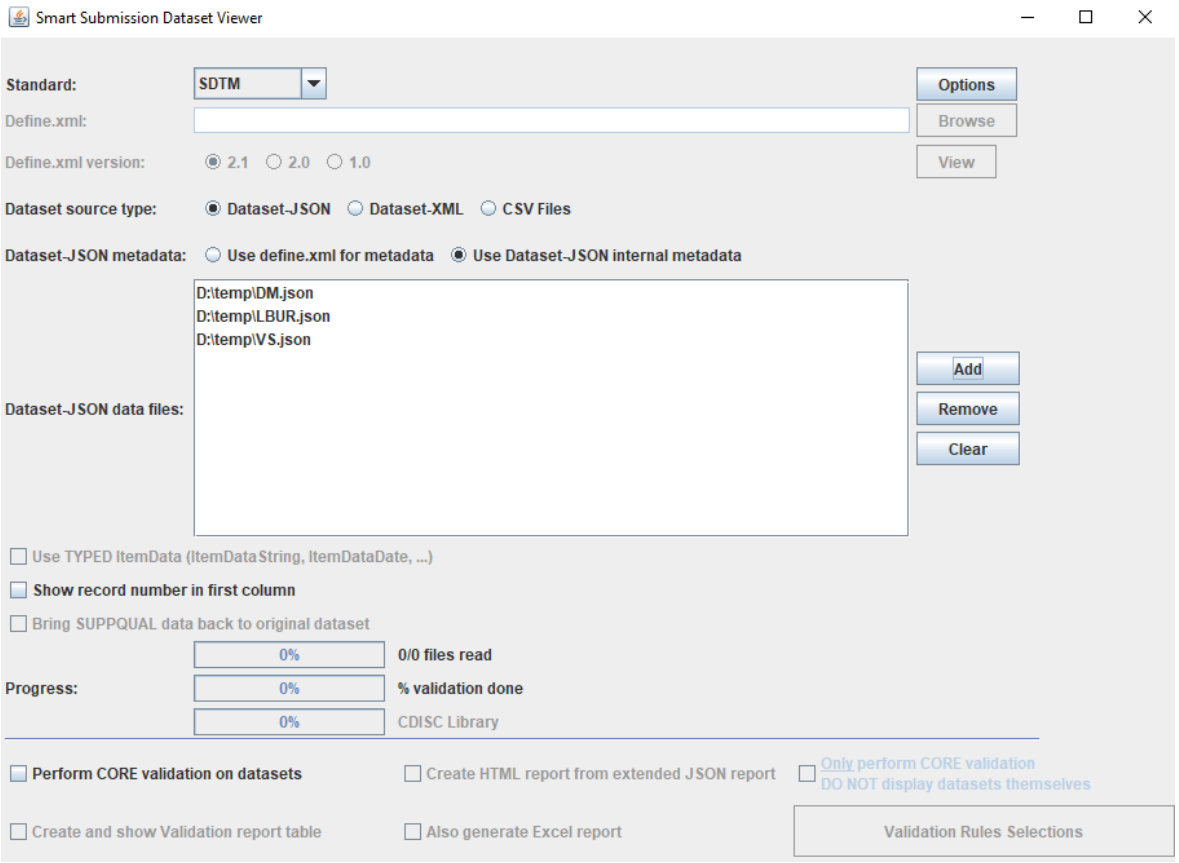
More features for visualization for the case of Dataset-JSON format

We expect that FDA will start accepting submissions in the new CDISC Dataset-JSON format (replacing the antiquated SAS Transport (XPT) format) later this year or early next year.
This also means that we want to make SDTM-ETL "Dataset-JSON fit".

When choosing for Dataset-JSON as the format for the generated datasets, the user is now already invited to use the "Smart Submission Dataset Viewer" for the visualization. This has great advantages, as this viewer is "smart" …
The disadvantage is that it takes more time as also a "cleaned-up" define.xml is generated.
The latter is however not always necessary when just testing the developed mappings, also as Dataset-JSON itself has some, but limited amount, of metadata within the Dataset-JSON files itself.
Therefore, we now added the option to omit the generation of a define.xml into the output folder where the datasets are written:



When then executing the transformation, no define.xml is generated and written to the output folder, only Dataset-JSON files. This information is also passed to the viewer, and the radiobutton "" in the viewer is automatically selected:



Remark also that when no define.xml is generated in the output folder, no CDISC CORE validation is possible[7].

New startup parameter in "properties.dat"

The file "properties.dat" contains a set of "start-up" parameters that are read in when the SDTM-ETL software is started. For example, it allows to state that when an ODM file is loaded, validation of the ODM can be skipped or not, as this is typical something that one will want to do only the first time when one works with this ODM file.
It e.g. also allows to add the key for use of ChatGPT and/or the CDISC Library API (see other tutorials on our website).

When executing the mapping scripts on ODM files with clinical data, there are two "flavors" of "ItemData" in the ODM "ClinicalData": "untyped" (classic) and "typed".
Most EDC vendors (about 80%) use "untyped ItemData", but also some (20%) like Viedoc, use "typed ItemData" (e.g. <ItemDataDate ItemOID="…">2023-02-07</ItemData>.

The new parameter "odmtypeditemdata" allows to say to the software that "typed ItemData" is to be used (the default is "false"). This e.g. allows Viedoc users to set this for once, and do not explicitly set this in the GUI using the radiobutton.

[7] The reason of this is that some CORE rules require a lookup into the define.xml.

```
skipodmvalidation=true
# As of SDTM-ETL v.4.4: for EDC systems that export ODM in "Typed ItemData" format
odmtypeditemdata=true
# As of SDTM-ETL v.4.4: set user-defined "default" mapping descriptions
adddefaultmappingdescriptions=true
# postpone ODM tree recalculation after loading a define.xml
postponeodmtreenoderecalculation=false
# set number of minutes between define.xml autosave
numminutesforautosave=15
```

A second new parameter is "adddefaultmappingdescriptions", allowing to state that "default mapping descriptions" should always be added (as explained before - see section "default mapping descriptions"). The default is "false".
For both, the choices can always be set or changed using the menu "Options - Properties".

New mapping script language functions

On request of a number of our customers, we have added some new "date/time" functions to the mapping script language. These are also documented in the document "Mapping Script Language Specification" (available on request). These functions are:

Function	Description	Example
dateadd()	Returns a date (ISO-8601) by adding an ISO-8601 "duration" to an existing date (ISO-8601 format)	$twodayslater = dateadd($BIRTHDATE,'P2D');
datetimeadd()	Returns a datetime (ISO-8601) by adding an ISO-8601 "duration" to an existing datetime (ISO-8601 format)	$oneyeartwosecondslater = datetimeadd($RFXSTDTC, 'P1YT1S');

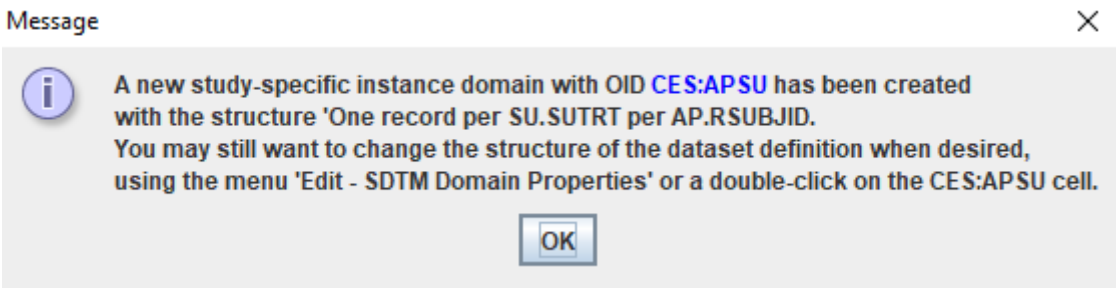
Also remark (once again) that users can easily develop and add new functions. These can be added to the file "functions.xsl". Developing new functions does however requires some knowledge of XSLT.

Bug fixes

- Automated generation of –LOBXFL based on the combination of –TESTCD, –CAT, –SCAT etc. was not supported in SDTM-ETL v.3.3. This has now been fixed.

- When generating Dataset-JSON or Dataset-XML files, with the option "View Results in Smart Submission Dataset Viewer", also empty files (like RELREC) were passed, and listed in the GUI of the Smart Submission Dataset Viewer. This could cause problems when processing such empty files in the viewer.
Fix: empty files are not passed to the Smart Submission Dataset Viewer anymore

- When using the automated generation of Supplemental Qualifier datasets for "non-standard variables", and using SAS Transport as the output format, the value for RDOMAIN in the SUPPAPxx dataset was truncated to two characters, i.e. "AP". Also, under circumstances, the "Structure" (define.xml "def:Structure") was not correctly assigned. These have now been fixed.
Furthermore, a message will now be displayed after the AP-domain instance has been created, e.g.:



- SAS Transport 5 generation failed in the (seldom) case that a non-standard variable (NSV) that is "banned" to SUPPxx, was declared as not being of data type "text" or "integer" or "float". For example, when an NSV was declared of being of data type "date", the system could not find a suitable value for the field length of QVAL in the SUPPxx dataset. This only happened for SAS Transport as the result format, due to SAS-XPT being a "fixed field length" format, similar to in punch cards. This has now being fixed by assigning suitable field lengths for "date", "datetime", "incompleteDatetime", etc. data types for NSVs.
NSVs are however usually (99% of the cases) being assigned the data type "text".

- The function "day-in-week" caused an error when the argument was a (ISO-8601) date and not an datetime. This has been fixed.
Also, the function will now return "-1" when the argument is not a valid date or datetime.

Experimental: Batch Execution for output in Dataset-JSON format

We expect that the FDA will start accepting SDTM/SEND submissions in the new CDISC Dataset-JSON format by the end of this year. This will be a huge step forward, leading to considerable time and money savings in the generation of submissions, and (though the possibility of using APIs and e.g. RESTful Web Services) may lead to much earlier and higher quality submissions. This may result in marketing authorizations 1-2 years earlier.

Therefore, we have put a lot of effort in getting everything right in generating results in Dataset-JSON format, as well using the Graphical User Interface as for batch execution.

Batch execution will become more and more important in future.
It is expected that in future, sponsors, service providers and regulatory authorities will not exchange SDTM and SEND anymore using "files", but is APIs. Whether the SDTM/SEND is then stored as files, in a database, or any other way, will not be important anymore.
For the use with APIs, JSON is ideal, and one even may think about SDTM-ETL not producing "files" anymore, but directly sending/storing the generated data(sets) somewhere else (e.g. in a repository) using the API.

Limitations of v.4.4

For batch execution using the new CDISC Dataset-JSON, not all combination of parameters have been thoroughly tested yet. For example, automated generation of RELREC records from "SDTM Variable for RELREC" variables has not been implemented yet.
For XPT format, it works perfectly in batch execution mode.

As Dataset-JSON will become important (see next section), we aim to have implemented all parameters of the batch execution mode for Dataset-JSON by the next version, or as an intermediate patch.

Further development of SDTM-ETL

We expect that FDA, with other regulatory authorities following, will soon accept submissions in the modern CDISC Dataset-JSON format, as this format has an enormous advantage over SAS-XPT, also for the FDA.
Once FDA formally accepts Dataset-JSON, we will release a version 5.0 of the software, where Dataset-JSON is the default output format. Further development efforts will then also concentrate on output in this format.

We will however keep supporting output in SAS Transport 5 (XPT) format as long as FDA and other regulatory authorities allow submissions in this format, as we realize that not every sponsor, CRO and service provider will want to make the transition immediately.

Once Dataset-JSON well established, we will discontinue output in Dataset-XML, as it essentially will become obsolete. We can however keep Dataset-XML output for customers who desire it (e.g. for academic studies).