SDTM-ETL 4.1 User Manual and Tutorial

Author: Jozef Aerts, XML4Pharma

Last update: 2022-07-27

Creating and editing Trial Design datasets

When starting from an ODM file with metadata including SDM-XML (Study Design Model in XML), it is pretty straightforward to generate trial design datasets from the SDM-XML using the SDTM-ETL software (see separate manual).

However, not everyone is using SDM-XML yet, and most EDC systems do even not export SDM-XML. So, how can one generate the trial design datasets when the information is not, or only limited, in the ODM file with the study metadata?

We have recognized this problem, and added a new module to the SDTM-ETL software: a "trial design dataset editor". This editor does not only allow to generate trial design datasets from scratch, but also allows to edit existing trial design in Dataset-XML format, and then save them in either the modern Dataset-XML or Dataset-JSON format, or in the outdated SAS-XPT format, which is unfortunately still required by the FDA[1].

The editor can also be run in standalone method (so without using the SDTM-ETL software). This is explained later in this document.

In this tutorial, we will demonstrate the use of the "trial design dataset editor" using the TE (Trial Elements) TA (Trial Arms) as an example. Some additional information will be provided for the case of TS (Trial Summary), as this is a somewhat special case.

Starting the editor from within SDTM-ETL

After having loaded an SDTM template or existing define.xml with SDTM-ETL mappings, create a study-specific instance of the desired domain, in this case the TE and TA domains. Do so by dragging the "TE" and the "TA" row from the template rows to the bottom of the table. This e.g. results in:

SR	STUDYID	DOMAIN	USUBJID	SR.SRSEQ	SR.SRGRPID	SR.SRREFID	SR.S
TA	STUDYID	DOMAIN	TA.ARMCD	TA.ARM	TA.TAETORD	TA.ETCD	TA.EL
TD	STUDYID	DOMAIN	TD.TDORDER	TD.TDANCVAR	TD.TDSTOFF	TD.TDTGTPAI	TD.TI
TE	STUDYID	DOMAIN	TE.ETCD	TE.ELEMENT	TE.TESTRL	TE.TEENRL	TE.TE
TI	STUDYID	DOMAIN	IE.IETESTCD	IE.IETEST	IE.IECAT	IE.IESCAT	IE.TIF
TM	STUDYID	DOMAIN	TM.MIDSTYPE	TM.TMDEF	TM.TMRPT		
TS	STUDYID	DOMAIN	TS.TSSEQ	TS.TSGRPID	TS.TSPARMCD	TS.TSPARM	TS.TS
TV	STUDYID	DOMAIN	TV.VISITNUM	TV.VISIT	TV.VISITDY	TV.ARMCD	TV.AF
OI	STUDYID	DOMAIN	OI.NHOID	OI.OISEQ	OI.OIPARMCD	OI.OIPARM	OI.OI
RELREC	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	RELTYPE	RELI
RELSPEC	STUDYID	USUBJID	REFID	SPEC	PARENT	LEVEL	
RELSUB	STUDYID	USUBJID	POOLID	RSUBJID	SREL		
SUPPQUAL	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	QNAM	QLAE
CES:TE	STUDYID	DOMAIN	TE.ETCD	TE.ELEMENT	TE.TESTRL	TE.TEENRL	TE.TE
CES:TA	STUDYID	DOMAIN	TA.ARMCD	TA.ARM	TA.TAETORD	TA.ETCD	TA.EL

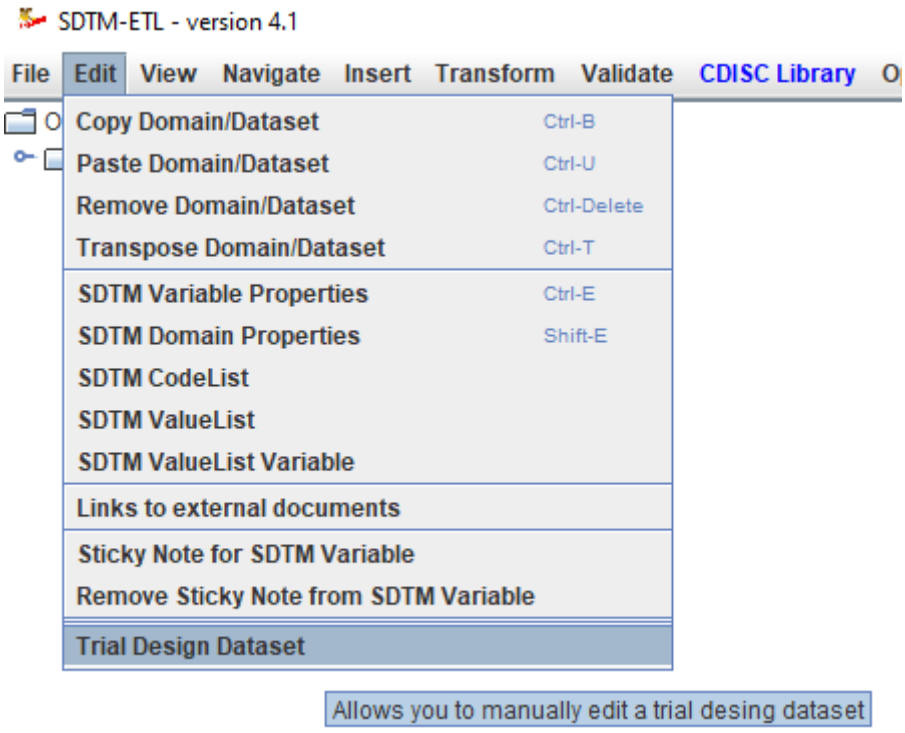
[1] SAS-XPT ("transport 5") is an over-30 year old format meant to exchange data between IBM mainframe and VAX computers (do you still have one at home). It is extremely inefficient with disk space and very software-unfriendly.

This ensures that the metadata for your study-spe cific TE and TA datasets will be included in the define.xml file. You will also be able to use the trial design dataset editor without having dragged-and-dropped the TA or TE row (or one of the other "trial design" rows), but in that case, the system will later require you to define the location of a define.xml file from which the metadata will then be taken.

It is important to realize that the generation of the trial design dataset is always driven by the metadata of a define.xml, be it the currently loaded define.xml (either using the TA or TE template row, or using the study-specific instance), or an external define.xml file. So it is important that you have a good set of metadata for your trial design dataset, such as having set **appropriate maximal lengths** (especially when outdated SAS-XPT needs to be generated), and having assigned codelists for specific variables.

If you have created a study-specific instance of your trial design domain (such as CES:TA/CES:TE in our example), you might first want to work on the metadata for its variables, setting maximal lengths, adapt data types when necessary, and especially assign codelists (e.g. for the EPOCH variable). Changing metadata for SDTM/SEND variables is explained in the other manuals.

In order to start generating a new trial design dataset or editing an existing one, now use the menu "Edit - Trial Design Dataset":



This will start up a separate window, which is the starting window for this module.

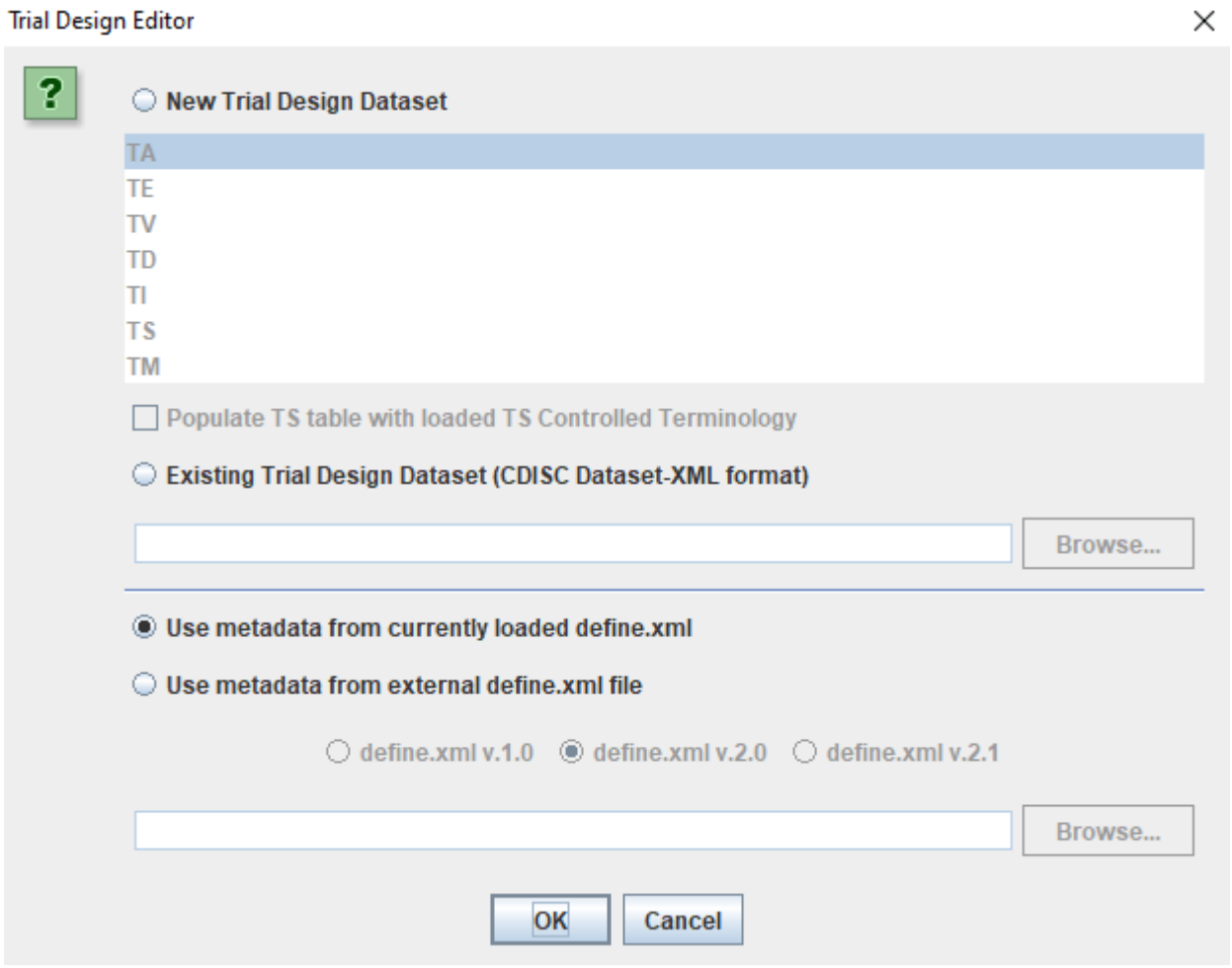
Starting the editor in standalone mode

There will be cases when you want to generate or edit a trial design dataset without starting SDTM-ETL. You can do so by double-click the icon for the file "TrialDesignEditor.bat".
First of all, the system will ask you whether you want to work with SDTM or SEND, and ask for the version of the IG, and for a version for the controlled terminology.

The "Trial Design Dataset Editor" start window will then appear.

Working with the Trial Design Dataset Editor

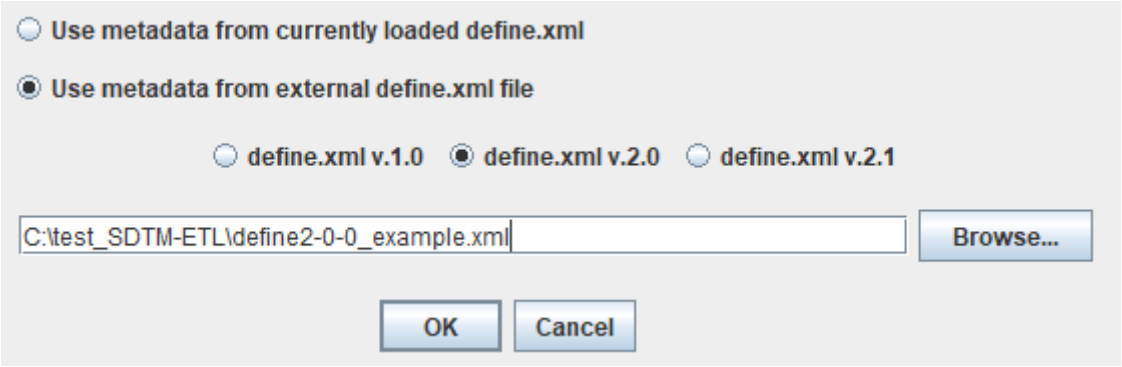
When the software has been started, either from within SDTM-ETL, or in standalone mode, the following start window is displayed:



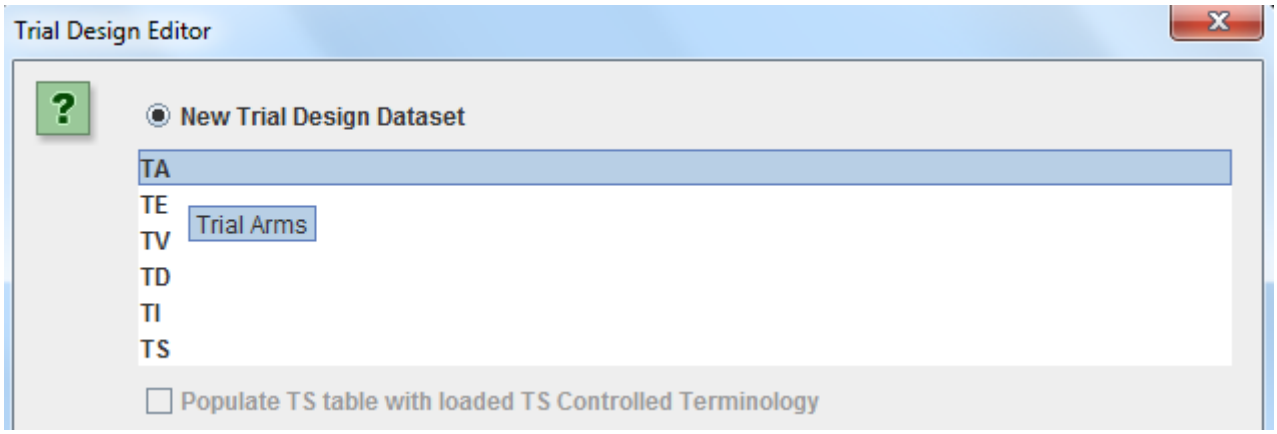
In case the software was started from within SDTM-ETL, the radiobutton "Use metadata from currently loaded define.xml" is preselected. This means that the define.xml from the SDTM-ETL will be used (in its current state). One can however then also choose to choose another define.xml file containing the metadata for the trial design dataset(s).
In case the software was started in standalone, the radiobutton "Use metadata from external define.xml file" is preselected, and the radiobutton "Use metadata from currently loaded define.xml" is disabled. So, in the latter case, it is always necessary to provide a define.xml file containing the trial design metadata.

In this tutorial we will continue with the option "Use metadata from external define.xml file".

First select the correct version of define.xml that you are working with. This is important as otherwise the file will not be parsed. In our case, we use a define.xml v.2.0 file. Select it using the "Browse" button on the right lower corner of the window. For example:



Now you need to decide whether you want to create a new trial design dataset (from scratch) or that you want to work on an already existing trial design dataset in Dataset-XML format.
We will first work with the case that one wants to start a completely new trial design dataset.
In order to do so, select the radiobutton "New Trial Design Dataset" and select a trial design domain from the list. For example:



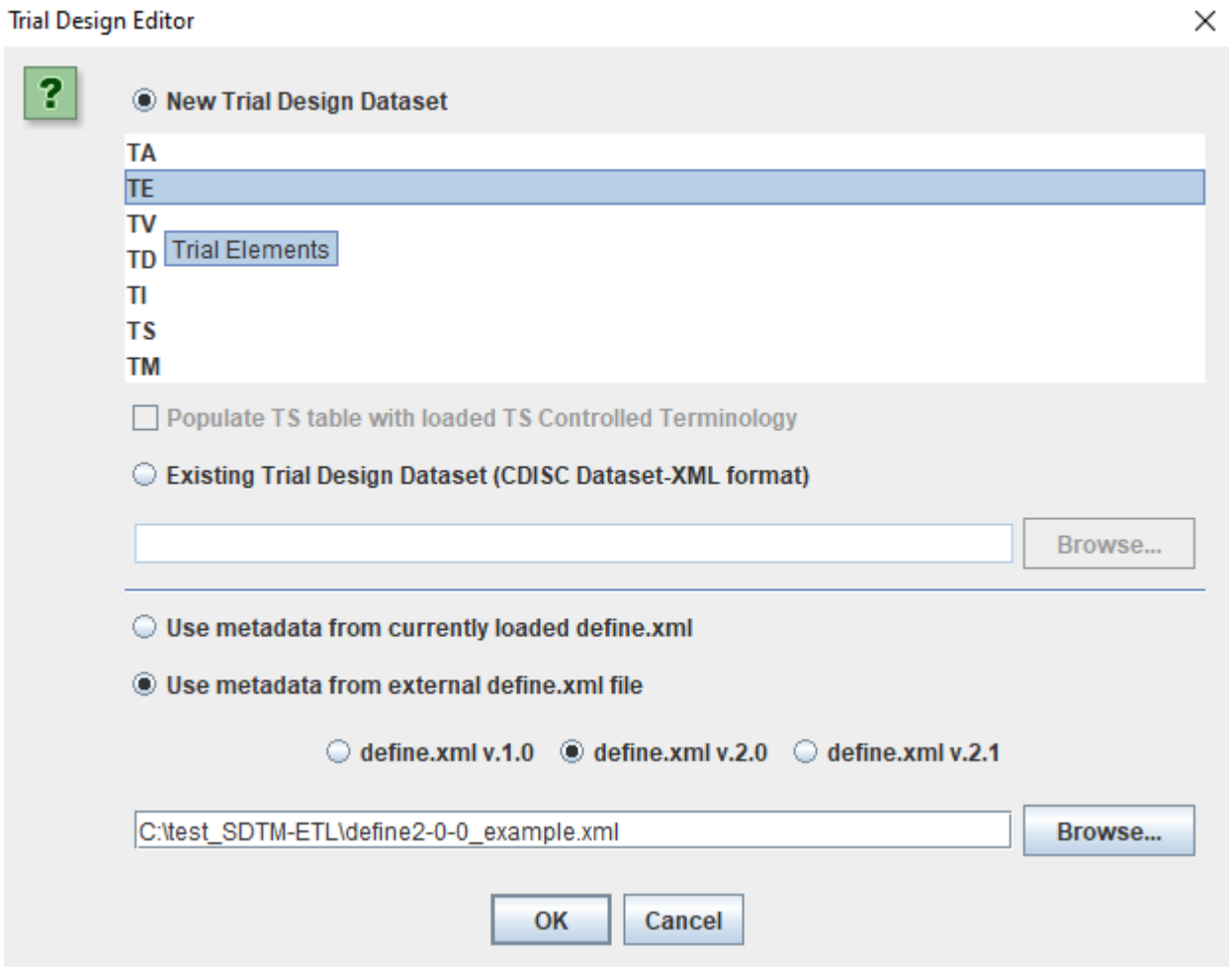
The tooltip on the selection shows the full name of the dataset.

The use of the checkbox "Populate TS table …" will be explained later. It is disabled for all choices except for TS.

Generating the TE dataset

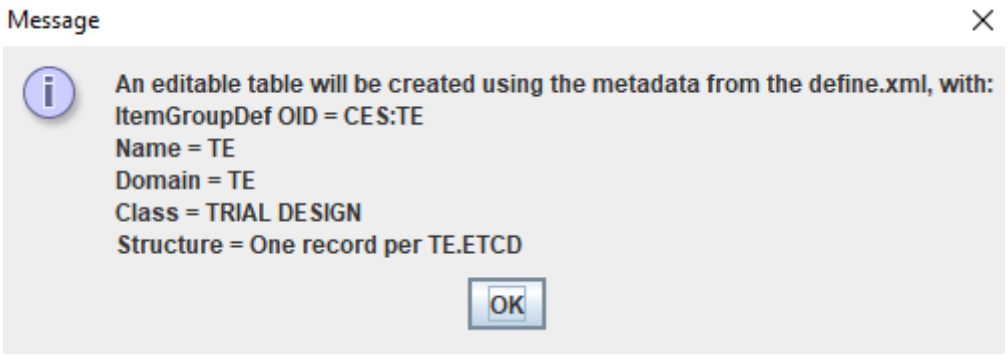
Let us first generate the TE (Trial Elements) dataset. Reason is that when generating the TA dataset, we can reuse variable values from the TE dataset, i.e. ETCD (Element code) and ELEMENT (Element Name). How this reuse is done is explained later on.

So we select "TE":

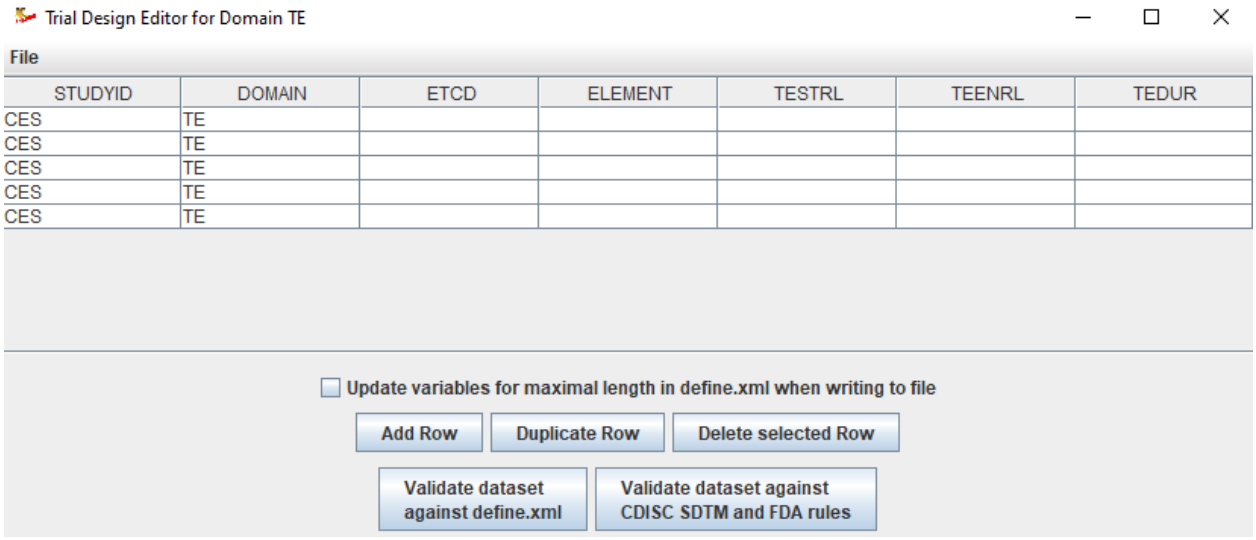


and click "OK".

First a dialog is shown containing the most important information:

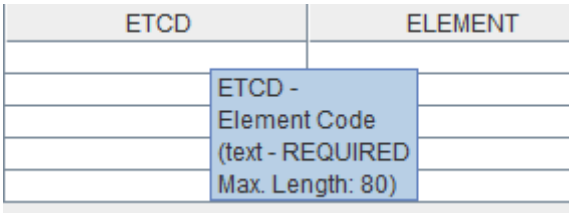


And after clicking "OK", the table to be edited appears:

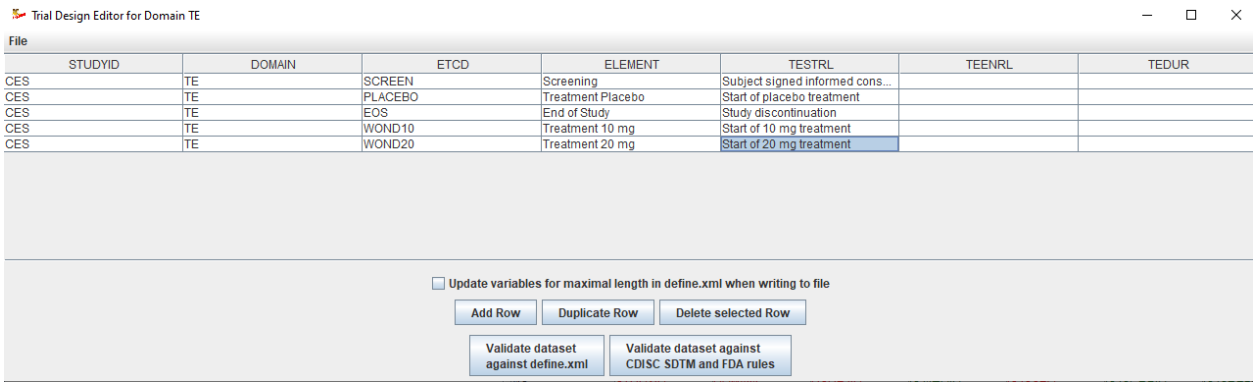


As one can see the fields for STUDYID and DOMAIN are pre-filled, as they always have the same value.

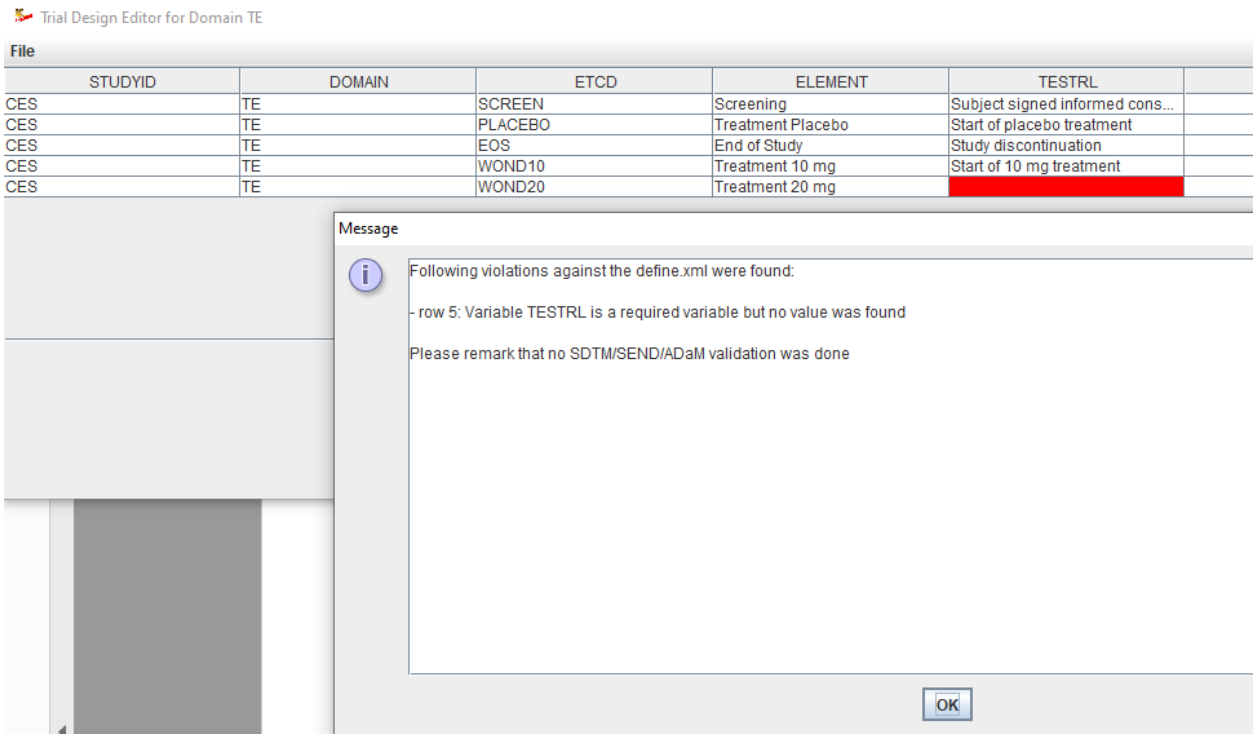
When hovering the mouse over a column header, the metadata information is displayed:



One can now start adding the information. This may e.g. lead to:



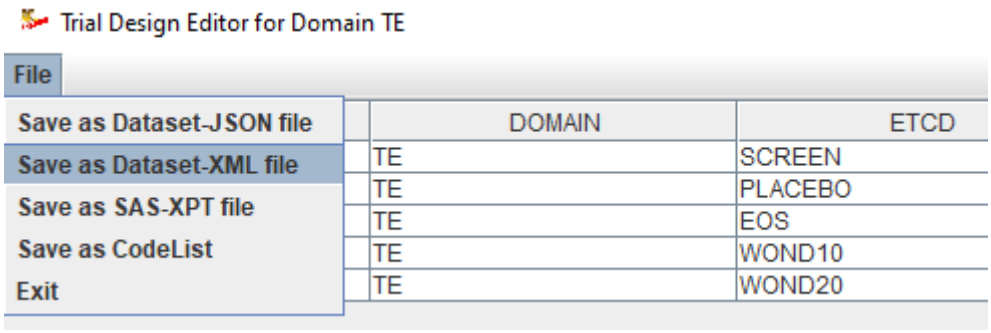
It is always a good idea to check the correctness of the structure using the button "Validate dataset against define.xml". For example, when a value for TESTRL is missing ("required" variable), one will get:



i.e. the cell is colored red and a message is displayed.
Also, when a column is of type "integer" (e.g. TAETORD in TA), and one types something in that is not an integer, the cell will get a red border, prompting to correct the value.

The use of the button "Validate dataset against CDISC SDTM and FDA rules" will be explained later.

Once everything is fine, one can save the TE dataset to file in either SAS Transport 5 (XPT), the new Dataset-JSON format, or Dataset-XML format.

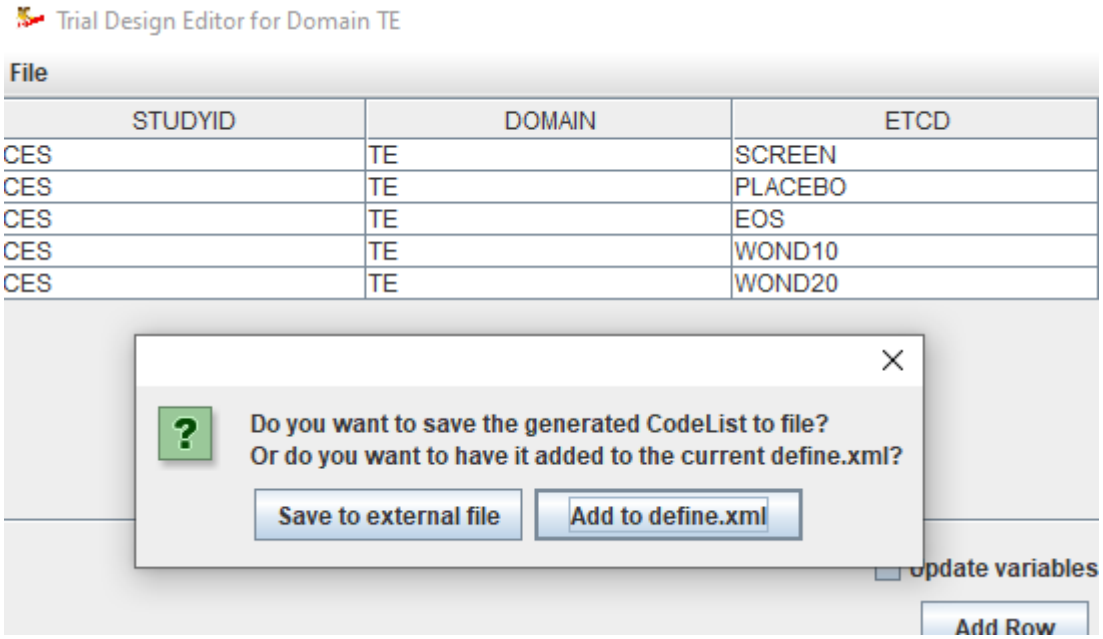


The latter can then later be used to reload the dataset contents and make changes or additions. So, it is always recommended to at least to save using Dataset-XML format.
SAS Transport 5 (XPT) is added as the regulatory authorities still mandate this outdated format. It is however expect that it will soon be replaced by Dataset-JSON format, so the latter is already supported.
When saving to SAS-XPT, it is recommended to check the checkbox "Update variables for maximal length in define.xml when writing to file". This will ensure that the generated XPT is as compact as possible[2].

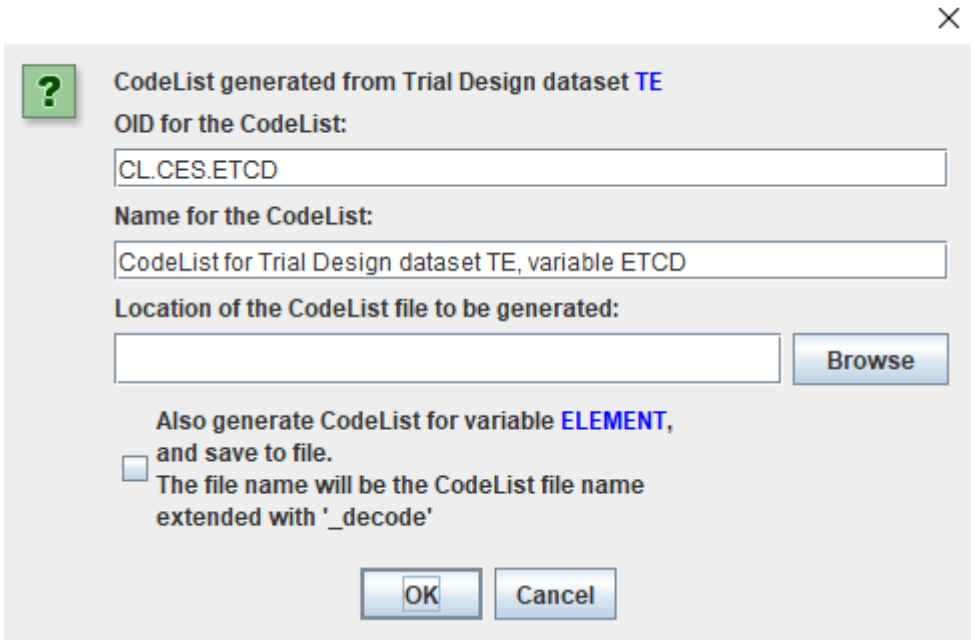
[2] SAS-XPT is well-known to be a very inefficient file format.

One also sees that there is a menu "Save as CodeList". Reason is that the values for ETCD and ELEMENT can then be saved as a codelist, so that it can be used in other datasets (such as TA, where ETCD and ELEMENT also appear). This will not only avoid "typing over", but also allows to guarantee data consistency.

When the menu "Save as CodeList" is used, the following dialog appears:



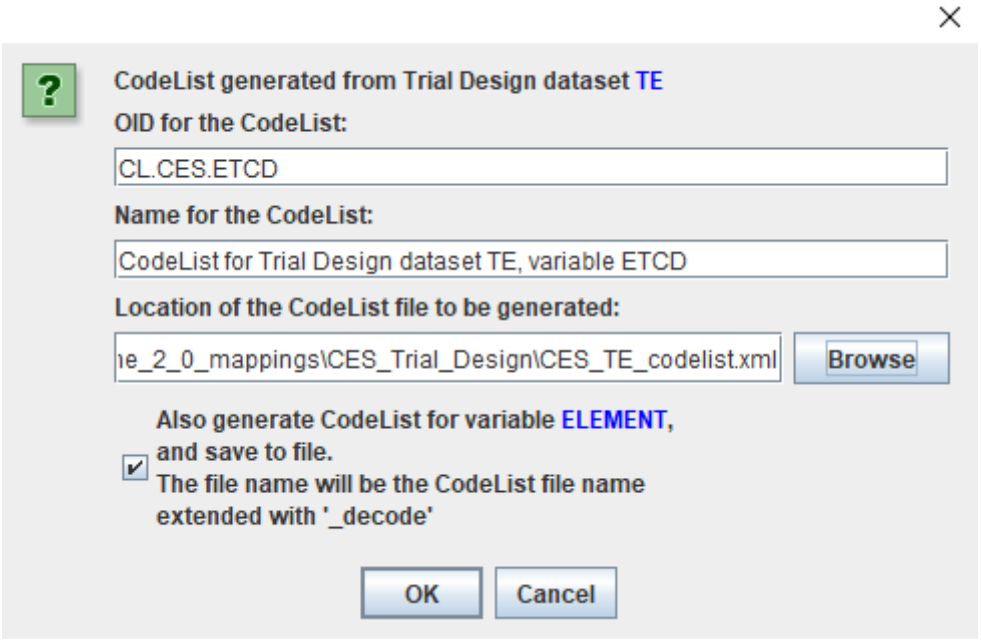
One can either select to save the codelist to an external file, or to immediately add it to the define.xml. The latter will only be possible when having started the Trial Design Dataset Editor from within SDTM-ETL(so not in "standalone" mode).
When "Save to external file" ic clicked, one is prompted to provide a location where the codelist will be stored as an ODM-XML file. One can then still later import it into SDTM-ETL using the menu "Insert - xxxx".



Values for the OID and Name of the CodeList are already suggested. One can of course still change these.

The additional checkbox "Also generate CodeList for the variable ELEMENT ..." is very interesting, as this allows to also generate a codelist with "decode" values, such as "Screening", "Placebo Treatment". This is useful as to the bad design of SDTM ("everything is a table") and variables such as ELEMENT in SDTM require their own separate codelist, this although the same information is also essentially present in the codelist for ETCD.

So we may have something like:



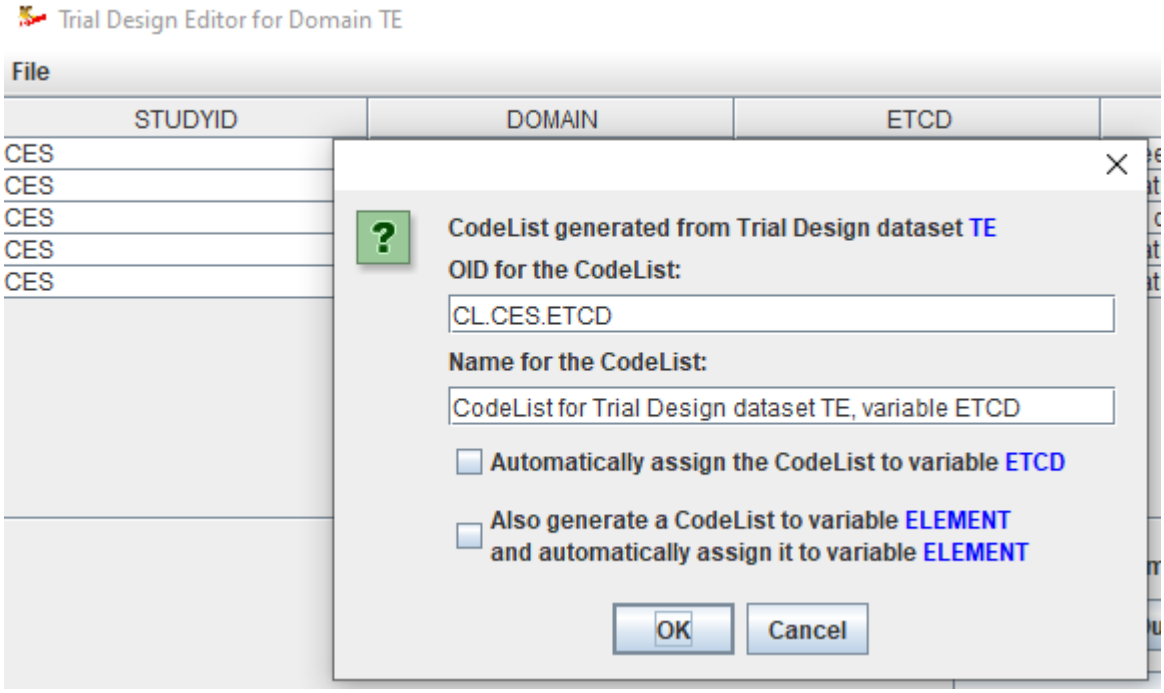
When clicking "OK", and all goes well, 2 messages will appear, one about having successfully generated the file "CES_TE_codelist.xml" and one about having successfully generated the file CES_TE_codelist_decode.xml". The content of the former looks like:

```xml
<CodeList xmlns="http://www.cdisc.org/ns/odm/v1.3" OID="CL.CES.ETCD"
    Name="CodeList for Trial Design dataset TE, variable ETCD" DataType="text">
    <CodeListItem CodedValue="SCREEN">
        <Decode>
            <TranslatedText>Screening</TranslatedText>
        </Decode>
    </CodeListItem>
    <CodeListItem CodedValue="PLACEBO">
        <Decode>
            <TranslatedText>Treatment Placebo</TranslatedText>
        </Decode>
    </CodeListItem>
    <CodeListItem CodedValue="EOS">
        <Decode>
            <TranslatedText>End of Study</TranslatedText>
        </Decode>
    </CodeListItem>
    <CodeListItem CodedValue="WOND10">
        <Decode>
            <TranslatedText>Treatment 10 mg</TranslatedText>
        </Decode>
    </CodeListItem>
    <CodeListItem CodedValue="WOND20">
        <Decode>
            <TranslatedText>Treatment 20 mg</TranslatedText>
        </Decode>
    </CodeListItem>
</CodeList>
```

and of the latter (CES_TE_codelist_decode.xml):

```xml
<CodeList xmlns="http://www.cdisc.org/ns/odm/v1.3" OID="CL.CES.ETCD.DECODE"
    Name="CodeList for Trial Design dataset TE, variable ETCD - decoded values" DataType="text">
    <EnumeratedItem CodedValue="Screening"/>
    <EnumeratedItem CodedValue="Treatment Placebo"/>
    <EnumeratedItem CodedValue="End of Study"/>
    <EnumeratedItem CodedValue="Treatment 10 mg"/>
    <EnumeratedItem CodedValue="Treatment 20 mg"/>
</CodeList>
```

When selecting "Add to define.xml", a similar dialog appears:



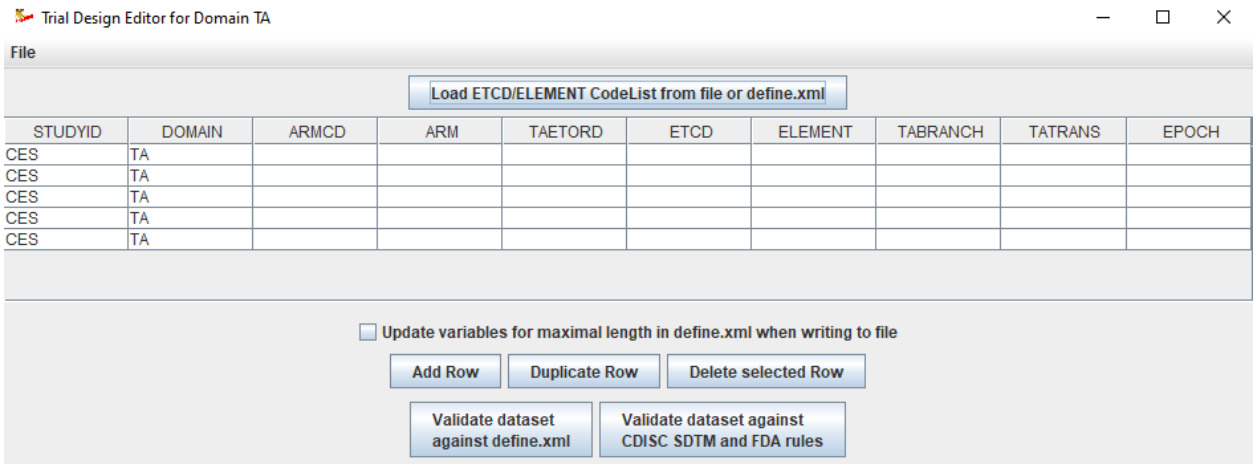
The difference being that the codelist will be added to the define.xml, and can be automatically be assigned to ETCD (first checkbox). Also here, one can have a corresponding codelist for ELEMENT having being generated, and assigned to the ELEMENT variable.
Also, it will be checked whether the OID already exists, and if so, one will be prompted to change it, or to overwrite the already existing codelist with the same OID, or to cancel the generation.

A list of the contents of the codelists generated using "File - Save as CodeList" is given below

Domain	CodeList variable	"Decode" CodeList variable
TE	ETCD	ELEMENT
TA	ARMCD	ARM
TI	IETESTCD	IETEST
TV	VISITNUM	VISIT
TD	TDORDER	TDANCVAR
TM	MIDSTYPE	TMDEF
TX (SEND)	SETCD	SET

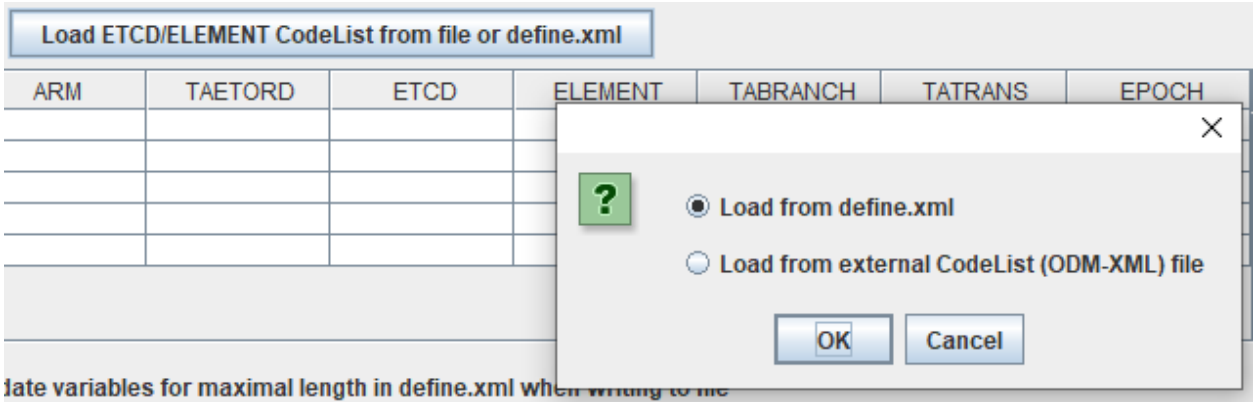
Generating the TA dataset

Similarly, one can now start generating the TA dataset:



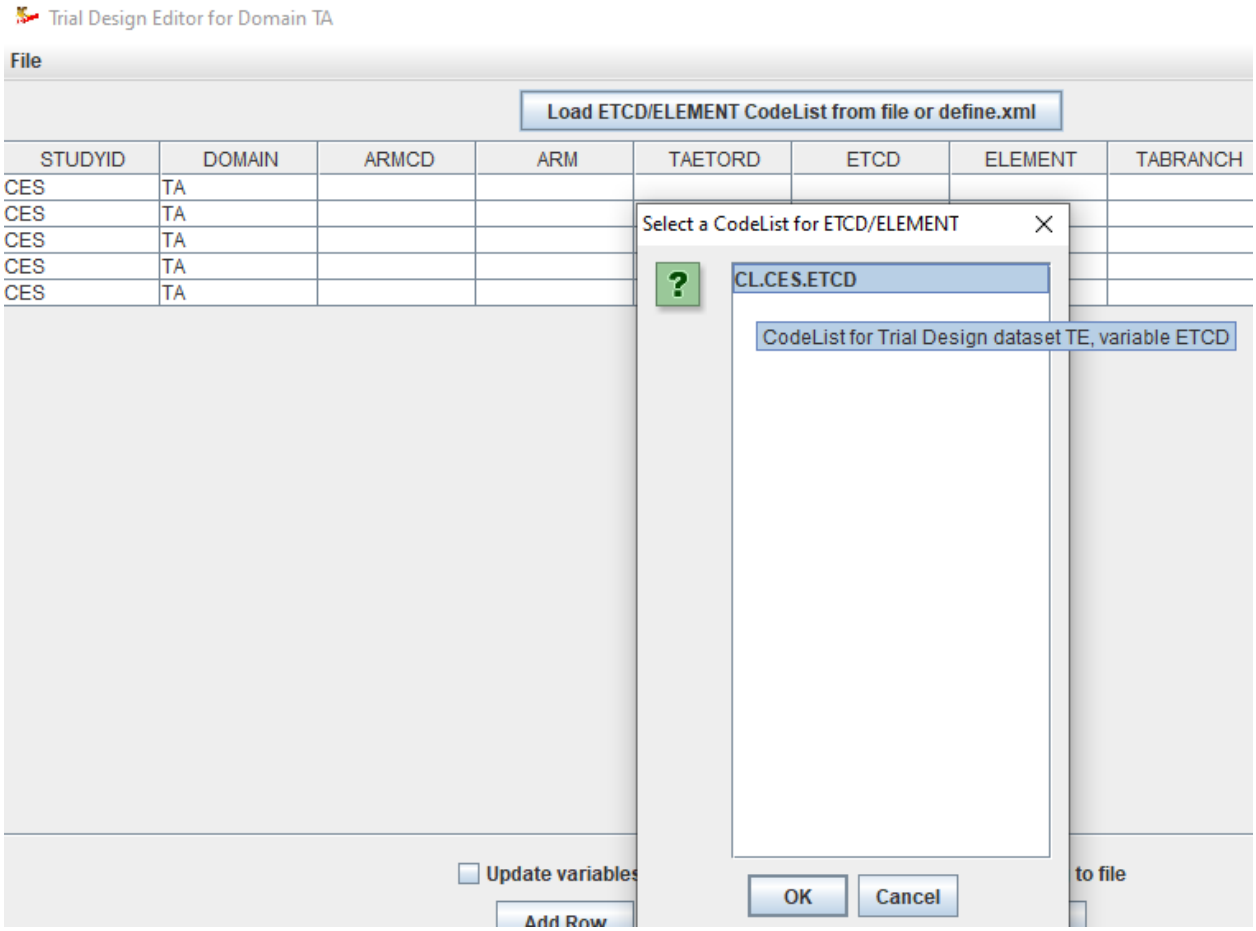
One immediately notices the additional button "Load ETCD/ELEMENT CodeList from file or define.xml". Reason is that the TA domain does not only represent the trial arms, but also how the trial elements are ordered within each arm[3].

When that button is clicked, the following dialog displays:

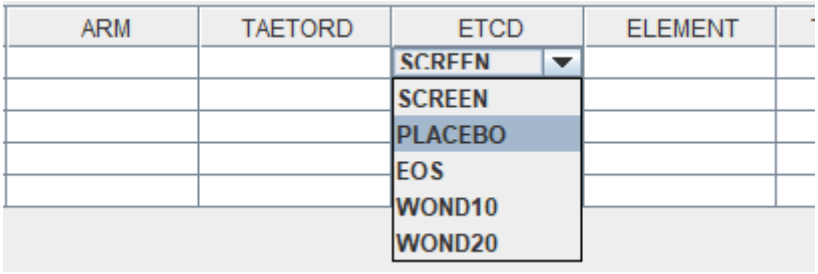


When "Load from define.xml" is selected, a list of all codelists in the define.xml is shown, from which one should pick the correct one. In case one loads the codelist(s) from an external file, one is prompted for its location, and a list of the contained codelists is shown, from which one should pick the right one. For example:

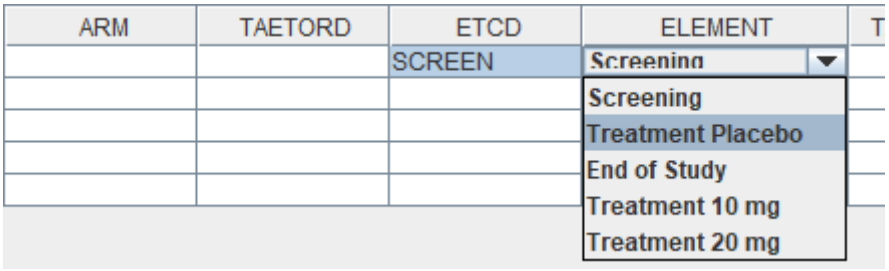
[3] This is essentially bad design: in a relational database, one would have a table for the trial arms, one for the trial elements, and a "relation" table allowing to define how the elements are ordered within each arm. However, SDTM has already for a ver long time thrown all first principles of good database design over board.



After clicking "OK", one will notice that the fields for "ETCD" and "ELEMENT" have now been replaced by dropdowns, like:

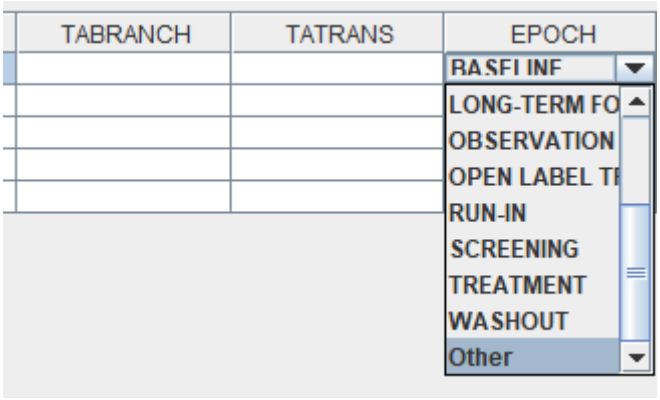


and:

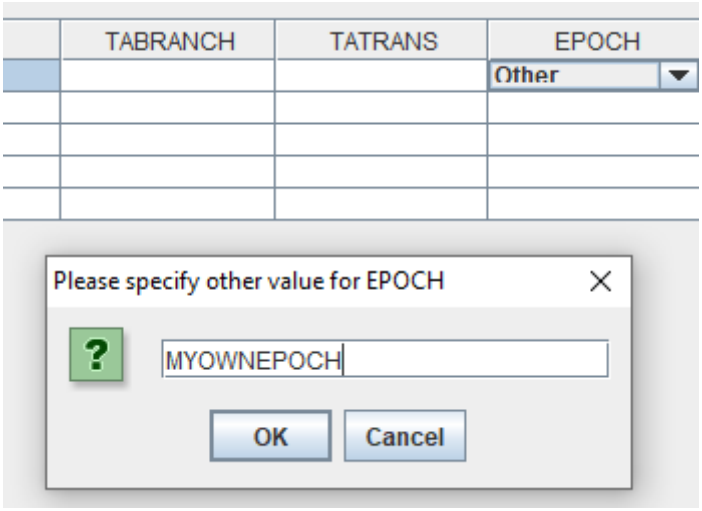


thus helping keeping consistency between the TE and TA datasets.

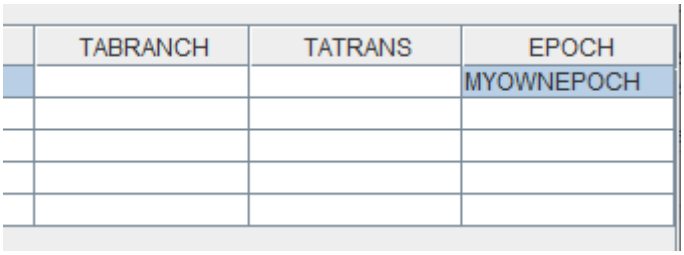
In TA, "EPOCH" is governed by a codelist, so this field also displays as a dropdown. The codelist is however extensible, so that one must be able to add new terms to it.
To do so, select the "EPOCH" field, and scroll down to the end of the list, and select "Other":



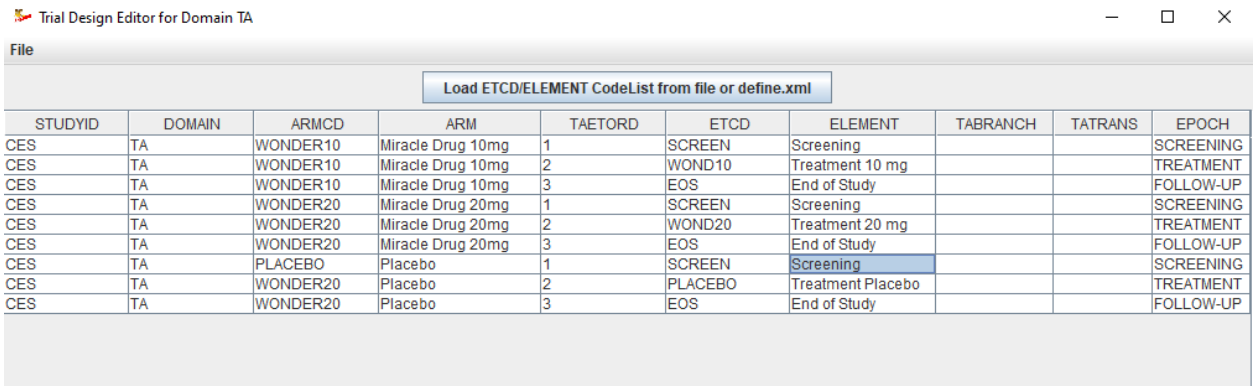
A text field dialog is then displayed, allowing to enter a new (extended) term for "EPOCH":



and when clicking "OK", the dropdown list is updated and the new term automatically selected:



A full "Trial Arms" design may then look like:



STUDYID	DOMAIN	ARMCD	ARM	TAETORD	ETCD	ELEMENT	TABRANCH	TATRANS	EPOCH
CES	TA	WONDER10	Miracle Drug 10mg	1	SCREEN	Screening			SCREENING
CES	TA	WONDER10	Miracle Drug 10mg	2	WOND10	Treatment 10 mg			TREATMENT
CES	TA	WONDER10	Miracle Drug 10mg	3	EOS	End of Study			FOLLOW-UP
CES	TA	WONDER20	Miracle Drug 20mg	1	SCREEN	Screening			SCREENING
CES	TA	WONDER20	Miracle Drug 20mg	2	WOND20	Treatment 20 mg			TREATMENT
CES	TA	WONDER20	Miracle Drug 20mg	3	EOS	End of Study			FOLLOW-UP
CES	TA	PLACEBO	Placebo	1	SCREEN	Screening			SCREENING
CES	TA	WONDER20	Placebo	2	PLACEBO	Treatment Placebo			TREATMENT
CES	TA	WONDER20	Placebo	3	EOS	End of Study			FOLLOW-UP

which can then be saved to file, either as Dataset-XML (for later changes), modern Dataset-JSON,

or outdated SAS Transport 5 (XPT) format.

Also here, "Save as CodeList" is very useful, and will save the distinct values of ARMCD and ARM (as the dataset is not only about "trial arms") either to ODM-XML files, or to the define.xml. This will then, for ARMCD, look like:

```
<CodeList xmlns="http://www.cdisc.org/ns/odm/v1.3" OID="CL.CES.ARMCD"
    Name="CodeList for Trial Design dataset TA, variable ARMCD" DataType="text">
    <CodeListItem CodedValue="WONDER10">
        <Decode>
            <TranslatedText>Miracle Drug 10mg</TranslatedText>
        </Decode>
    </CodeListItem>
    <CodeListItem CodedValue="WONDER20">
        <Decode>
            <TranslatedText>Miracle Drug 20mg</TranslatedText>
        </Decode>
    </CodeListItem>
    <CodeListItem CodedValue="PLACEBO">
        <Decode>
            <TranslatedText>Placebo</TranslatedText>
        </Decode>
    </CodeListItem>
```

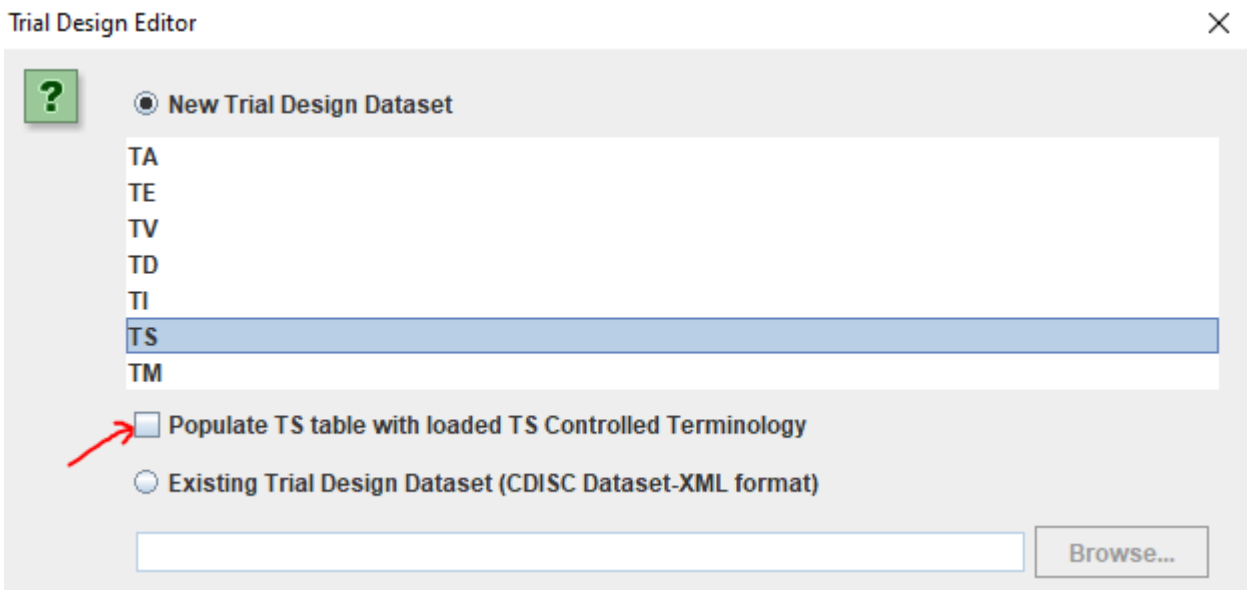
and for the "decode" ARM codelist:

```
<CodeList xmlns="http://www.cdisc.org/ns/odm/v1.3" OID="CL.CES.ARMCD.DECODE"
    Name="CodeList for Trial Design dataset TA, variable ARMCD - decoded values" DataType="text">
    <EnumeratedItem CodedValue="Miracle Drug 10mg"/>
    <EnumeratedItem CodedValue="Miracle Drug 20mg"/>
    <EnumeratedItem CodedValue="Placebo"/>
</CodeList>
```

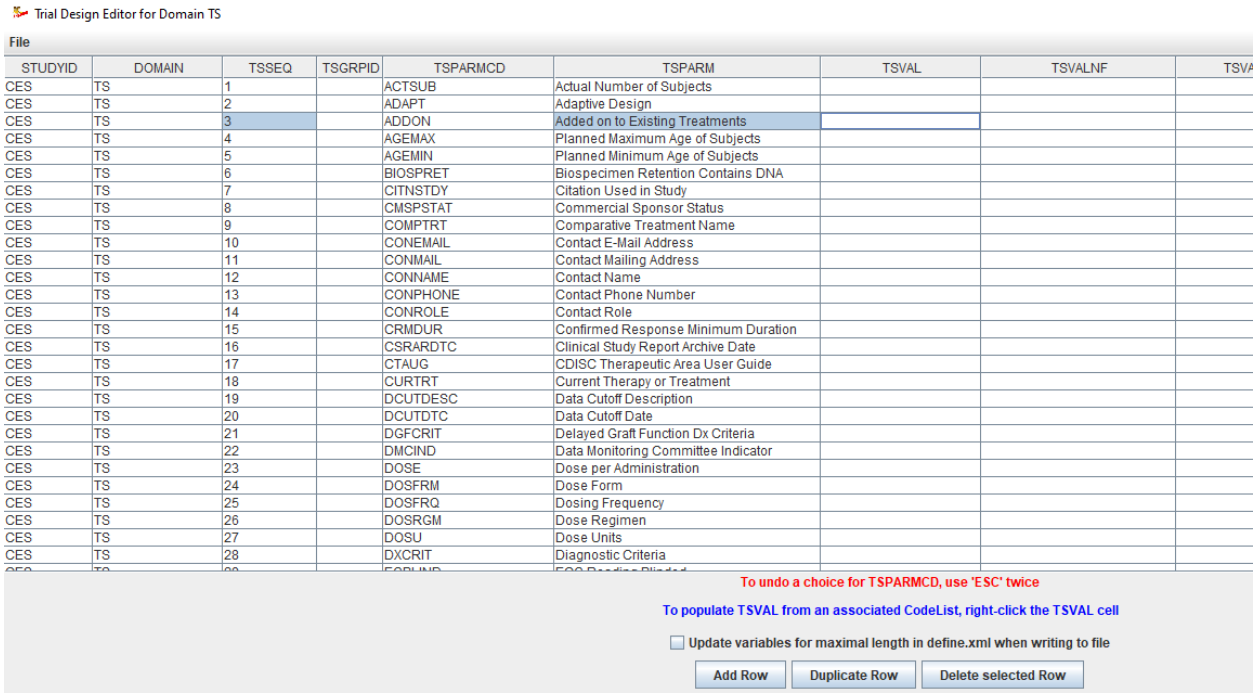
Generating the TS (Trial Summary) dataset

TS (Trial Summary) is another example of a badly designed dataset, a mix of trial design information and post-study collected information. Essentially it is just a parameter-value list (or better "Entity - Attribute - Value" table), with some of the values being coded, some being a number, and some being just free text.
If there are two or more values for the same parameter, this will result in two or more rows ("if you only have SAS-XPT, everything is a table ...).

When one selects to start generating a TS dataset, an additional checkbox becomes available:



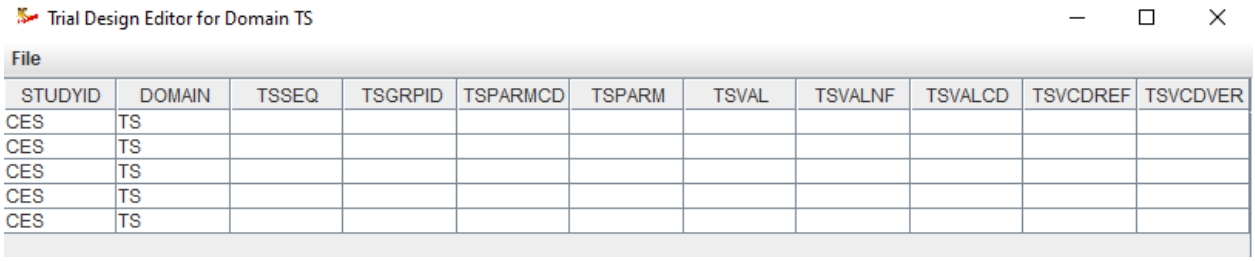
Checking this checkbox is especially interesting when one has or add a large number of TS parameters: in that case, the table will be filled with all TS information from the loaded CDISC controlled terminology, and one will then develop the table mostly by deleting rows (for parameters one doesn't need), and duplicating rows (for parameters with multiple values).
If values for "TSVAL" are coded, this will also be taken into account.
The result with the checkbox "Populate TS table with loaded TS Controlled Terminology" then, after a few seconds, leads to:



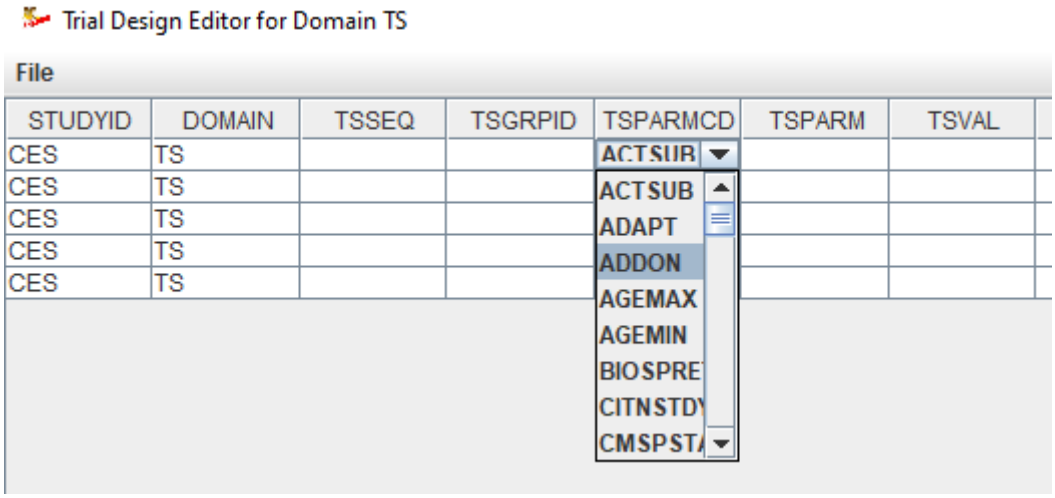
STUDYID	DOMAIN	TSSEQ	TSGRPID	TSPARMCD	TSPARM	TSVAL	TSVALNF	TSVA
CES	TS	1		ACTSUB	Actual Number of Subjects			
CES	TS	2		ADAPT	Adaptive Design			
CES	TS	3		ADDON	Added on to Existing Treatments			
CES	TS	4		AGEMAX	Planned Maximum Age of Subjects			
CES	TS	5		AGEMIN	Planned Minimum Age of Subjects			
CES	TS	6		BIOSPRET	Biospecimen Retention Contains DNA			
CES	TS	7		CITNSTDY	Citation Used in Study			
CES	TS	8		CMSPSTAT	Commercial Sponsor Status			
CES	TS	9		COMPTRT	Comparative Treatment Name			
CES	TS	10		CONEMAIL	Contact E-Mail Address			
CES	TS	11		CONMAIL	Contact Mailing Address			
CES	TS	12		CONNAME	Contact Name			
CES	TS	13		CONPHONE	Contact Phone Number			
CES	TS	14		CONROLE	Contact Role			
CES	TS	15		CRMDUR	Confirmed Response Minimum Duration			
CES	TS	16		CSRARDTC	Clinical Study Report Archive Date			
CES	TS	17		CTAUG	CDISC Therapeutic Area User Guide			
CES	TS	18		CURTRT	Current Therapy or Treatment			
CES	TS	19		DCUTDESC	Data Cutoff Description			
CES	TS	20		DCUTDTC	Data Cutoff Date			
CES	TS	21		DGFCRIT	Delayed Graft Function Dx Criteria			
CES	TS	22		DMCIND	Data Monitoring Committee Indicator			
CES	TS	23		DOSE	Dose per Administration			
CES	TS	24		DOSFRM	Dose Form			
CES	TS	25		DOSFRQ	Dosing Frequency			
CES	TS	26		DOSRGM	Dose Regimen			
CES	TS	27		DOSU	Dose Units			
CES	TS	28		DXCRIT	Diagnostic Criteria			

and one can now start adding information and deleting/duplicating rows.

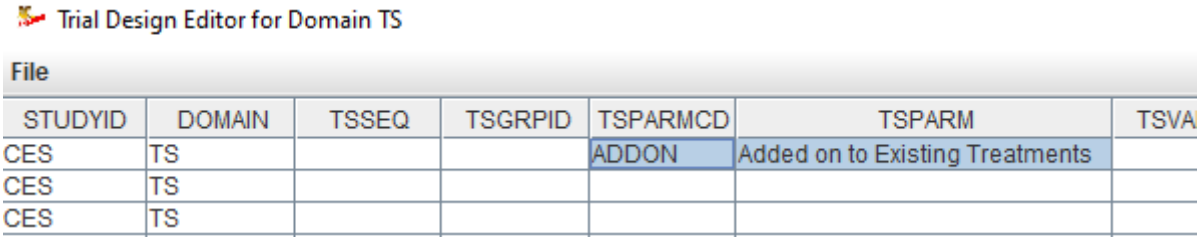
If one has only a small number of TSPARMCD values to submit, one can leave the checkbox "Populate TS table with loaded TS Controlled Terminology" unchecked. This will lead to a table where TSPARMCD and TSPARM are not populated yet:



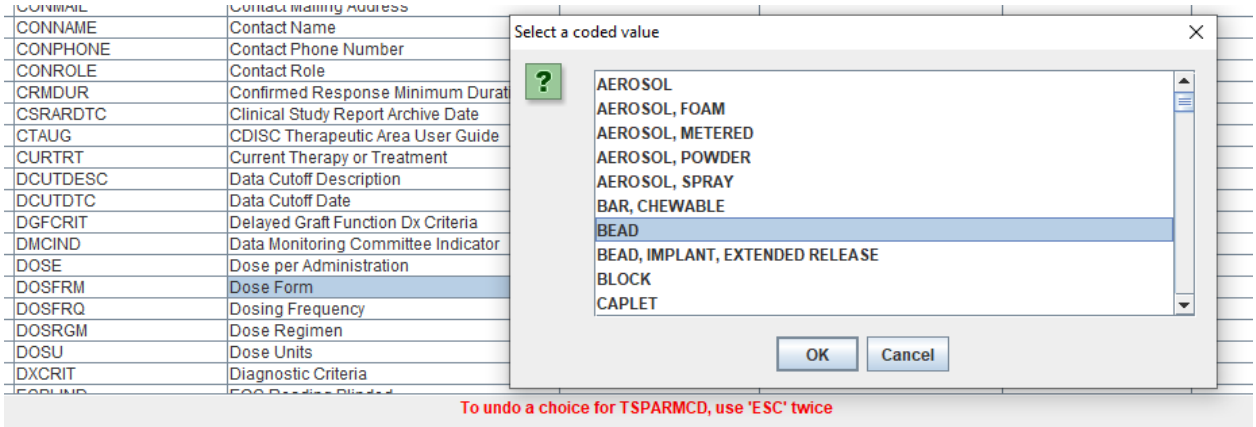
when selecting a TSPARMCD cell, a dropdown is presented:



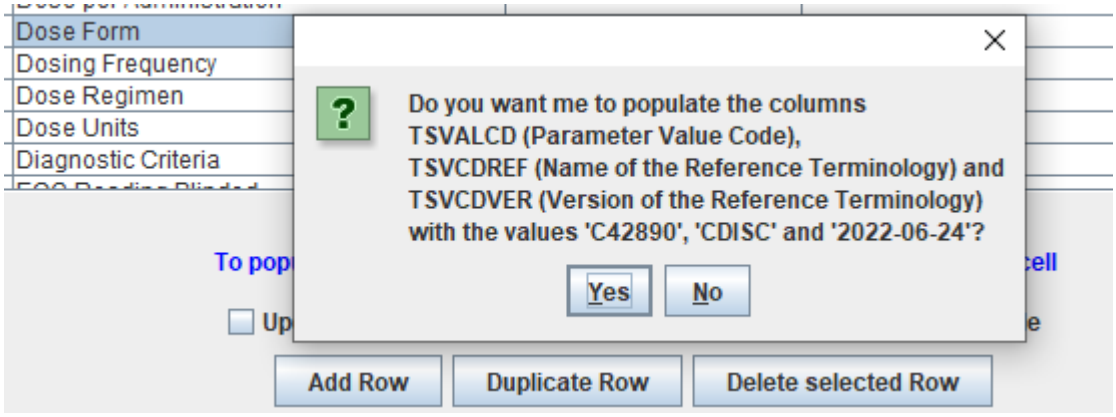
and when one selects one (e.g. ADDON), the corresponding value for TSPARM is automatically selected:



To see whether TSVAL is coded, right-click. If it is, a choice list will be displayed. For example, for TSVAL with TSPARMCD=DOSFRM (dose form):



If one then selects a value, and clicks "OK", the following dialog is displayed:



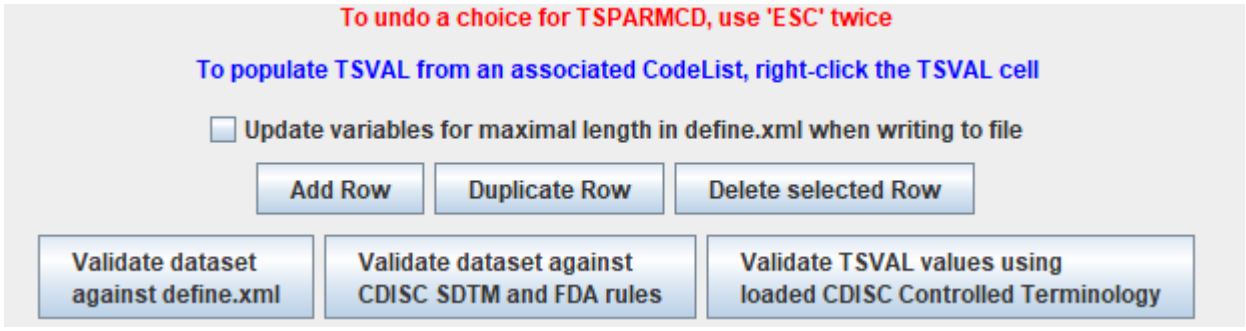
proposing to automatically populate the fields TSVALCD, TSVCDREF and TSVCDVER. When clicking "Yes", this then e.g. leads to:

22		DMCIND	Data Monitoring Committee Indicator					
23		DOSE	Dose per Administration					
24		DOSFRM	Dose Form	BEAD		C42890	CDISC	2022-06-24
25		DOSFRQ	Dosing Frequency					
26		DOSRGM	Dose Regimen					

If there is no associated controlled terminology, a message is displayed: e.g.:

Remark that it will be your responsibility to add the correct values for TSSEQ. You will need to start at "1" again for each unique value of TSPARMCD. Automated assignment of TSSEQ is a feature foreseen for the next release.

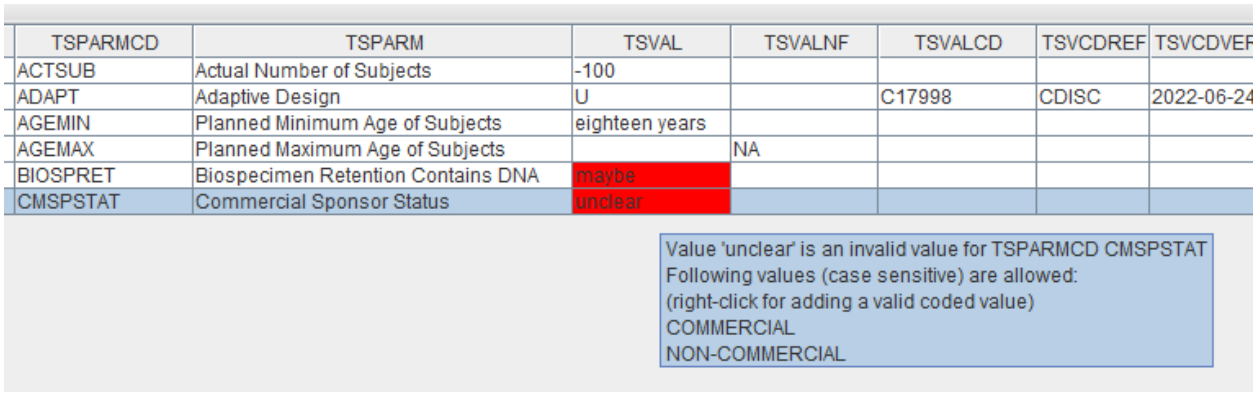
Now have a look at the lower part of the window:



It has 3 "validate" buttons: one to validate the structure against the define.xml requirements. This just checks whether all "required" fields are filled, nothing more. So, don't expect too much of it. The second button "Validate dataset against CDISC SDTM and FDA rules", has been put out of function in version 4.1 of the software. Reason is that we are waiting for CORE, which will be "the

only truth for validation rules"[4].
The third button "Validate TSVAL values against CDISC Controlled Terminology" does exactly what it says. It takes the value of TSVAL, checks when it is supposed to be coded, and if so, compares the value against a CDISC codelist, when one is provided. At the same time, it also checks the 1:1 correspondence of TSPARMCD and TSPARM.

The result of such a validation can e.g. be:



This validation however does not protect you from adding illogical values, like the value for TSVAL "eighteen years" for AGEMIN. This is as TSVAL is always a string, even when logically, an integer or e.g. an ISO-8601 duration is expected. Also here, we are waiting for CDISC CORE to have such rules implemented[5].

Remark that whether a value for TSVAL is coded or not may depend on the version of the controlled terminology used! So, it is always a good idea to use the latest version of all CDISC controlled terminology.

[4] Pinnacle21 does not allow anymore to call their software from within other applications. So, we had to disable that. Anyway, Pinnacle21 has become famous for the large amount of "false positives" and pretty bad quality of the software. This is why we decided to completely move to CDISC CORE.
[5] Problematic is that the FDA (i.e. Pinnacle21) is changing these all the time, and that they are not always transparent.