SDTM-ETL 4.0 Creating Related Records (RELREC)

Author: Jozef Aerts, XML4Pharma

Last update: 2020-10-16

The Related Records (RELREC) domain

Often, records between domains/datasets are related. The typical example is when a medication (goes in CM - Concomitant Medication) is administered due to an adverse event (goes into AE - Adverse Events). There are however many more cases where we want to document such relations, such as a lab value due to some event (not necessarily an adverse one) or another observation.
As SDTM/SEND is limited to tables, and these tables are NOT relational tables, the SDTM team had to develop another mechanism: the RELREC "Related Records" table. Although their choice was surely not the best one, it is as it is.

The RELREC domain allows to describe that there is a relation between two data points, a data point and a group of datapoints, and between different groups of data points, but not what that relation exactly is.

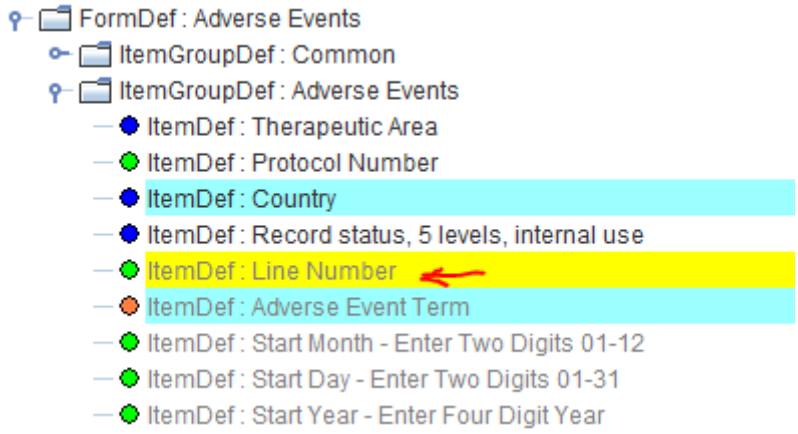
In this tutorial, we will develop a typical relationship between an adverse event and a concomitant medication, where several medications may have been administered for one single adverse event. In the example, we will use the classic case where each adverse event is documented in an adverse event (AE) CRF as a "AE line number", and in the "Concomitant Medications" form, this "line number" is referenced.

The AE dataset definition

In this tutorial, we assume that you have already created mappings for the "AE" form. These will typically contain mappings for AETERM (the adverse event term, i.e. short description of the adverse event), start and end dates (AESTDTC, AEENDTC) when known, the severity (AESEV), relationship to study drug (AEREL), the outcome (AEOUT), and of course mappings for the "severe adverse event" category variables (the "AESxxx" variables).
Another variable that is very often used is AESPID (sponsor identifier), which is often mapped to the "line" number on the AE CRF. According to the SDTM-IG v.3.2:

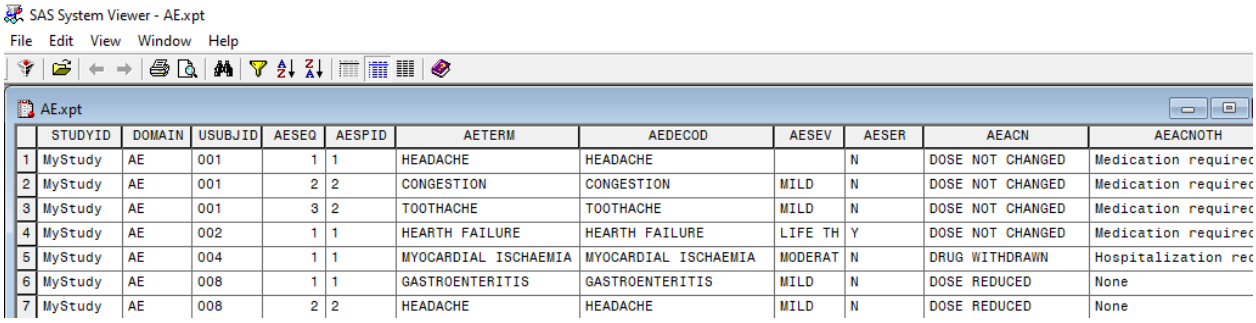
AESPID	Sponsor-Defined Identifier	Char		Identifier	Sponsor-defined identifier. It may be pre-printed on the CRF as an explicit line identifier or defined in the sponsor's operational database. Example: Line number on an Adverse Events page.	Perm

If we look into the ODM tree on the left side of the screen, we see that we do have such an item:



so, we will map this to AESPID. In most cases, a simple-drag and drop will suffice.

When we then create the SDTM dataset, we find something like:

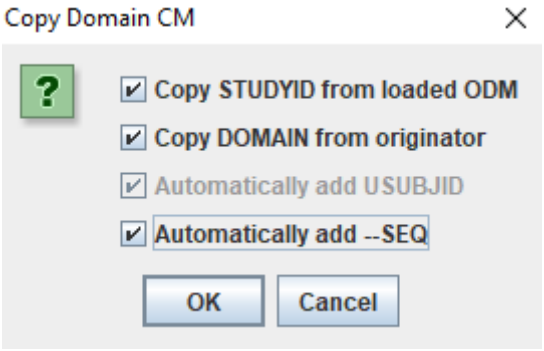


	STUDYID	DOMAIN	USUBJID	AESEQ	AESPID	AETERM	AEDECOD	AESEV	AESER	AEACN	AEACNOTH
1	MyStudy	AE	001	1	1	HEADACHE	HEADACHE		N	DOSE NOT CHANGED	Medication required
2	MyStudy	AE	001	2	2	CONGESTION	CONGESTION	MILD	N	DOSE NOT CHANGED	Medication required
3	MyStudy	AE	001	3	2	TOOTHACHE	TOOTHACHE	MILD	N	DOSE NOT CHANGED	Medication required
4	MyStudy	AE	002	1	1	HEARTH FAILURE	HEARTH FAILURE	LIFE TH	Y	DOSE NOT CHANGED	Medication required
5	MyStudy	AE	004	1	1	MYOCARDIAL ISCHAEMIA	MYOCARDIAL ISCHAEMIA	MODERAT	N	DRUG WITHDRAWN	Hospitalization rec
6	MyStudy	AE	008	1	1	GASTROENTERITIS	GASTROENTERITIS	MILD	N	DOSE REDUCED	None
7	MyStudy	AE	008	2	2	HEADACHE	HEADACHE	MILD	N	DOSE REDUCED	None

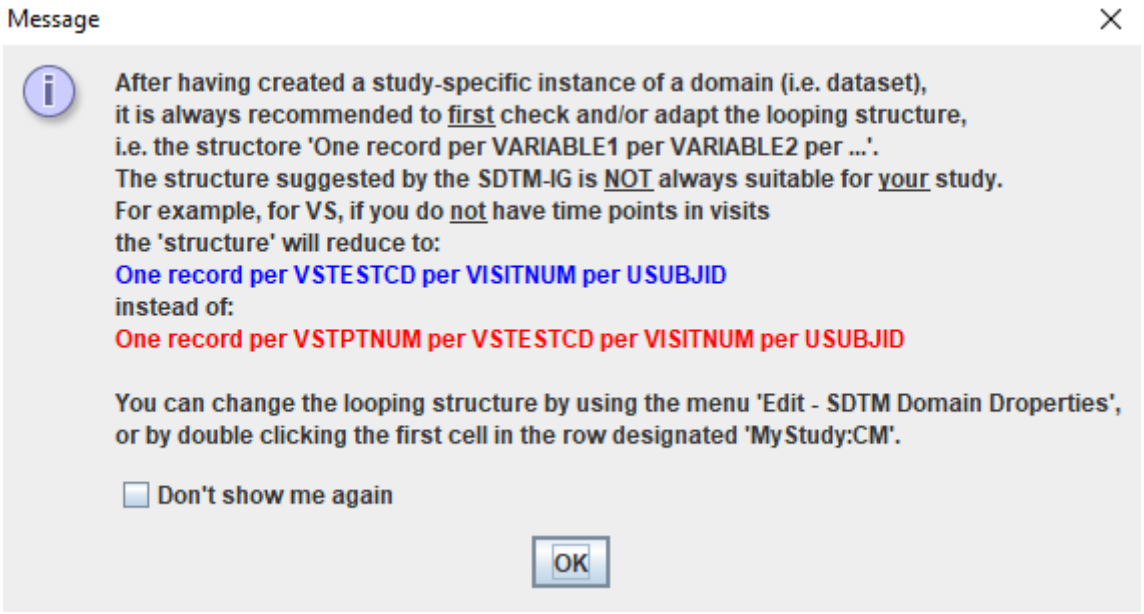
With AESPID populated with the line number from the CRF. If only a single CRF is used to capture all adverse events over all the visits, the value of AESPID will often correspond to the value of AESEQ (sequence number). The difference is that AESEQ is derived, whereas AESPID directly comes from the CRF.

Creating the CM dataset mapping

In order to generate the mappings for the CM domain, simply drag-and-drop the “CM” row from the template (upper part of the SDTM table) to the bottom. A dialog pops up:



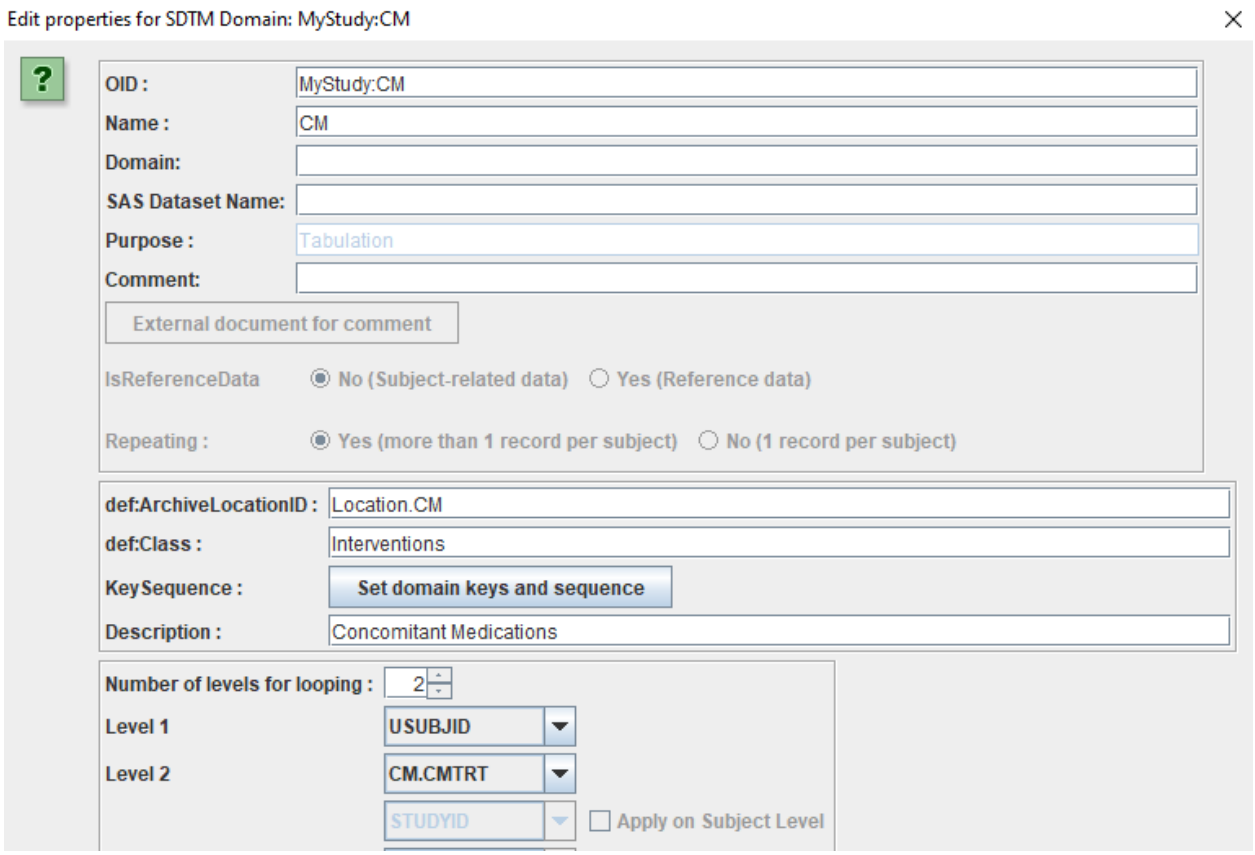
As we also want the value of CMSEQ to be automatically generated, we check the checkbox “Automatically add --SEQ”. After clicking “OK”, we get;



telling us that we should have a look at the structure (i.e. looping order) of our dataset definition. We can set that we do not want to be reminded about this anymore, by checking the "Don't show me again" checkbox. Clicking "OK" than leads to a "CM" row at the bottom of the table:

MyStudy:GLOBAL	RFSTDTC	RFENDTC	RFXSTDTC	RFXENDTC	
MyStudy:DM	STUDYID	DOMAIN	USUBJID	SUBJID	DM.RFSTDTC
MyStudy:EC	STUDYID	DOMAIN	USUBJID	EC.ECSEQ	EC.ECGRPID
MyStudy:VS	STUDYID	DOMAIN	USUBJID	VS.VSSEQ	VS.VSGRPID
MyStudy:AE	STUDYID	DOMAIN	USUBJID	AE.AESEQ	AE.AEGRPID
MyStudy:CM	STUDYID	DOMAIN	USUBJID	CM.CMSEQ	CM.CMGRPID

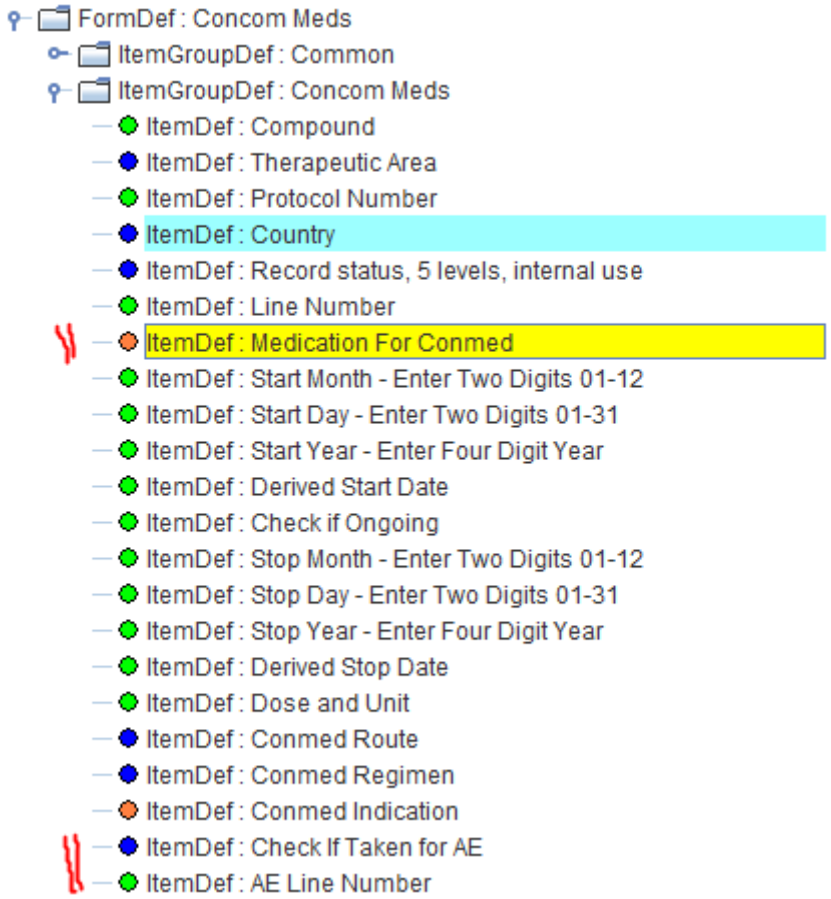
We see that the fields "STUDYID", "DOMAIN", "STUDYID", "CM.CMSEQ" have already been "greyed", to indicate that these already contain a mapping.
If we then double click "MyStudy:CM" (or select it and then use "Edit - SDTM Domain Properties"), the "properties" dialog for CM shows up:



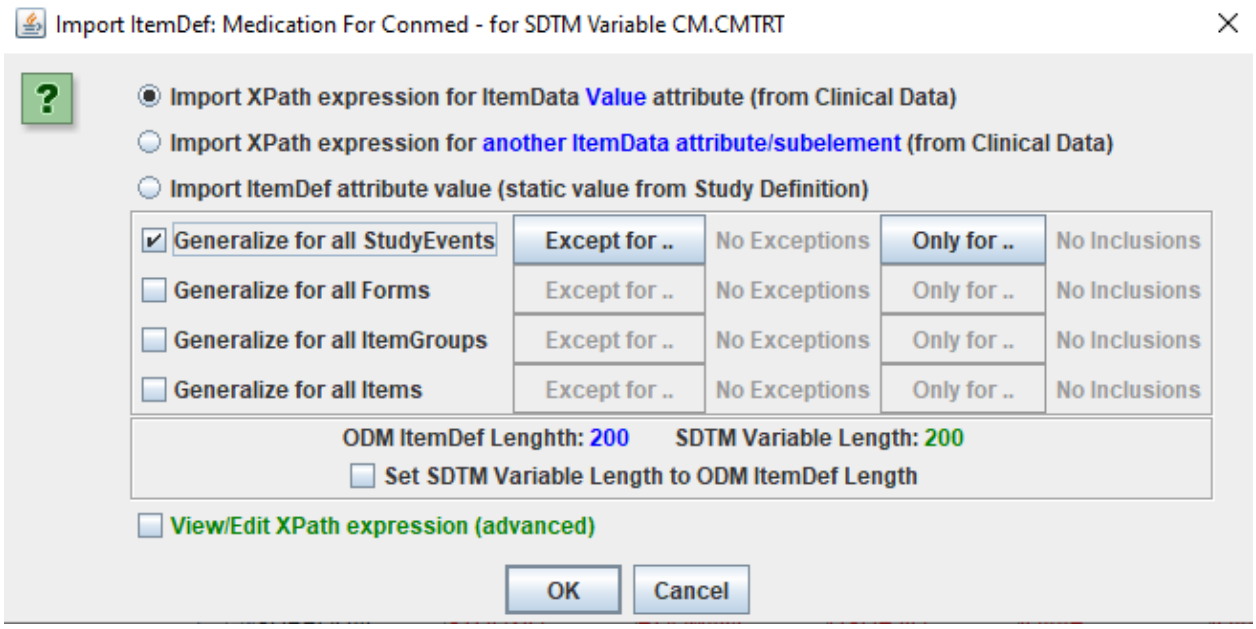
Important here is that we set the right structure, i.e. the variables over which we want to loop. In this case, it is very simple, and we can accept the proposal to first loop over the subjects and then over the values of "CMTRT" ("Reported Name of Drug, Med, or Therapy").
Also, do not forget the "keys" for the define.xml by using the "Set domain keys and sequence", and then adding "STUDYID", "USUBJID", and "CMTRT". This will allow the reviewer to understand how you have defined uniqueness of records.

After having done so, we can start mapping.

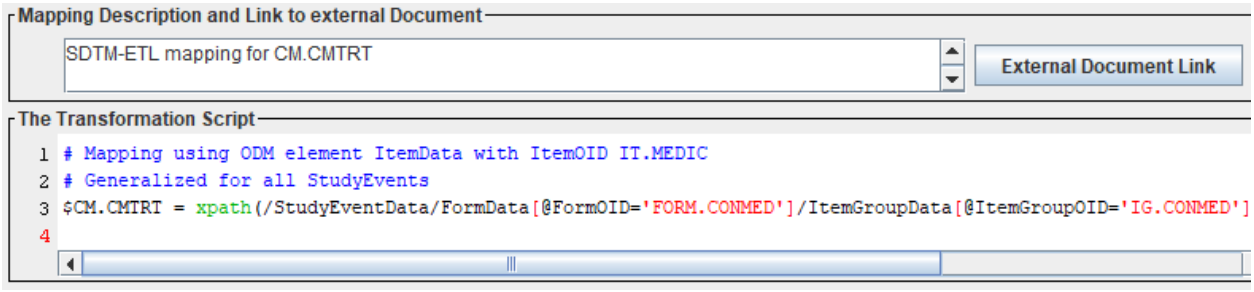
As CMTRT is a key variable, let us do the mapping first. An obvious candidate in the ODM tree is the item "Medication For Conmed":



Also mind the item “AE Line number”, we will need it later.
We can simply drag-and-drop it to the cell CM.CMTRT, resulting in a dialog:

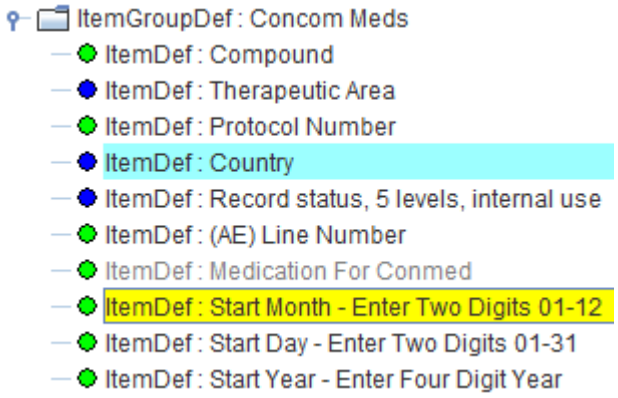


If there is more than one visit in which the medication is captured, do not forget to check the checkbox “Generalize for all StudyEvents”. After clicking “OK”, the mapping script is generated and displayed:

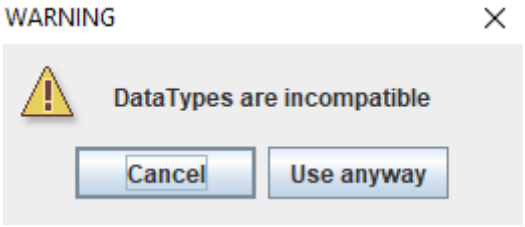


At the moment, there is nothing more to do.

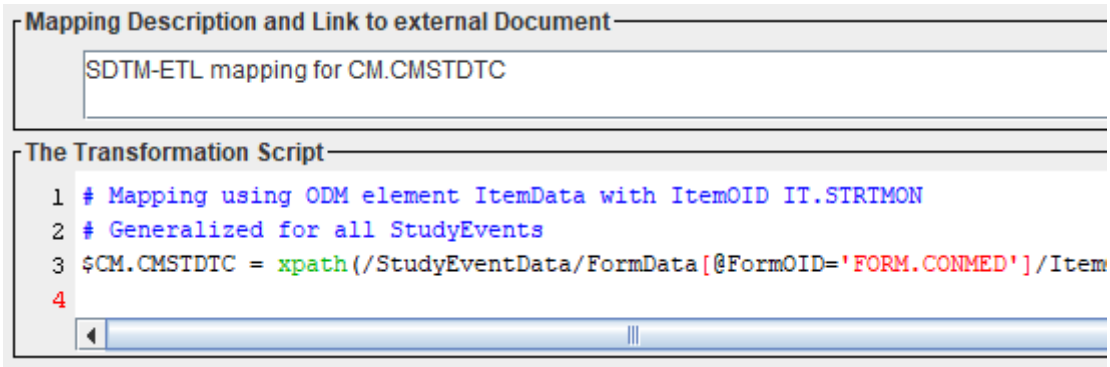
In CM, very important are start and end date of the medication. In the ODM tree, we see the items:



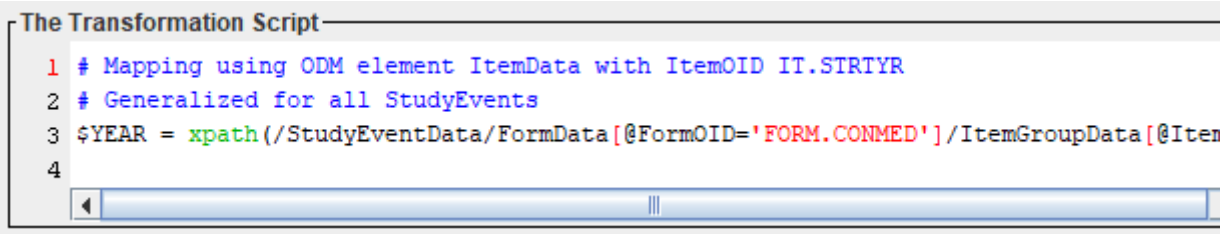
“start month”, “start day” and “start year”. We will use these to form the CM start date in CMSTRDTC (“Start Date/Time of Medication”). First just drag and drop from “Start Year” to “CM.CMSTRDTC”. Accept to use the value itself “Import XPath Expression from ItemData Value”). The system will then protest that the data types are incompatible (it expects a full date):



Click “Use anyway”, leading to a first part of the mapping script being generated:

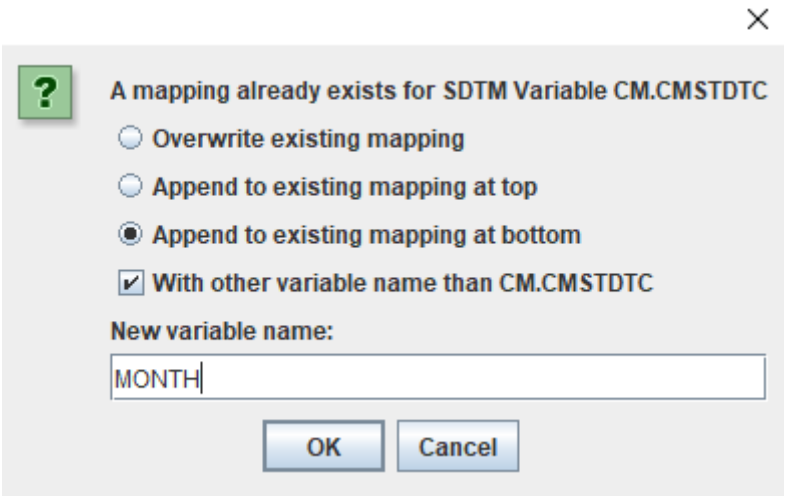


Then rename (just by selecting and typing) the variable “$CM.CSSTDTC” into e.g. “$YEAR”.

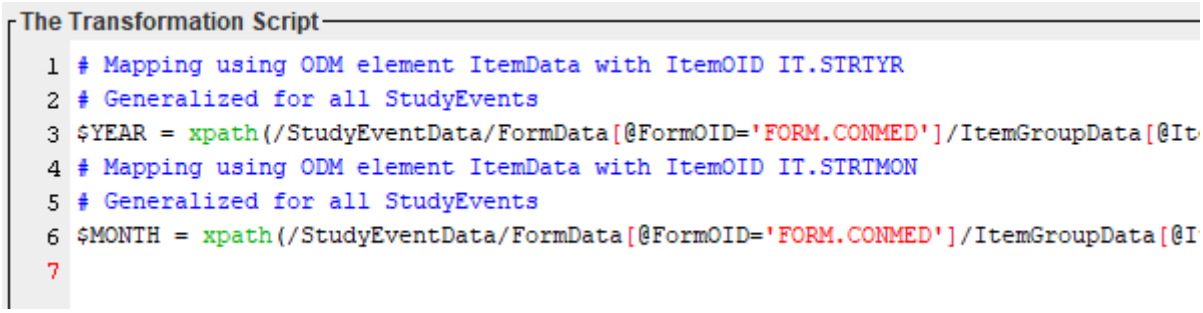


Click “OK” to accept.

Then drag-and-drop “Start Month” to the same cell. As there is already a mapping in the cell, the system asks us whether we want to overwrite the existing mapping, or append. It also allows us to rename the variable name for this mapping part, which we gratefully accept and set to “MONTH”.



Again, we can accept all the logical proposals the system makes, and see a new group of lines appearing in the mapping script:



Then we do almost exactly the same for “Start Day”, assigning the temporary variable the name “DAY”, leading to:

The Transformation Script

```
1  # Mapping using ODM element ItemData with ItemOID IT.STRTYR
2  # Generalized for all StudyEvents
3  $YEAR = xpath(/StudyEventData/FormData[@FormOID='FORM.CONMED']/ItemGroupDa
4  # Mapping using ODM element ItemData with ItemOID IT.STRTMON
5  # Generalized for all StudyEvents
6  $MONTH = xpath(/StudyEventData/FormData[@FormOID='FORM.CONMED']/ItemGroupD
7  # Mapping using ODM element ItemData with ItemOID IT.STRTDAY
8  # Generalized for all StudyEvents
9  $DAY = xpath(/StudyEventData/FormData[@FormOID='FORM.CONMED']/ItemGroupDat
10
```

All we still need to do is to “glue” the 3 parts of the start date together, using the “concat(” (concatenation) function, and assigning it to the final variable $CM.CMSTDTC. We type:

```
7  # Mapping using ODM element ItemData with ItemO]
8  # Generalized for all StudyEvents
9  $DAY = xpath(/StudyEventData/FormData[@FormOID='
10 $CM.CMSTDTC =
```

and then either can use the “concat” button on the panel near the bottom, but with some experience, it will often be faster to just type:

```
8  # Generalized for all StudyEvents
9  $DAY = xpath(/StudyEventData/FormData[@FormOID='FORM.CONMED']/ItemGroupDa
10 $CM.CMSTDTC = concat($YEAR,'-',$MONTH,'-',$DAY);
```

where the color coding of course helps considerably.
Don’t forget the semicolon at the end!
Click “OK” to store the scrip in memory.

We can now already try to execute the mapping in order to see what we already get.
We use “Transform - Generate Transform (XSLT) Code for SAS-XPT”, followed by “Execute Transformation (XSLT) Code”, then select a file with ODM clinical data, and finally click “Execute Transformation on Clinical Data”.
All these steps are explained in much more detail in earlier tutorials explaining the basics.

We then e.g. get:

CM.xpt

	STUDYID	DOMAIN	USUBJID	CMSEQ	CMTRT	CMSTDTC
1	MyStudy	CM	001	1	TYLENOL	2006-06-10
2	MyStudy	CM	001	2	SUDAFED	2006-06-11
3	MyStudy	CM	002	1	TYLENOL	2006-06-10
4	MyStudy	CM	003	1	TYLENOL	2006-8-10
5	MyStudy	CM	004	1	TYLENOL	2006-06-10
6	MyStudy	CM	005	1	TYLENOL	2006-03-10
7	MyStudy	CM	005	2	ASPIRIN	1999-03-10
8	MyStudy	CM	005	3	IBUPROFEN	1999-06-10
9	MyStudy	CM	006	1	TYLENOL	1999-06-10
10	MyStudy	CM	008	1	TYLENOL	2006-07-07
11	MyStudy	CM	008	2	IBUPROFEN	2006-07-07
12	MyStudy	CM	009	1	TYLENOL	1999-06-10
13	MyStudy	CM	010	1	TYLENOL	1999-06-10
14	MyStudy	CM	011	1	TYLENOL	1999-06-10
15	MyStudy	CM	012	1	TYLENOL	1999-06-10

Remark that in case you have partial dates, you might need to write a litte more script (using “if” statements) to take care that your partial dates are correctly formatted. See the SDTM-IG about correctly formattings dates and times in ISO-8601 format.

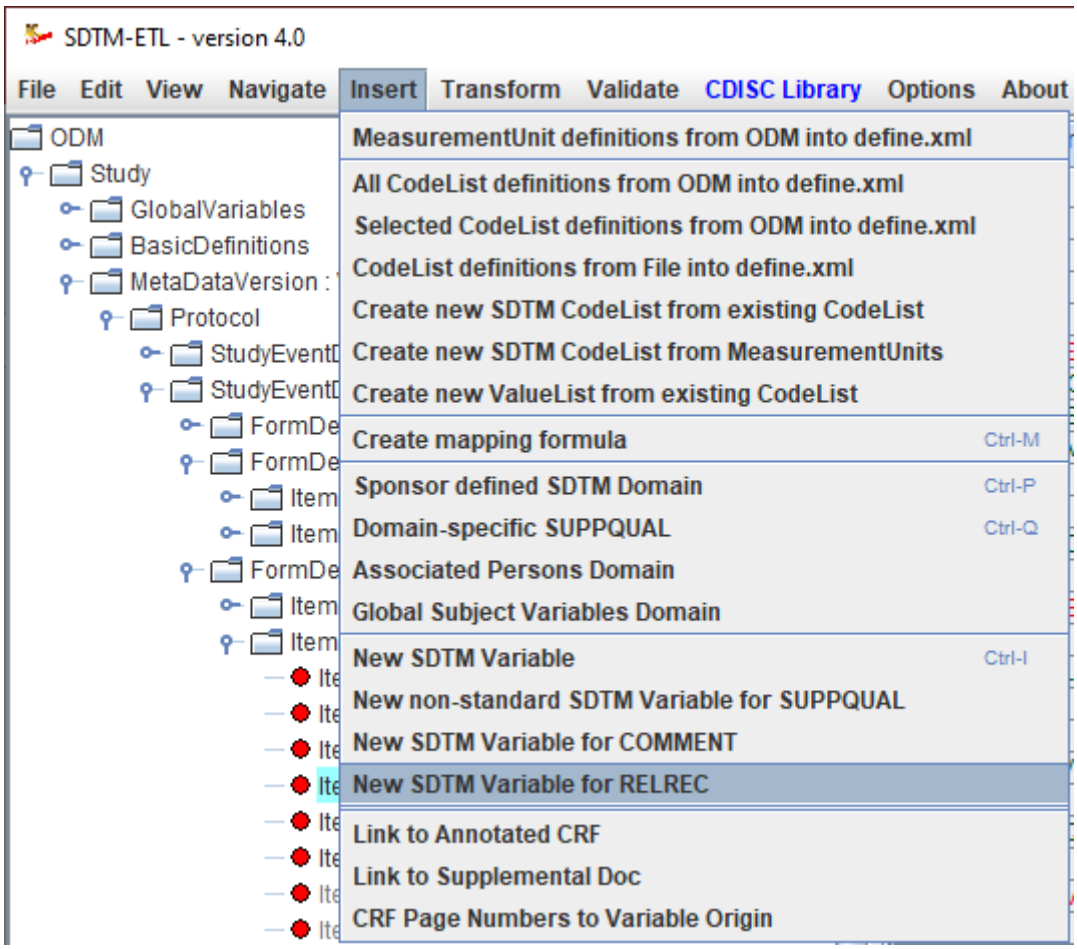
You can then now do similar mappings very easily for all other SDTM variables in CM for which you have information. This will mostly be possible by simple drag-and-drop, with in some cases, a little amount of mapping script (e.g. to combine date parts).

Setting up the relationships

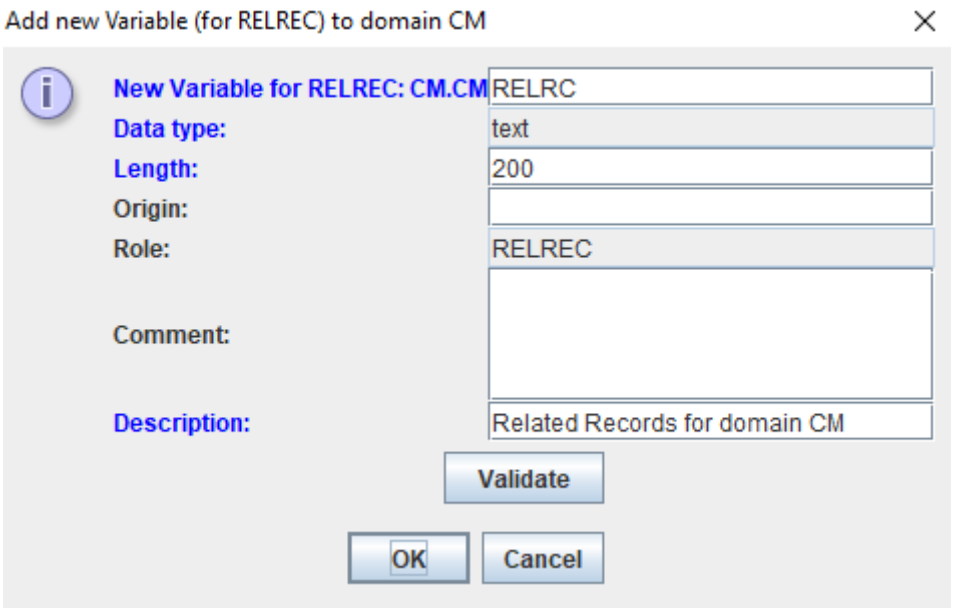
In SDTM-ETL, there is usually no need to set up a separate RELREC (“Related Records”) dataset in order to generate the relations: in most cases, you can just define a special “relation” variable that, at execution time, will be split off, and take care that a RELREC dataset is automatically set up and populated for you.

In our case here, the “adverse event” is the source of the relation, and the “medication” is the target of the relation, i.e. the medication is administered as a consequence of the adverse event. “Relation variables” are always defined at the “target” domain/dataset, so here in “Concomitant Medications”.

In order to set up the relation, select a cell in “MyStudy:CM” and use the menu “Insert - New SDTM Variable for RELREC”:



This then leads to the following dialog:

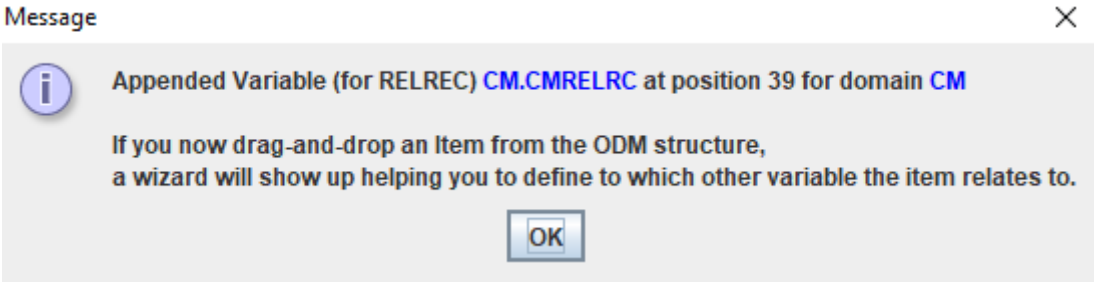


For the “Length”, set a value that you think is suitable for what is the relation is about. In our case, the relation is about AESPID, so you can set a low value here. The exact value is of minor importance here, and in the case of Dataset-XML, it is not of importance at all.
In the field “Origin”, set “CRF” in case the relation is explicetly described in the CRF.
You can add a Comment, describing the relation, e.g. “Relation between CM treatment and AE line number”. This description will however not appear in the RELREC dataset (blame SDTM!), but

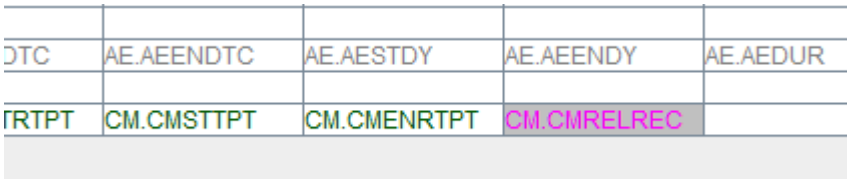
will just be a comment in the mapping script, so that you later understand what you did.

After clicking “OK”, we get:

Followed by:

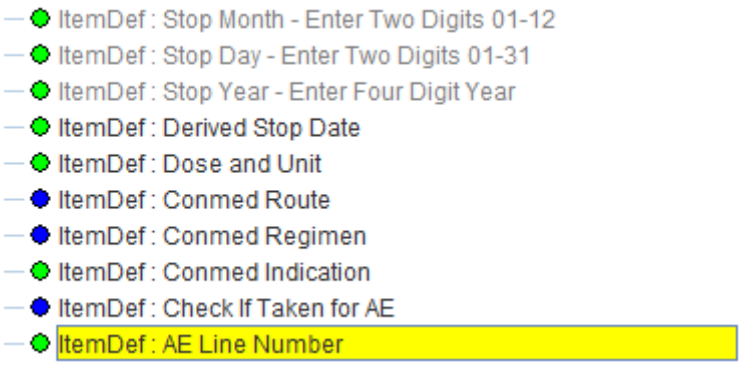


also followed by the advice to, in the next step, drag-and-drop one of the “CM” fields to the cell that just has been created. We first simply get:

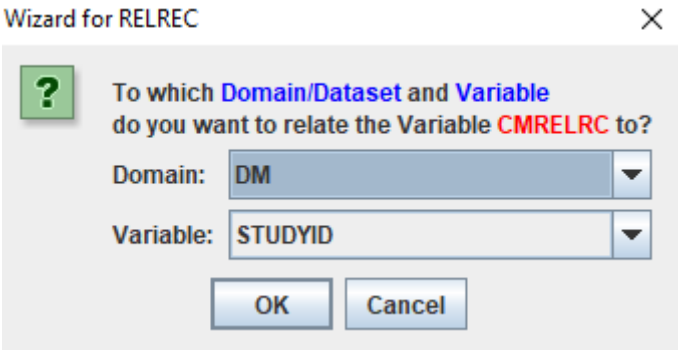


i.e. an additional (still empty) variable has been set up for storing the relation (that is why it has this special purple color).

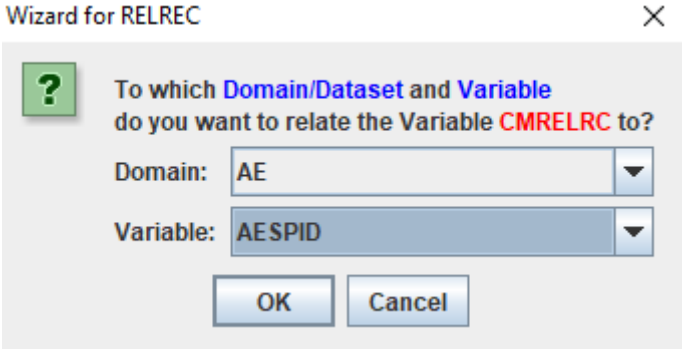
As the relation is from the “AE line number” in CM (NOT “line number”!), now drag-and-drop this item from the ODM tree into the new cell:



After asking to accept whether the value itself of the datapoint needs to be used, a wizard dialog shows up:



This wizard asks about the SDTM variable that is the “target” of the relationship. In our case, it is the “AE line number”, which we mapped to AESPID.
So we need to select “AE” for the domain, and “AESPID” for the variable:



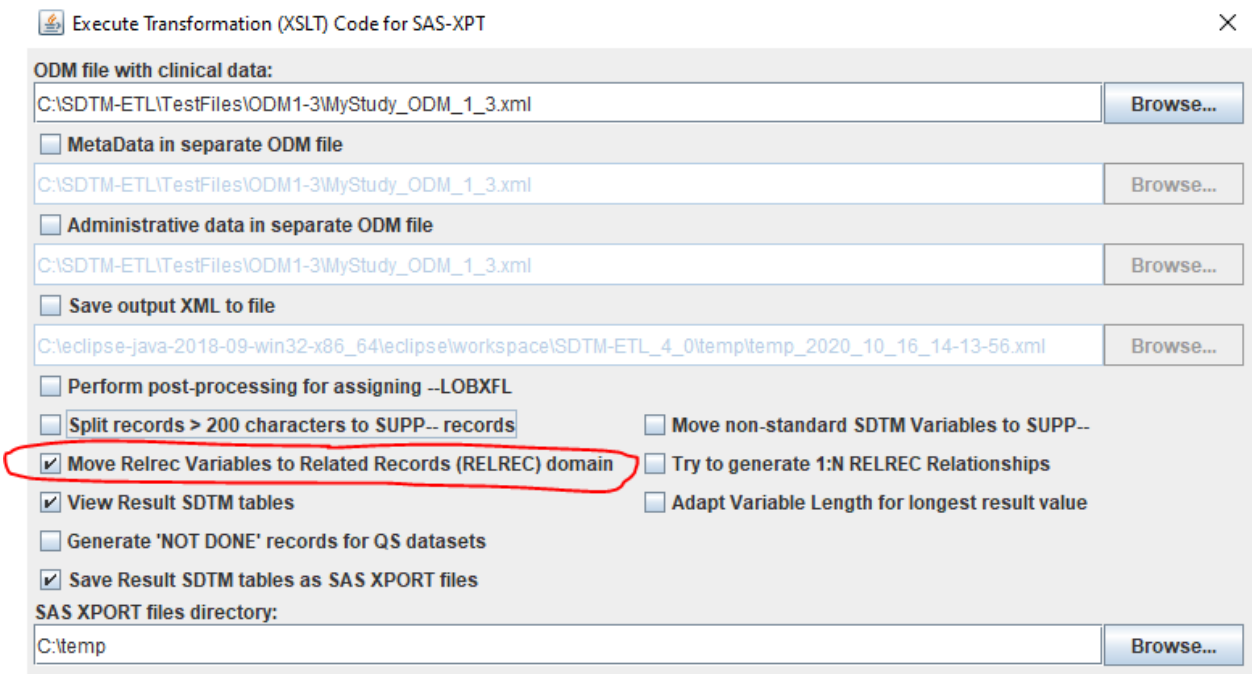
After clicking “OK”, the mapping script is generated:

The Transformation Script

```
1 # Mapping using ODM element ItemData with ItemOID CM.CMRELRC
2 # for RELREC record with a relationship to variable AESPID in domain AE
3 $TEMP = xpath(/StudyEventData/FormData[@FormOID='FORM.CONMED']/ItemGroupData[@ItemGroupOID='IG.CONMED']/ItemData[@ItemOID='IT.AE_LINE
4 $CM.CMRELRC = relrec('AE','AESPID',$TEMP);
```

In line #3, a temporary variable is generated, taking the value of “AE line number” in the CM form, and in line #4, it is stated that the relation is between “AESPID” in “AE” (which we already generated before) and the temporary variable keeping the line number in CM.
Like this, the statement reads as that there is a relation between the “AE line number” in CM and the line number in AE.

We can now execute the mappings again on clinical data. Once again, we use “Transform - Generate Transformation (XSLT) Code for SAS-XPT), and go through the usual procedure until we once again come at:



It is now of ultimate importance that the checkbox “Movbe Relrec Variale to Related Records (RELREC) domain”. Usually, the system already checks it when it detects that there is such a special variable defined.

Clicking “Execute Transformation on Clinical Data” then executes the mappings, and automatically generates a RELREC dataset. It looks like:

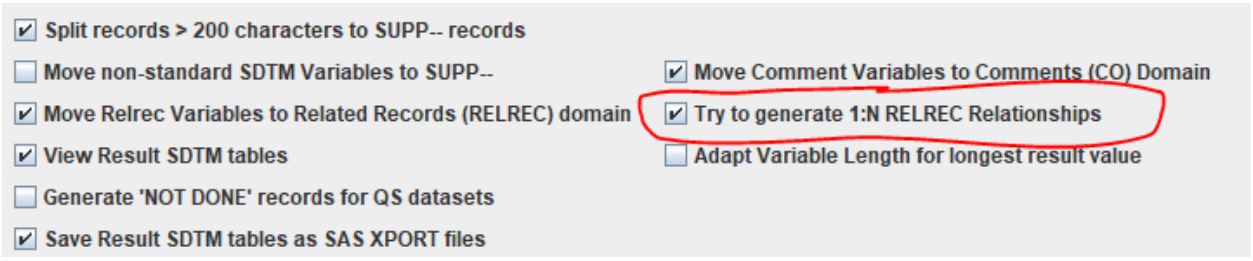
RELREC.xpt

	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	RELTYPE	RELID
1	MyStudy	CM	001	CMSEQ	1		1
2	MyStudy	AE	001	AESPID	1		1
3	MyStudy	CM	001	CMSEQ	2		2
4	MyStudy	AE	001	AESPID	2		2
5	MyStudy	CM	008	CMSEQ	1		3
6	MyStudy	AE	008	AESPID	2		3
7	MyStudy	CM	008	CMSEQ	2		4
8	MyStudy	AE	008	AESPID	2		4

Trying to generate 1:N relationships

In the above result table, we see that 2 concomitant medications were provided for the same AE for subject 008. Both are depicted as 1:1 relationships.
In such cases, we can ask the software to try to do “grouping”, and to generate a 1:N relationship.

In order to do so, the checkbox “” must be checked.



In our case, the result then is:

RELREC.xpt

	STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	RELTYPE	RELID
1	MyStudy	AE	001	AESPID	1		1
2	MyStudy	CM	001	CMSEQ	1		1
3	MyStudy	AE	001	AESPID	2		2
4	MyStudy	CM	001	CMSEQ	2		2
5	MyStudy	AE	008	AESPID	2		3
6	MyStudy	CM	008	CMSEQ	1		3
7	MyStudy	CM	008	CMSEQ	2		3

We see that we now have only 7 rows, and that we have 3 records for RELID=3, stating that for subject 008, the AE with line number 2 is related to 3 concomitant medications.

Further materials

A somewhat older movie explaining the above steps can be found at:

http://xml4pharmaserver.com/SDTM-ETL/P12/P12.html