SDTM-ETL 5.0 User Manual and Tutorial
Creating a cleaned define.xml

Author: Jozef Aerts, XML4Pharma

Last update: **2025-02-29**

Table of contents

Introduction

In SDTM-ETL, a define.xml is used operationally to store all mappings, domain/dataset and variable information, and codelists and valuelists.

At a certain moment, one will however want to generate a define.xml that can be used for submission to the regulatory authorities, so "cleaned".

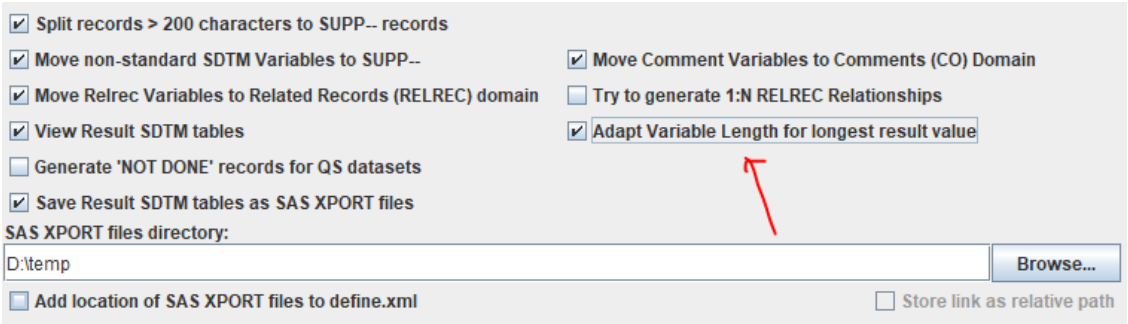
Generating a "cleaned" define.xml

When doing so, one must first ensure that:

- you have mappings for all variables that are "required" and "expected"[1]
- for the latter, if there is no data at all (which should be the exception), the mappings script can be just the statement that the value is "empty", e.g. $LB.LBLOINC="";
- you have generated subsetted codelists and also assigned them to your variables where appropriate. For example, it does not make sense to submit the complete codelists for LBTESTCD and LBTEST (over 2000 terms), when you only had 10 distinct lab tests. Similar for the "UNIT" codelist which currently has over 700 terms[2]. In the case of units, one will probably want to generate more than one subset, e.g. one for LBORRESU/LBSTRESU and one for CMDOSU (dosing unit) as these are completely different things.
  For generating codelist subsets, see the tutorials "Subsetting and Extending Codelists". Subsetting of codelists can also be done when using the codelist-codelist mapping wizard, see the tutorial "Using the CodeList Mapping Wizard".
- especially in the case of having to generate datasets in SAS Transport format, you have assigned correct data types and variable lengths. The maximal length of a variable can be assigned manually using the menu "Edit - SDTM/SEND Variable Properties" or at dataset

[1] Even when there is no data for an "expected variable", regulatory authorities require that the column must still be present (but will then be empty) in the submission dataset.
[2] Unfortunately, CDISC still refuses to move to UCUM notation for units, which is used all over the world in healthcare informatics, except for ... clinical research.

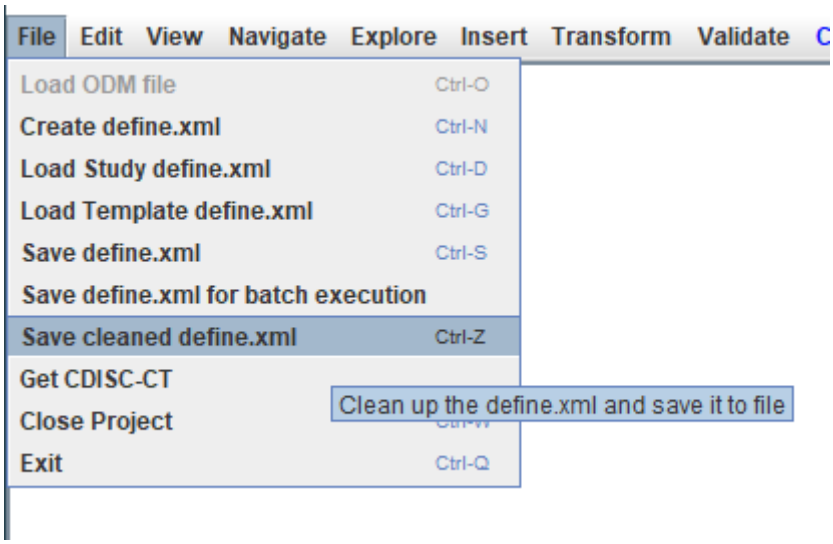
generation time itself using the checkbox "Adapt Variable Length for longest result value"



- you have added valuelists where needed and useful. Remember that valuelists are there for making review by the (regulatory) reviewer easier. Except for a few exceptions (e.g. SUPPQUAL datasets), there is no formal obligation for adding valuelists. For example, it is up to you to decide whether it makes sense to generate a valuelist for VSORRESU and VSSTRESU when you only have "height", "weight" and "body mass index", and the unit for "height" is always "cm", the unit for "weight" is always "kg" and the unit for "body mass index" is always "kg/m2". If there were choices on the CRF for the unit however, for example, between "cm", "m", "inches" for "height", you might want to develop a valuelist for VSORRESU, where you e.g. state that "WHERE VSTESTCD=HEIGHT", there is an associated codelist with the values "cm", "m", "IN". In such a case, you will have to develop such a codelist by subsetting the VSRESU codelist.
  Good examples (but also having "overkill") can be found in the define.xml 2.1 specification on the CDISC website.
  Generating valuelists will often make sense for laboratory data, especially for LBORRES and LBORRESU and for LBSTRESN and LBSTRESU, as one will have a multitude of not only units, but also maybe of ordinal values. Explaining which of these depending on the test code (LBTESTCD) will often make life easier for the reviewer.
  For generation of valuelists, see the tutorial "Working with the WhereClause in define.xml 2.0/2.1".

Once you have your mappings and definitions of datasets and in a decent state, you can start generating a "cleaned" define.xml. ***Please remark that you will not be able to use this "cleaned" define.xml for further mapping development, so best keep it apart***.

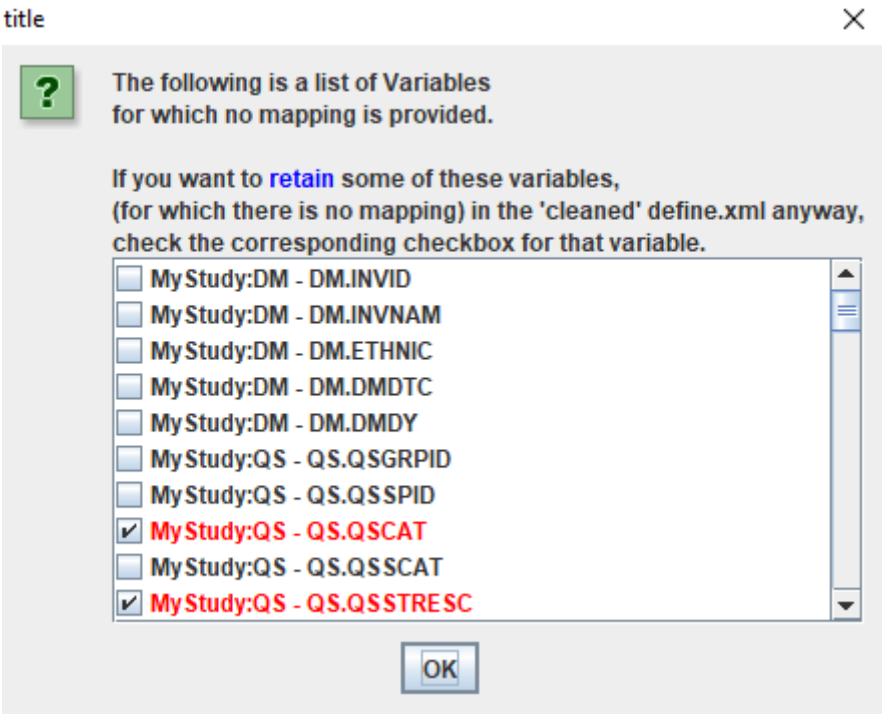
In order to start generating a "cleaned" define.xml, use the menu "File - Save cleaned define.xml":



The following dialog is then displayed:

Often, but now always, you will want to "remove SDTM variables that do not have a mapping provided". This is especially the case for "permissible" variables. For example, it doesn't make sense to keep VSSCAT when you never used it.

When clicking the checkbox "Remove SDTM variables that do not have a mapping provided", a new window pops up, allowing you to make exceptions:



In the provided list, variables shown in red are "expected" or "required" variables for which there is no mapping (yet), so you probably do not want to remove them from your cleaned define.xml, and their checkbox is automatically checked. Not retaining them (i.e. unchecking the checkbox) may lead to having violations against the SDTM or SEND standard.

You can now add other variables to be retained, even when there is no mapping (yet) for them. This will especially be important for study design domains[3], which in SDTM-ETL are developed in a somewhat different way. See further on for more details.

In the case of a regulatory submission, and as long this is required by them, you will probably also leave the checkbox "Move non-standard SDTM/SEND Variables to SUPP--". Whether non-standard variables should be kept in the original dataset (and marked as such in the define.xml) or "banned" to a supplemental qualifier dataset (like SUPPDM, SUPPVS, SUPPLB) is and remains a heavily debated issue within CDISC. Essentially, the requirement is a consequence of the primitiveness of the review tools at some of the regulatory authorities.

Also, as long as SAS Transport 5 is mandated by the regulatory authorities, and you have a CO (Comments) dataset in your set (or have it automatically generated) leave the checkbox "Automatically generate multiple COVAL variables when COVAL length > 200 characters" checked. In such a case, the system will look into what length you have set to the COVAL variable in your CO dataset definitions. For example, if you have set the length to 500, it will generate metadata in the define.xml for COVAL, with a length of 200, a COVAL1 with a length of 200, and a COVAL2 with a length of 100.

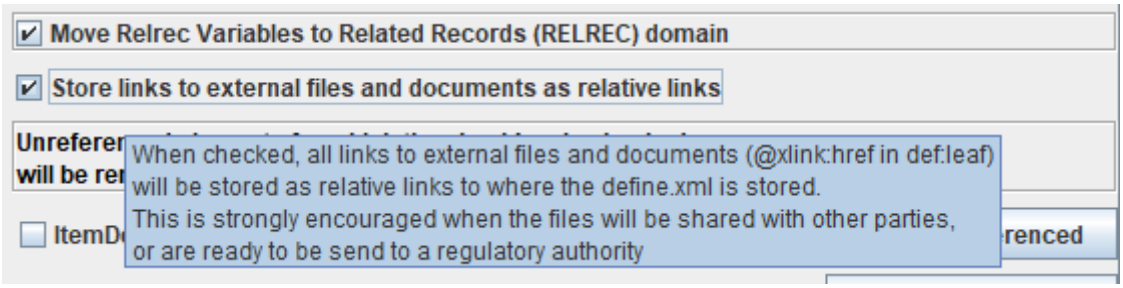
Usually, also the checkbox "Move Relrec Variables to Related Records (RELREC) Domain" will need to remain checked. Unfortunately, CDISC choose for the most primitive way to assign relations between variables from different domains, but this is once again a

---

[3] Essentially, study design domains should not be in SDTM. The better way would be that regulatory authorities accept an ODM file, as the latter already contains the fully study design. That study design datasets must be submitted again testifies of the "everything is a table" thinking within some parts of CDISC and at the regulatory authorithies.

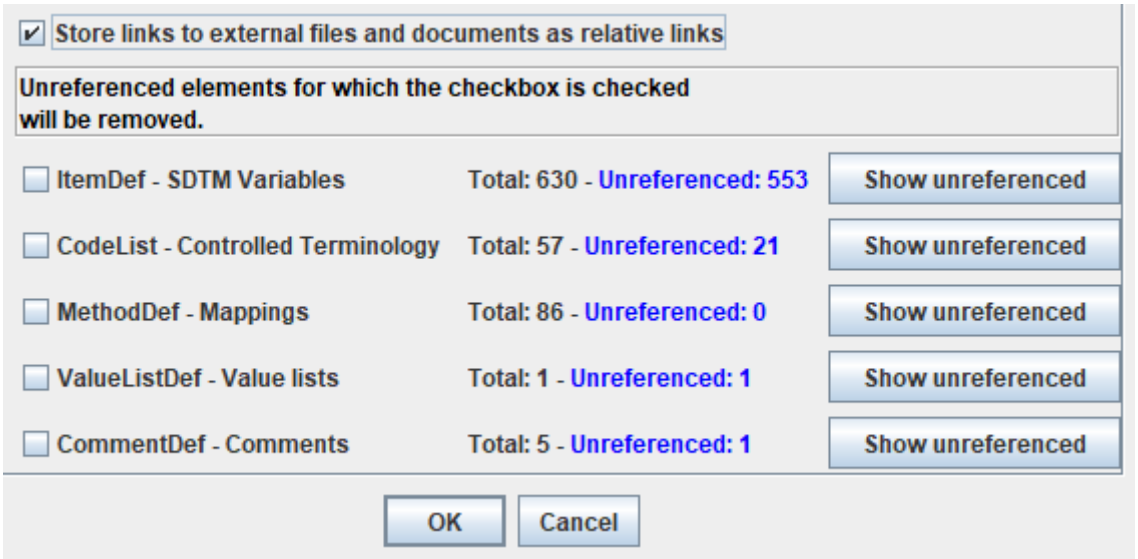
consequence of "everything is a table" thinking. When using e.g. JSON, this could be done in a much more elegant and straightforward way.

Also the next checkbox, "Store links to external files and documents as relative links" will usually remain checked.

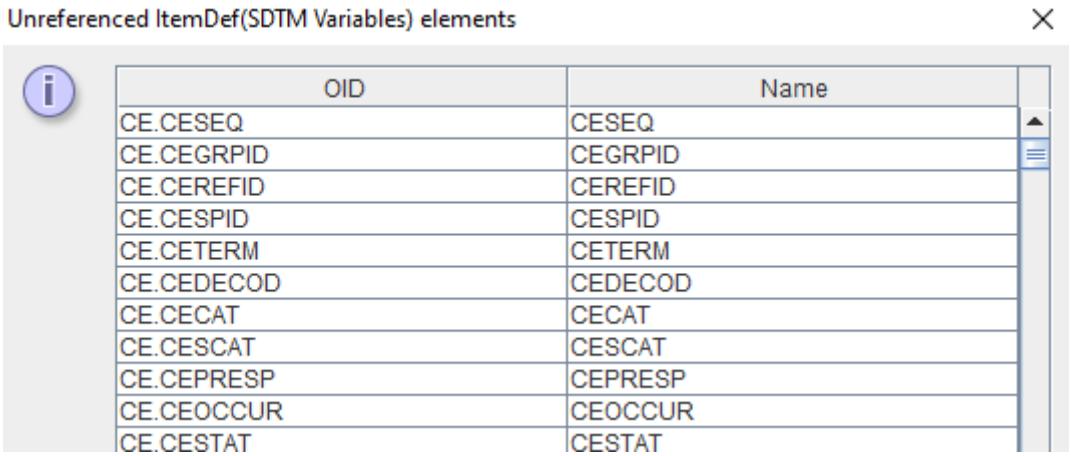


Last but not least, there is the checkbox "Remove Mapping Scripts from the define.xml", In earlier versions of the software, this was automatically done, but some of our users asked to allow the feature that the mapping scripts themselves are retained in the "MethodDef" elements in the define.xml.

The next section is a very important one in order to tailor the cleaned define.xml to your needs:

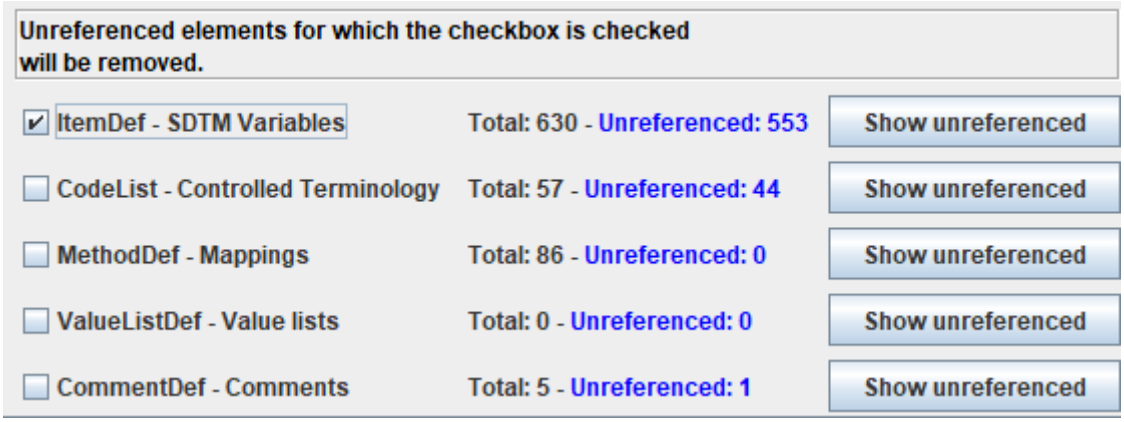


Very probably, you will not have mappings for each and every SDTM or SEND domain of the Implementation Guide. This also means that there is a large number of "ItemDef"s (variable definitions), especially from the template, for which there is no mapping, and you do not want to appear in your "cleaned" define.xml. For example, in case you do not have a QS dataset, you will not want to retain QSSEQ, QSGRPID, QSTESTCD, etc. as variable definitions in the define.xml.
By clicking the "Show unreferenced" button, you can see which variable definitions are not used. For example:

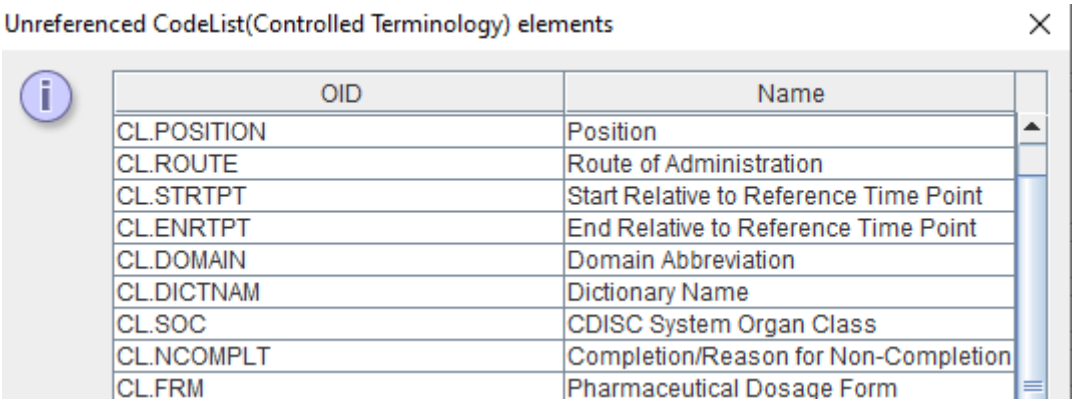


as we do not have a CE dataset for our study at all.

Checking the checkbox "ItemDef - SDTM Variables" will then lead to:

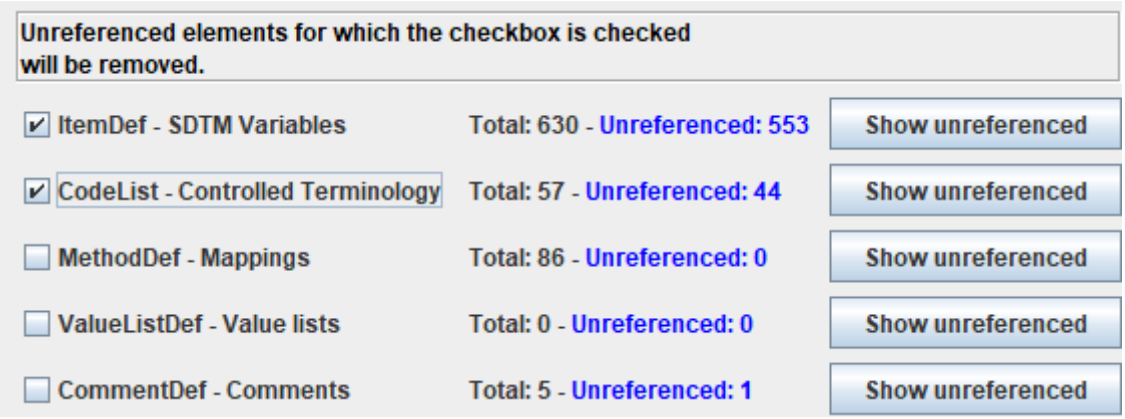


One sees that the number of "unreferenced codelists" increases, as we may have some codelists that are not used, even though there still is a variable for them.
Clicking "Show unreferenced" on "Codelists - Controlled Terminology" then displays:

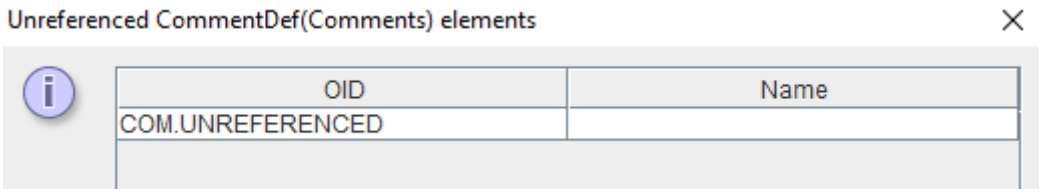


We find "POSITION" as an unreferenced codelist, though we do have a VS dataset defined, but we did not use VSPOS (a "permissible" variable) in our case. So the "POSITION" codelist needs not necessarily be retained.

If we then check the checkbox "CodeList - Controlled Terminology", all codelists that are not referenced, will be removed, and we get:

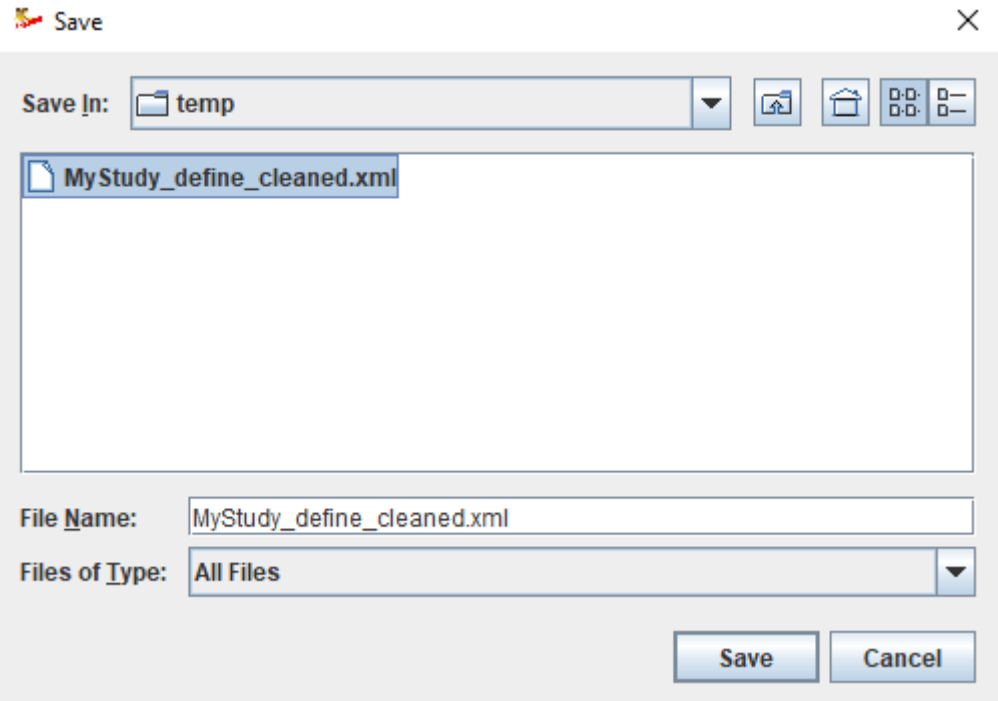


We then see that we have 86 mappings being defined, and none of them is "orphaned" or "dangling", i.e. each "MethodDef" in our define.xml is referenced. So, in this case, we can leave the checkbox "MethodDef - Mappings" unchecked. Essentially, it doesn't matter whether we check it or leave it unchecked in this case.
The same applies to "ValueListDef - Value lists": there are none defined that are not referenced.
We seem to have one "CommentDef" though that is never referenced, i.e. (by clicking "Show unreferenced"):



So, we check the checkbox "CommentDef - Comments" to have that unreferenced CommentDef removed from our cleaned define.xml.

Clicking the "OK" button then leads to a file selector, allowing us to define where the define.xml file needs to be written to. For example:



After a few seconds the cleaned define.xml can be found in the directory under the file name provided.

Please notice again that a "cleaned" define.xml cannot be used for the development of further mappings. It is however possible to further finetune it e.g. for submission purposes.

Further processing

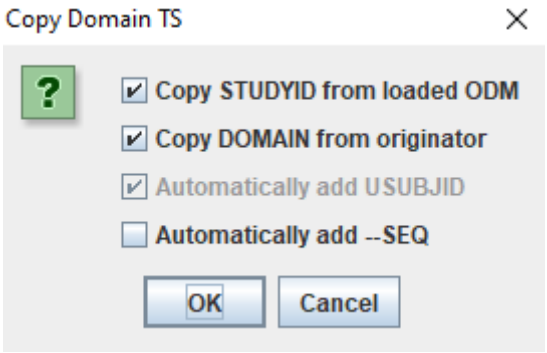
Sometimes, especially near submission time, one will further want to tailor the define.xml file. This can partially be done using the SDTM-ETL software, but in many cases it will be easier to do this using an define.xml editor, such as our "Define-XML Designer" which comes as a separate product. This makes sense as the further tailoring will often be done by another person, and the cost of a license of the "Define-XML Designer" is considerably lower than that of a license for SDTM-ETL.

Setting properties for Trial Design datasets

Trial Design are usually generated in a somewhat different way in SDTM-ETL, as they cannot always be retrieved from the ODM file.
For details, see the tutorial "Creating and editing trial design datasets".

In order to have the definition of such trial design datasets in the cleaned define.xml, one should follow the following procedure, here demonstrated for the TS dataset.

First drag-and-drop the "TS" row from the template to the bottom. This leads to:



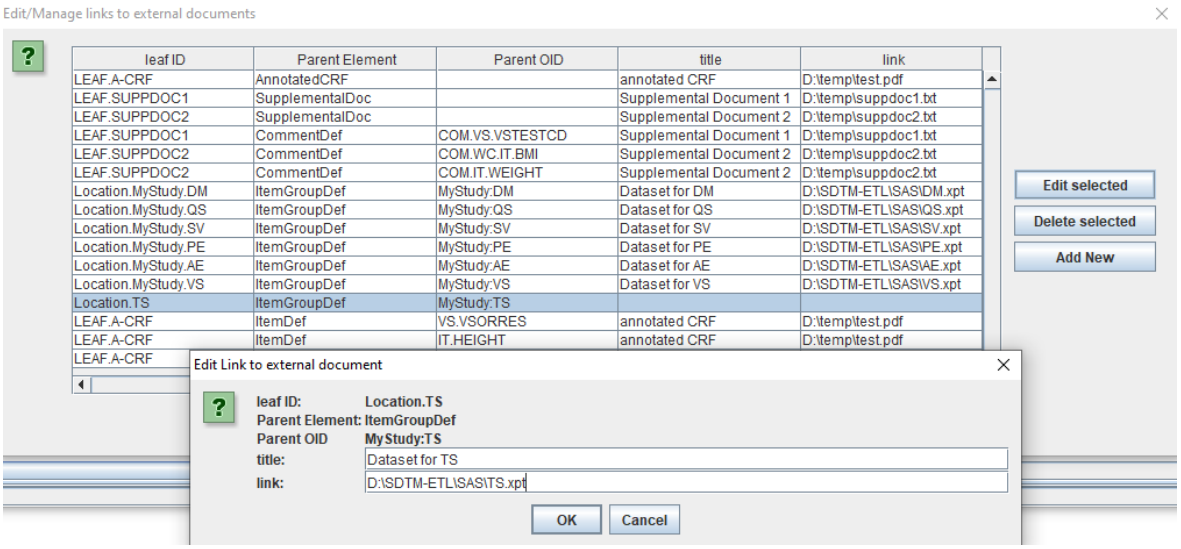
Remark that the checkbox "Automatically add --SEQ" remains unchecked, as the "sequence number" has a completely different meaning in TS than it has in non-trial-design datasets.

After clicking "OK" we get:

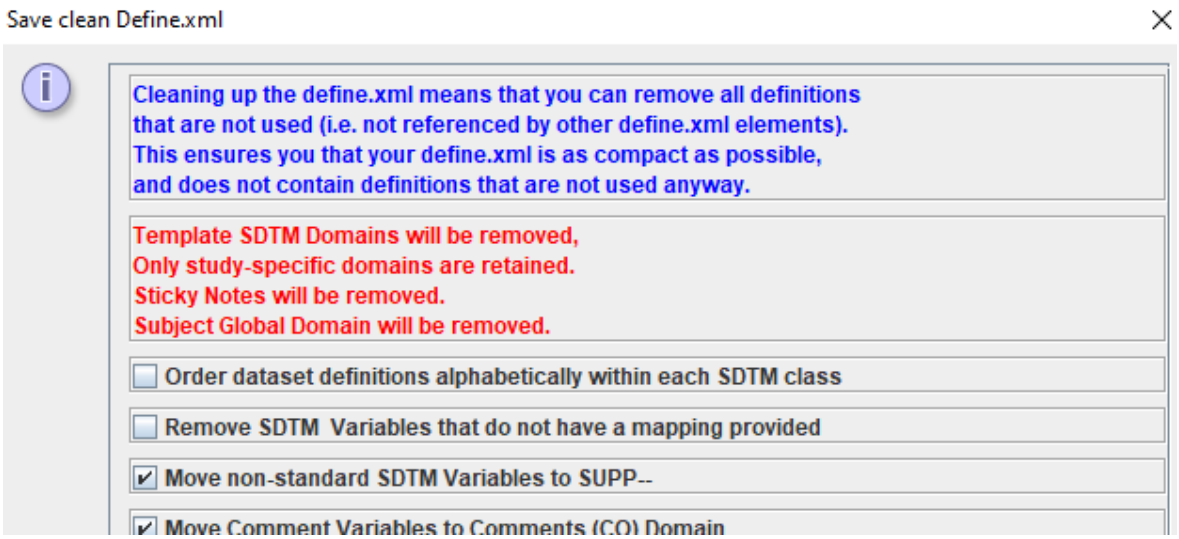
MyStudy:AE	STUDYID	DOMAIN	USUBJID	AE.AESEQ	AE.AEGRPID	AE.AEREFID
MyStudy:VS	STUDYID	DOMAIN	USUBJID	VS.VSSEQ	VS.VSGRPID	VS.VSSPID
MyStudy:TS	STUDYID	DOMAIN	TS.TSSEQ	TS.TSGRPID	TS.TSPARMCD	TS.TSPARM

Now set suitable properties, especially for the "maximum length", for each variable, using "Edit - SDTM variable. For example, the default "maximum length" for TSPARM in the template is 80, but one may want to set it to a lower value, depending on what was used when generating the TS.xpt dataset (in the case of use of SAS-XPT).
Do NOT add mappings however.

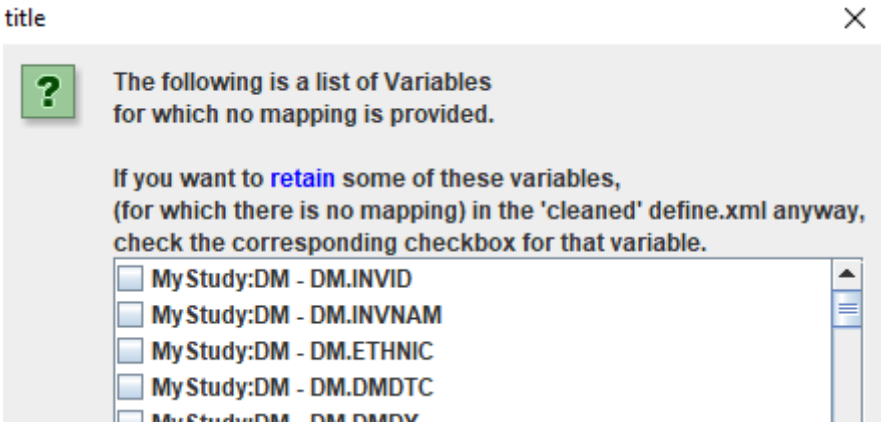
Also, add a link to where the TS.xpt file resides (or will reside) using the menu "Edit - Links to external documents", e.g.:



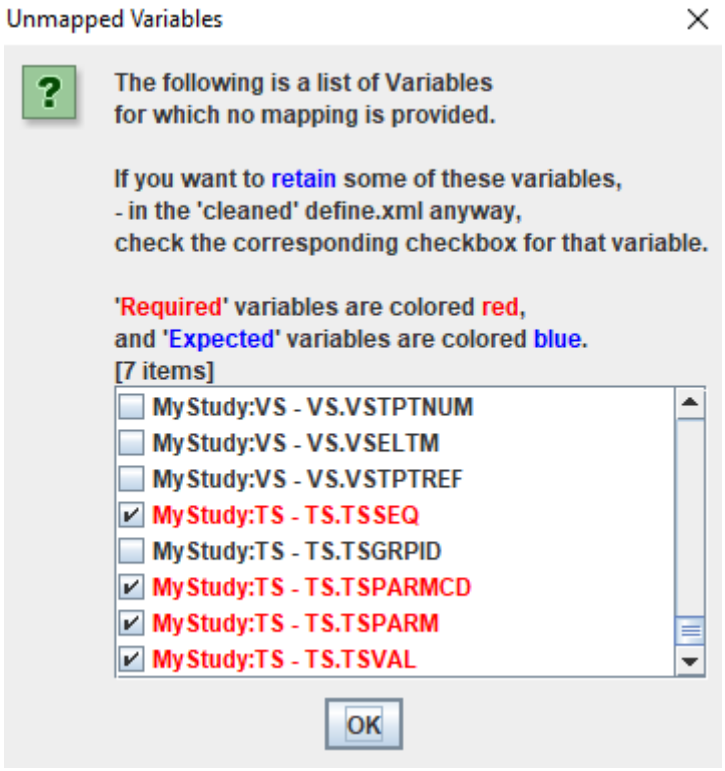
When now using the menu "File - Save cleaned define.xml", we come to:



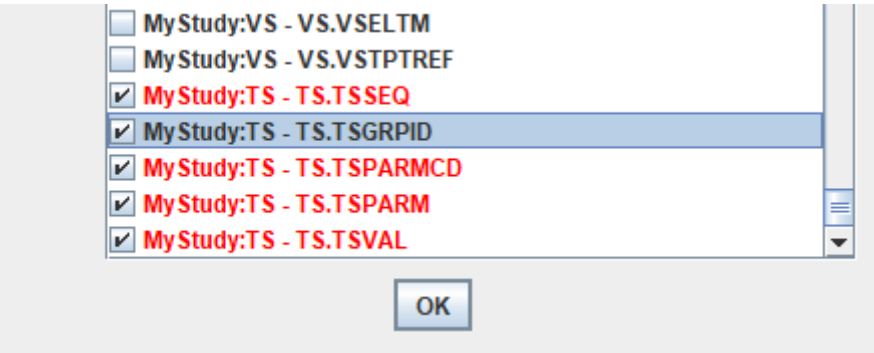
Now check the checkbox "Remove SDTM variables that do not have a mapping provided", which will open a new dialog, allowing us to make exceptions:



As "TS" was the last domain added, scroll down to the bottom:



We see that the variables TSSEQ, TSPARMCD, TSPARM and TSVAL are already marked as an exception for removal, as these are "required" variables. If we have also used TSGRPID when generating the TS.xpt dataset, also check "MyStudy:TS - TS.TSGRPID":



Remark that as of SDTMIG v.3.2, there are some additional variables such as TSVALNF, TSVALCD, TSVCDREF, TSVCDVER. The above screenshot is however still from a SDTMIG v.3.2 study.

The further procedure is as described as above, and in the final "cleaned" define.xml file we will find:

```
<ItemGroupDef Domain="TS" IsReferenceData="Yes" Name="TS" OID="MyStudy:TS"
              Purpose="Tabulation" Repeating="No" SASDatasetName="TS"
              def:ArchiveLocationID="Location.TS"
              def:Class="Trial Design"
              def:Structure="One record per TS.TSPARMCD">
   <Description>
      <TranslatedText xml:lang="en">Trial Summary</TranslatedText>
   </Description>
   <ItemRef ItemOID="STUDYID"
            Mandatory="Yes"
            MethodOID="IMP.MyStudy:TS.34.STUDYID"
            OrderNumber="1"
            Role="Identifier"/>
   <ItemRef ItemOID="DOMAIN"
            Mandatory="Yes"
            MethodOID="IMP.MyStudy:TS.34.DOMAIN"
            OrderNumber="2"
            Role="Identifier"/>
   <ItemRef ItemOID="TS.TSSEQ"
            Mandatory="Yes"
            OrderNumber="3"
            Role="Identifier"/>
   <ItemRef ItemOID="TS.TSGRPID"
            Mandatory="No"
            OrderNumber="4"
            Role="Identifier"/>
   <ItemRef ItemOID="TS.TSPARMCD"
            Mandatory="Yes"
            OrderNumber="5"
            Role="Identifier"/>
   <ItemRef ItemOID="TS.TSPARM"
            Mandatory="Yes"
            OrderNumber="6"
            Role="Synonym Qualifier"/>
   <ItemRef ItemOID="TS.TSVAL"
            Mandatory="Yes"
            OrderNumber="7"
            Role="Result Qualifier"/>
   <def:leaf xmlns:xlink="http://www.w3.org/1999/xlink"
             ID="Location.TS"
             xlink:href="..\SDTM-ETL\SAS\TS.xpt">
      <def:title>Dataset for TS</def:title>
   </def:leaf>
</ItemGroupDef>
```

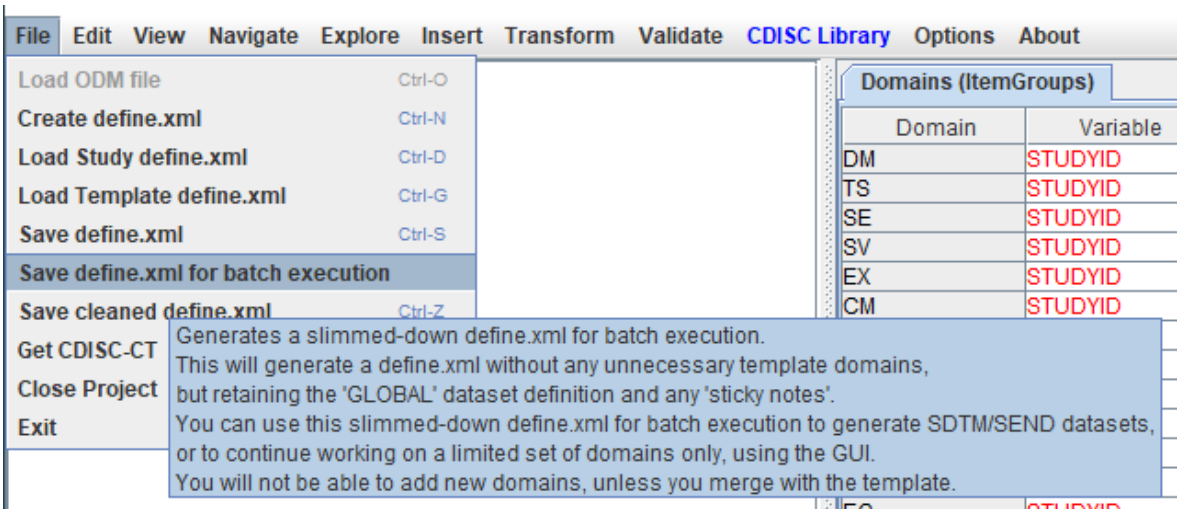
and e.g.

```
<ItemDef DataType="text"
         Length="8"
         Name="TSPARMCD"
         OID="TS.TSPARMCD"
         SASFieldName="TSPARMCD">
   <Description>
      <TranslatedText xml:lang="en">Trial Summary Parameter Short Name</TranslatedText>
   </Description>
   <CodeListRef CodeListOID="CL.TSPARM"/>
</ItemDef>
<ItemDef DataType="text"
         Length="40"
         Name="TSPARM"
         OID="TS.TSPARM"
         SASFieldName="TSPARM">
   <Description>
      <TranslatedText xml:lang="en">Trial Summary Parameter</TranslatedText>
   </Description>
</ItemDef>
<ItemDef DataType="text"
         Length="30"
         Name="TSVAL"
         OID="TS.TSVAL"
         SASFieldName="TSVAL">
   <Description>
      <TranslatedText xml:lang="en">Parameter Value</TranslatedText>
   </Description>
</ItemDef>
```

Remark again that there is a separate feature for generating the Trial Design datasets themselves, using the menu "Edit - Trial Design Dataset".

Differences with feature "Save define.xml for Batch Execution"

In the "File" menu, one also find the entry "Save define.xml for Batch Execution":



This is however something completely different …

"Save define.xml for batch execution" will save a define.xml that is slimmed down, i.e. the

template domains ("ItemGroups" in the define.xml") will be removed, allowing faster execution when running SDTM-ETL in "batch" execution mode. In principal, one could load such a "slimmed down" also in SDTM-ETL in the GUI, but as the template domains are missing, one will not be able to develop mappings than for the dataset definitions which one already had.

For further details, see the tutorial "SDTM-ETL Light and running in batch execution mode".