SDTM-ETL 3.2 User Manual and Tutorial

Author: Jozef Aerts, XML4Pharma

Last update: 2017-07-06

Auto-generation of comments and putting them in the Comments (CO) domain

There are two ways of populating (instances of) the CO domain. The first is to first create a study-specific instance of the CO domain (by drag-and-drop to the bottom), and then go ahead as with any other dataset. When doing so, don't care about the 200 character limitation of COVAL (a relict of the outdated SAS-XPT format), as this will be care of automatically by the software when generating the SAS datasets ("automated splitting").

At the ODM side, comments can often be simply have been captured as data points, so can easily be used in the mappings. Comments can also come as "annotations" on data points, such as e.g.:

```
<FormData FormOID="FORM.AE">
    <ItemGroupData ItemGroupOID="IG.AE" ItemGroupRepeatKey="1" TransactionType="Insert">
        <AuditRecord>
            <UserRef UserOID="USR.inv001"/>
            <LocationRef LocationOID="LOC.site002"/>
            <DateTimeStamp>2006-02-04T14:21:32-05:00</DateTimeStamp>
            <SourceID>46898304</SourceID>
        </AuditRecord>
        <ItemData ItemOID="IT.TAREA" Value="ONC"/>
        <ItemData ItemOID="IT.PNO" Value="143-02"/>
        <ItemData ItemOID="IT.SCTRY" Value="USA"/>
        <ItemData ItemOID="IT.F_STATUS" Value="V"/>
        <ItemData ItemOID="IT.LINE_NO" Value="1"/>
        <ItemData ItemOID="IT.AETERM" Value="HEADACHE">
            <Annotation SeqNum="1" TransactionType="Insert">
                <Comment SponsorOrSite="Site">Could be due to the subject also having a viral infection</Comment>
            </Annotation>
        </ItemData>
        <ItemData ItemOID="IT.AESTMON" Value="04"/>
```

In such a case, you might need to extend the XPath expression yourself.
For example, for the above example, you may need:

$AECOMMENT = xpath(/StudyEventData/FormData[@FormOID='FORM.AE']/ItemGroupData [@ItemGroupOID='IG.AE']/ItemData[@ItemOID='IT.AETERM']/Annotation/Comment

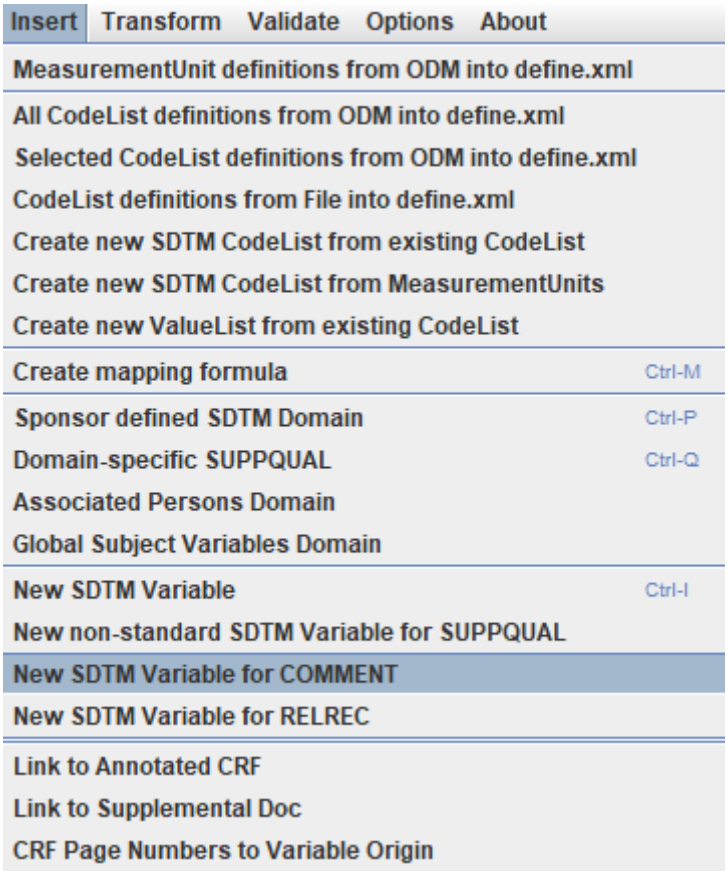
Such an XPath expression can however easily be generalized for more use cases.

The easiest way however is to have the comments auto-generated.

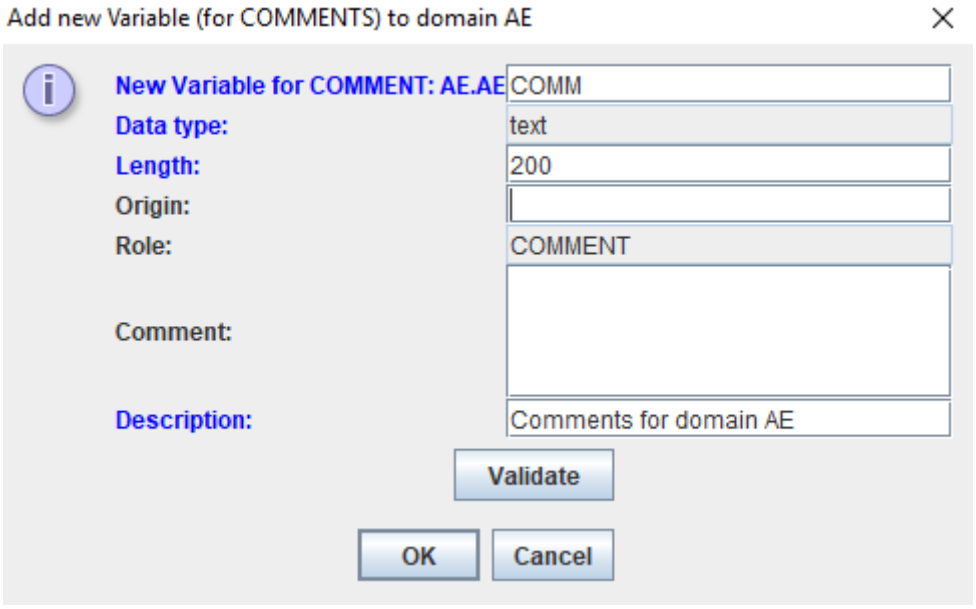
We will develop an example for the AE domain where we derive a comment from the "annotation" on AETERM in the ODM data.

First select a cell from the study-specific AE row, then use the menu "Insert – New SDTM Variable for COMMENT"[1]

[1] In the case of SEND, it will of course be "New SEND Variable for COMMENT"

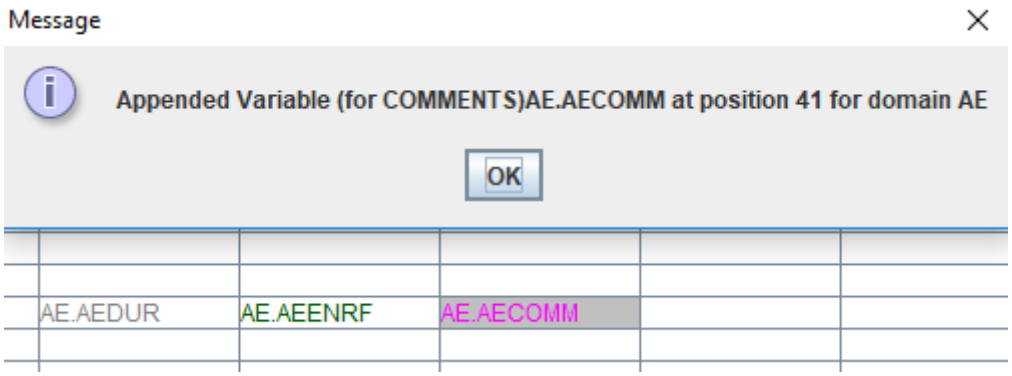


This will lead to:

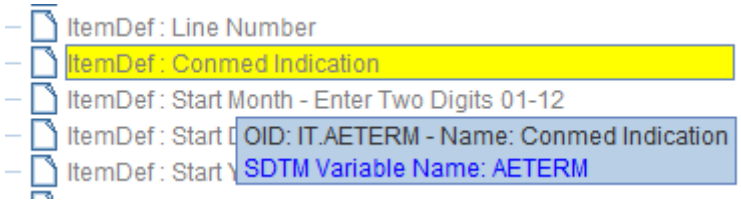


The system will auto-generate an OID (first line), but you can easily change it if you want. The data type is fixed to "text" (which makes sense), the maximal length is preset to 200. You can however select a smaller value when you are sure that the comments will never be longer than your value.
For "Origin" you can set whatever you like, it is not an "Origin" in the sense of define.xml. It is just internal information which will later not appear in the final define.xml anyway. It can however be useful as internal information. You might e.g. put "AE form, comments field", or "Comments form".
You can also add a further comment, also for internal purposes.
The "Validate" button can be used to check whether sufficient information is provided.

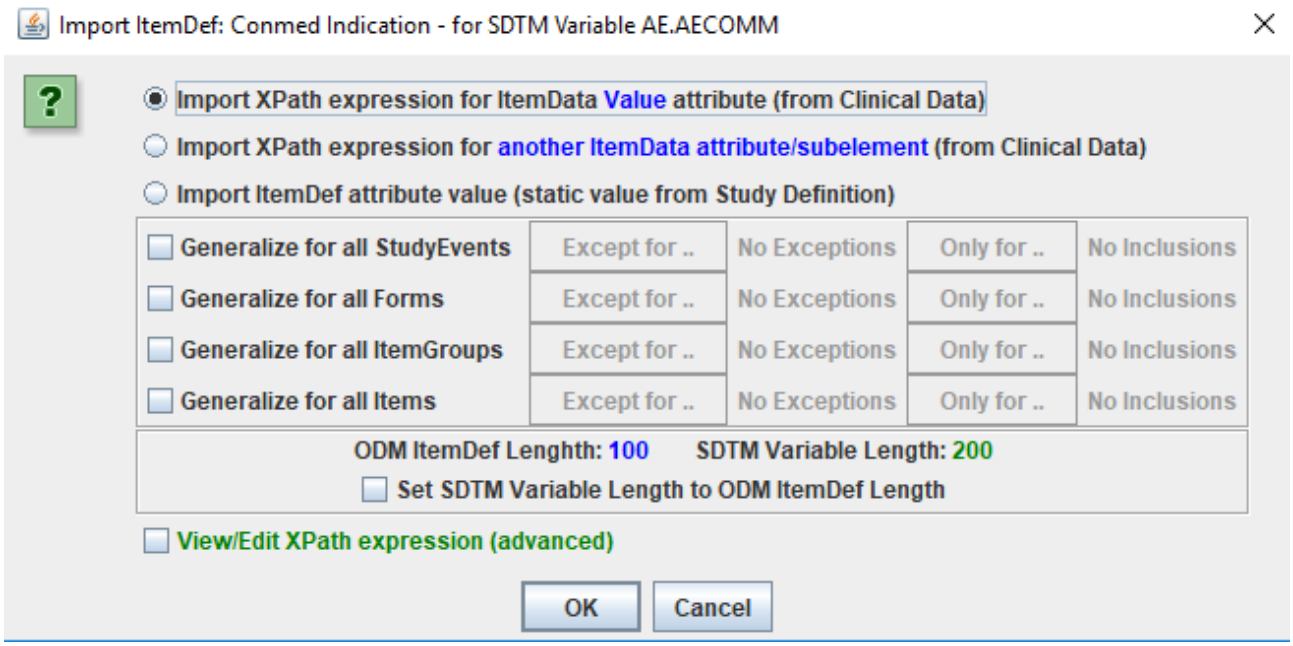
After clicking "OK", we get:



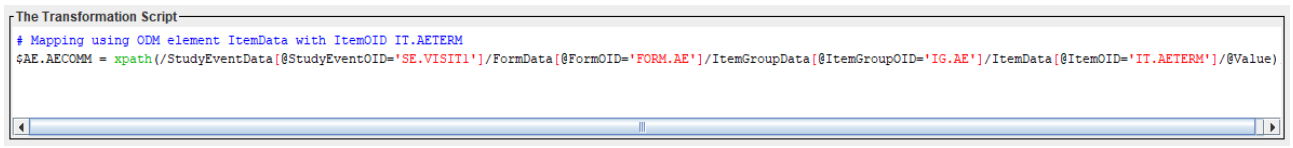
We can now start doing the mapping. As we want to catch all the comments on "AETERM" which come as annotations in the ODM dataset, drag-and-drop "Conmed Indication" (OID=IT.AETERM) from the ODM tree to the SDTM cell "AE.AECOMM":



After dropping to cell "AE.AECOMM" we get:



This leads to the following transformation script:



i.e.: $AE.AECOMM = xpath(/StudyEventData[@StudyEventOID='SE.VISIT1']/FormData[@FormOID='FORM.AE']

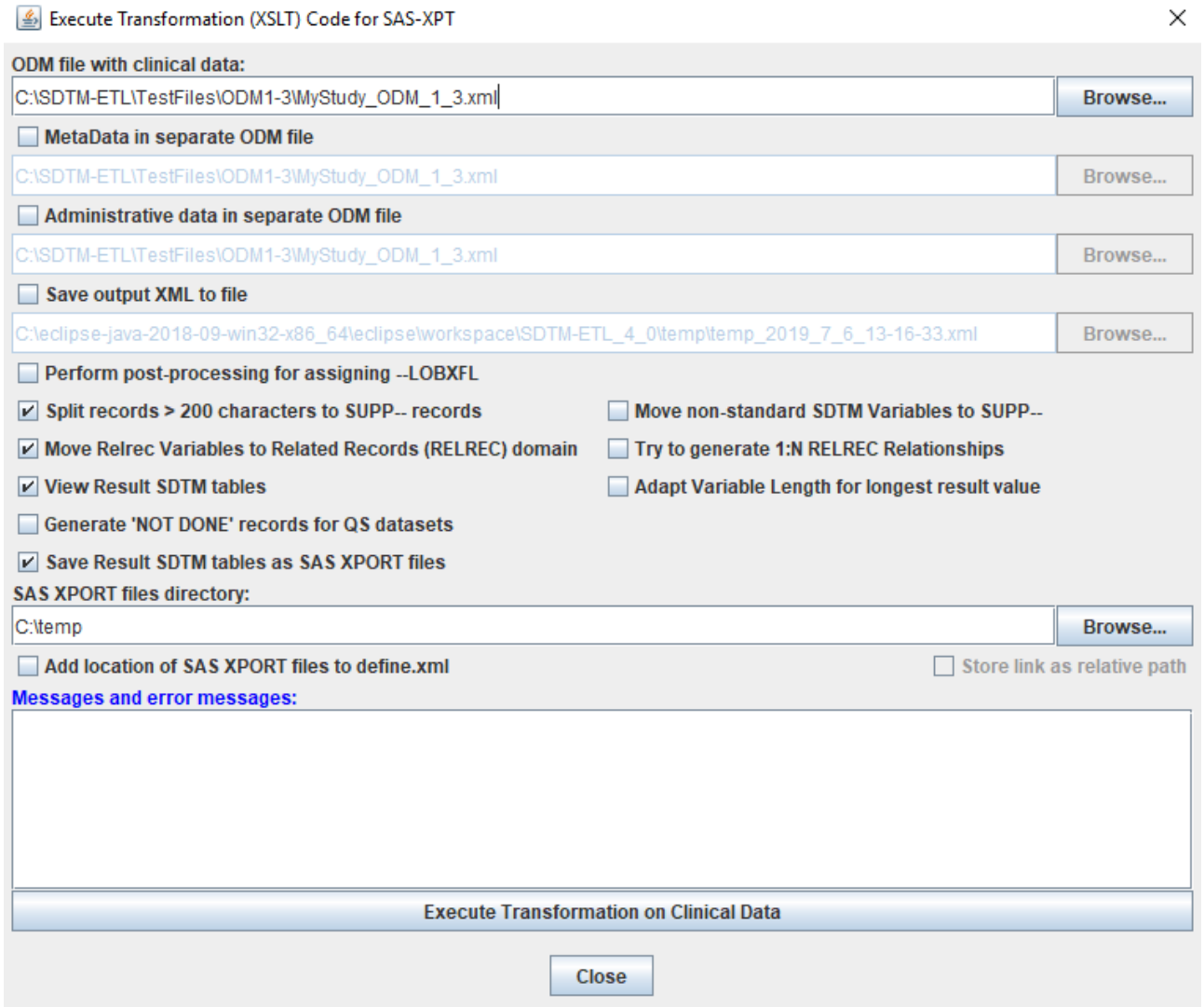
/ItemGroupData[@ItemGroupOID='IG.AE']/ItemData[@ItemOID='IT.AETERM']/@Value);

we do however not want the value of "AETERM", but the annotation/comment on it. So we change the XPath into:

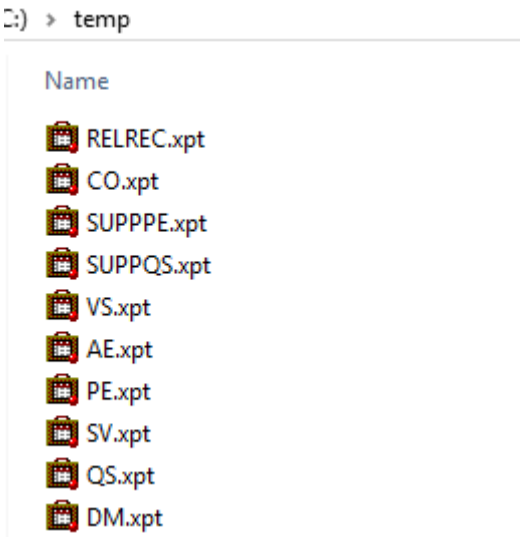
$AE.AECOMM =
xpath(/StudyEventData[@StudyEventOID='SE.VISIT1']/FormData[@FormOID='FORM.AE']
/ItemGroupData[@ItemGroupOID='IG.AE']
/ItemData[@ItemOID='IT.AETERM']/Annotation/Comment);

Don't forget to close the bracket after "/Comment" (from the xpath() function), and to have the semicolon at the end.

Executing the transformation now using the menu "Transform – Generate Transformation (XSLT) Code for SAS-XPT), followed by "Execute Transformation (XSLT) Code" leads to:



To demonstrate that comments longer than 200 characters are no problem, we check the checkbox "Save Result SDTM tables as SAS XPORT files", and add a location where they should be generated.
Clicking "Execute Transformation on Clinical Data" then executes the mappings. The generated SAS XPT files are:



Although we did not explicitly develop a CO dataset, we see that there is one generated anyway, as all "COMMENT" variables are automatically moved to a newly generated CO dataset.

The latter looks like:

CO.xpt

	STUDYID	DOMAIN	RDOMAIN	USUBJID	COSEQ	IDVAR	IDVARVAL	COREF	COVAL	COVAL1	COEV
1	MyStudy	CO	AE	001	1	AESEQ	1		Could be due to the subjec		
2	MyStudy	CO	AE	001	2	AESEQ	3		The quick brown fox jumps	Voyez le brick géant que j'exami	
3	MyStudy	CO	AE	002	1	AESEQ	1		reanimated by investigator		
4	MyStudy	CO	AE	004	1	AESEQ	1		No symptoms observed by su		
5	MyStudy	CO	AE	008	1	AESEQ	1		probably caused by eating		

The second row corresponds to the third AE record for subject 001. The comment was more than 200 characters long, and has correctly been split into a COVAL and COVAL1. This was necessary due to the by the FDA mandated use of the outdated XPT format, which can not cope with variable values of more than 200 characters. So the SDTM team decided to mandate that such values must be split into a COVAL, COVAL1, COVAL2 …

Final remarks:

- This is only a simple example of auto-generating a CO dataset.
- Comments do not necessarily need to come from "annotations" in the ODM dataset. The can as well come from an ODM field that is just a normal data collection point.
- If there are different domains with such comments (for auto-generation), these will all be combined into a single CO dataset.
- As for all mappings, the value for the comment can also be derived. For example, we did a simple exercise that when VSORRES is larger than 200 and VSORRESU is 'lb' (pounds), a comment is generated: "Subject is overweight, BMI = xx.y". However, in the case of a regulatory submission, only comments that were really collected should be included
- In the "autogeneration" mode, currently only one CO record per SDTM record is possible (one XXCOMM variable).