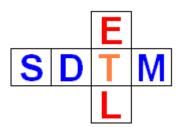
# SDTM-ETL 3.2 User Manual and Tutorial

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 studyspecific 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 When doing so, don't care about the 200 character limitation of COVAL (a relict of the outdated SAS-XPT format), as this will be care of automatically by the software when generating the SAS datasets ("automated splitting").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.:



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"<sup>1</sup>

<sup>&</sup>lt;sup>1</sup> In the case of SEND, it will of course be "New SEND Variable for COMMENT"

Insert Transform Validate Options About							
MeasurementUnit definitions from ODM into define.xml							
All CodeList definitions from ODM into define.xml							
Selected CodeList definitions from ODM into define.xml							
CodeList definitions from File into define.xml							
Create new SDTM CodeList from existing CodeList							
Create new SDTM CodeList from MeasurementUnits							
Create new ValueList from existing CodeList							
Create mapping formula Ctrl-M							
Sponsor defined SDTM Domain Ctrl-P							
Domain-specific SUPPQUAL Ctrl-Q							
Associated Persons Domain							
Global Subject Variables Domain							
New SDTM Variable Ctrl-I							
New non-standard SDTM Variable for SUPPQUAL							
New SDTM Variable for COMMENT							
New SDTM Variable for RELREC							
Link to Annotated CRF							
Link to Supplemental Doc							
CRF Page Numbers to Variable Origin							

#### This will lead to:

Add new	Variable (for COMMENTS) to dor	nain AE X
(i)	New Variable for COMMENT: A	E.AE COMM
	Data type:	text
	Length:	200
	Origin:	
	Role:	COMMENT
	Comment:	
	Description:	Comments for domain AE
		Validate
	ОК	Cancel

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:

Me	essage				×
(	i) Appende	d Variable (for CO	MMENTS)AE.AECOM	/IM at position 41 f	or domain AE
			ОК		
	AE.AEDUR	AE.AEENRF	AE.AECOMM		

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":

– 🗋 ItemDef : Line Number
— Discrete Indication
- 🗋 ItemDef : Start Month - Enter Two Digits 01-12
- 🗋 ItemDef : Start 🛛 OID: IT.AETERM - Name: Conmed Indication
- 🗋 ItemDef : Start V SDTM Variable Name: AETERM

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

 $\times$ Import ItemDef: Conmed Indication - for SDTM Variable AE.AECOMM ? Import XPath expression for ItemData Value attribute (from Clinical Data) Import XPath expression for another ItemData attribute/subelement (from Clinical Data) Import ItemDef attribute value (static value from Study Definition) Generalize for all StudyEvents Except for .. No Exceptions Only for .. No Inclusions Generalize for all Forms Except for .. No Exceptions Only for .. No Inclusions Generalize for all ItemGroups Except for .. No Exceptions Only for .. No Inclusions Generalize for all Items Except for ... No Exceptions Only for .. No Inclusions ODM ItemDef Lenghth: 100 SDTM Variable Length: 200 Set SDTM Variable Length to ODM ItemDef Length View/Edit XPath expression (advanced) OK Cancel

#### This leads to the following transformation script:

The Transformation Script
# Mapping using ODM element ItemData with ItemOID IT.AETERM
sAE.AECOMM = xpath(/StudyEventData[@StudyEventOID='SE.VISIT1']/FormData[@FormOID='FORM.AE']/ItemGroupData[@ItemGroupOID='IG.AE']/ItemData[@ItemGID='IG.AE']/ItemGroupData[@ItemGID='IG.AE']/ItemGroupData[@ItemGID='IG.AE']/ItemGroupData[@ItemGID='IG.AE']/ItemGroupData[@ItemGID='IG.AE']/ItemGroupData[@ItemGID='IG.AE']/ItemGroupData[@ItemGID='IG.AE']/ItemGroupData[@ItemGID='IG.AE']/ItemGroupData[@ItemGID='IG.AE']/ItemGroupData[@ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']/ItemGID='IG.AE']//ItemGID='IG.AE']//ItemGID='IG.AE']//ItemGID='IG.AE']//ItemGID='IG.AE']//ItemGID='IG.AE']//ItemGID='IG.AE']//ItemGID='IG.AE']//ItemGID='IG.AE']//ItemGID='IG.AE']//ItemGID='IG.AE']//ItemGID='IG.AE']//ItemGID='IG.AE']//ItemGID='IG.AE']//ItemGID='IG.AE']//ItemGID='IG.AE']//ItemGID='IG.AE']//ItemGID='IG.AE']//ItemGID='IG.AE']//ItemGID='IG.AE']//ItemGID='IG.AE']//I

### 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);

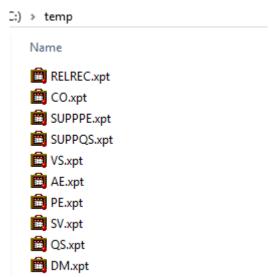
Don't forget to close the bracket after "/Comment" (from the xpath() function), and to have the semicolon at the end.

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

ODM file with clinical data:       Browse         C:SDTM-ETL\TestFiles\ODM1-3MyStudy_ODM_1_3.xml       Browse         MetaData in separate ODM file       Browse         C:SOTM-ETL\TestFiles\ODM1-3MyStudy_ODM_1_3.xml       Browse         Administrative data in separate ODM file       Browse         C:SOTM-ETL\TestFiles\ODM1-3MyStudy_ODM_1_3.xml       Browse         Save output XML to file       Browse         C:dclipse-java-2018-09-win32-x86_64teclipse\workspace\SDTM-ETL_4_0ttemptemp_2019_7_6_13-16-33.xml       Browse         Perform post-processing for assigningLOBXFL       Split records > 200 characters to SUPP records       Move non-standard SDTM Variables to SUPP         V New Relrec Variables to Related Records (RELREC) domain       Try to generate 1:N RELREC Relationships       View Result SDTM tables       Adapt Variable Length for longest result value         Generate 'NOT DONE' records for QS datasets       Save Result SDTM tables as SAS XPORT files       SAS XPORT files define.xml       Browse         Add location of SAS XPORT files to define.xml       Store link as relative path         Messages and error messages:       Execute Transformation on Clinical Data       Close	Execute Transformation (XSLT) Code for SAS-XPT		×
MetaData in separate ODM file C:\SDTM-ETLYTestFiles\ODM1-3My\Study_ODM_1_3.xml Browse Administrative data in separate ODM file C:\SDTM-ETLYTestFiles\ODM1-3My\Study_ODM_1_3.xml Browse Save output XML to file C:\sectips=Java-2018-09-win32-x86_64eelipse\workspace\SDTM-ETL_4_0temptemp_2019_7_6_13-16-33.xml Browse Perform post-processing for assigningLOBXFL Solution of solution of solution of solution of solution of solution of SAS XPORT files to define.xml Messages and error messages: Execute Transformation on Clinical Data	ODM file with clinical data:		
C:SDTM-ETL/TestFiles/ODM1-3MyStudy_ODM_1_3.xml       Browse         Administrative data in separate ODM file       Erowse         C:SDTM-ETL/TestFiles/ODM1-3MyStudy_ODM_1_3.xml       Browse         Save output XML to file       Erowse         C:teclipse-Java-2018-09-win32-x86_64teclipse/Java-2019_7_6_13-16-33.xml       Browse         Perform post-processing for assigning -LOBXFL       Split records > 200 characters to SUPP records       Move non-standard SDTM Variables to SUPP         V Nove Refrec Variables to Related Records (RELREC) domain       Try to generate 1:N RELREC Relationships       View Result SDTM tables         Generate 'NOT DONE' records for QS datasets       Adapt Variable Length for longest result value       Generate 'NOT DONE' records for QS datasets         SAS XPORT files directory:       C:temp       Browse         Add location of SAS XPORT files to define.xml       Store link as relative path         Messages and error messages:       Execute Transformation on Clinical Data	C:\SDTM-ETL\TestFiles\ODM1-3\MyStudy_ODM_1_3.xml		Browse
Administrative data in separate ODM file C:\SDTM-ETL\TestFiles\ODM1-3MyStudy_ODM_1_3.xml Browse Save output XML to file C:\eclipse-java-2018-09-win32-x86_64\eclipse\workspace\SDTM-ETL_4_0\temptemp_2019_7_6_13-16-33.xml Perform post-processing for assigning -LOBXFL Split records > 200 characters to SUPP records Solve Relrec Variables to Related Records (RELREC) domain View Result SDTM tables Adapt Variable Length for longest result value Generate 'NOT DONE' records for QS datasets SAS XPORT files directory: C:ttemp Browse Browse Browse Ctemp Execute Transformation on Clinical Data	MetaData in separate ODM file		
C:\SDTM-ETL\TestFiles\ODM1-3\MyStudy_ODM_1_3.xml       Browse         Save output XML to file       C:\ecilpse-java-2018-09-win32-x86_64\ecilpse\workspace\SDTM-ETL_4_0templtemp_2019_7_6_13-16-33.xml       Browse         Perform post-processing for assigning -LOBXFL       Split records > 200 characters to SUPP records       Move non-standard SDTM Variables to SUPP         Move Relrec Variables to Related Records (RELREC) domain       Try to generate 1:N RELREC Relationships       View Result SDTM tables       Adapt Variable Length for longest result value         Generate 'NOT DONE' records for QS datasets       Save Result SDTM tables as SAS XPORT files       SAX XPORT files directory:         C:temp       Browse       Browse       Browse         Add location of SAS XPORT files to define.xml       Store link as relative path         Messages and error messages:       Execute Transformation on Clinical Data	C:\SDTM-ETL\TestFiles\ODM1-3\MyStudy_ODM_1_3.xml		Browse
Save output XML to file         C:leclipse-java-2018-09-win32-x86_64/eclipse/workspace/SDTM-ETL_4_0/temp!temp_2019_7_6_13-16-33.xml       Browse         Perform post-processing for assigningLOBXFL       Move non-standard SDTM Variables to SUPP         Move Relrec Variables to Related Records (RELREC) domain       Try to generate 1:N RELREC Relationships         View Result SDTM tables       Adapt Variable Length for longest result value         Generate 'NOT DONE' records for QS datasets       Save Result SDTM tables as SAS XPORT files         SAS XPORT files directory:       C:temp         Add location of SAS XPORT files to define.xml       Store link as relative path         Messages and error messages:       Execute Transformation on Clinical Data	Administrative data in separate ODM file		
C:eclipse-java-2018-09-win32-x86_64/eclipse/workspace/SDTM-ETL_4_0/templtemp_2019_7_6_13-16-33.xml       Browse         Perform post-processing for assigning -LOBXFL       Split records > 200 characters to SUPP records       Move non-standard SDTM Variables to SUPP         Move Relrec Variables to Related Records (RELREC) domain       Try to generate 1:N RELREC Relationships         View Result SDTM tables       Adapt Variable Length for longest result value         Generate 'NOT DONE' records for QS datasets       Save Result SDTM tables as SAS XPORT files         SAS XPORT files directory:       C:temp       Browse         Add location of SAS XPORT files to define.xml       Store link as relative path         Messages and error messages:       Execute Transformation on Clinical Data	C:\SDTM-ETL\TestFiles\ODM1-3\MyStudy_ODM_1_3.xml		Browse
Perform post-processing for assigning -LOBXFL         ✓ Split records > 200 characters to SUPP records       Move non-standard SDTM Variables to SUPP         ✓ Move Relrec Variables to Related Records (RELREC) domain       Try to generate 1:N RELREC Relationships         ✓ View Result SDTM tables       Adapt Variable Length for longest result value         Generate 'NOT DONE' records for QS datasets       Adapt Variable Length for longest result value         Save Result SDTM tables as SAS XPORT files       SAS XPORT files directory:         C:ttemp       Browse         Add location of SAS XPORT files to define.xml       Store link as relative path         Messages and error messages:       Execute Transformation on Clinical Data	Save output XML to file		
✓ Split records > 200 characters to SUPP records       Move non-standard SDTM Variables to SUPP         ✓ Move Relrec Variables to Related Records (RELREC) domain       Try to generate 1:N RELREC Relationships         ✓ View Result SDTM tables       Adapt Variable Length for longest result value         Generate 'NOT DONE' records for QS datasets       Adapt Variable Length for longest result value         Generate 'NOT DONE' records for QS datasets       Browse         ✓ Save Result SDTM tables as SAS XPORT files       Browse         Add location of SAS XPORT files to define.xml       Browse         Messages and error messages:       Execute Transformation on Clinical Data	C:\eclipse-java-2018-09-win32-x86_64\eclipse\workspace\SDTM-E	TL_4_0\temp\temp_2019_7_6_13-16-33.xml	Browse
<ul> <li>✓ Move Relrec Variables to Related Records (RELREC) domain</li> <li>○ Try to generate 1:N RELREC Relationships</li> <li>○ View Result SDTM tables</li> <li>○ Generate 'NOT DONE' records for QS datasets</li> <li>○ Save Result SDTM tables as SAS XPORT files</li> <li>SAS XPORT files directory:</li> <li>C:ttemp</li> <li>○ Add location of SAS XPORT files to define.xml</li> <li>○ Store link as relative path</li> <li>Messages and error messages:</li> </ul>	Perform post-processing for assigningLOBXFL		
✓ View Result SDTM tables       □ Adapt Variable Length for longest result value         □ Generate 'NOT DONE' records for QS datasets       ✓         ✓ Save Result SDTM tables as SAS XPORT files       SAS XPORT files directory:         C:\temp       Browse         □ Add location of SAS XPORT files to define.xml       □ Store link as relative path         Messages and error messages:       □         Execute Transformation on Clinical Data       □	Split records > 200 characters to SUPP records	Move non-standard SDTM Variables to SUPP	
Generate 'NOT DONE' records for QS datasets         ✓ Save Result SDTM tables as SAS XPORT files         SAS XPORT files directory:         C:\temp         Browse         Add location of SAS XPORT files to define.xml         Messages and error messages:         Execute Transformation on Clinical Data	$\checkmark$ Move Relrec Variables to Related Records (RELREC) domain	Try to generate 1:N RELREC Relationships	
✓ Save Result SDTM tables as SAS XPORT files         SAS XPORT files directory:         C:\temp         Add location of SAS XPORT files to define.xml         Messages and error messages:         Execute Transformation on Clinical Data	View Result SDTM tables	Adapt Variable Length for longest result value	
SAS XPORT files directory:   C:\temp   Add location of SAS XPORT files to define.xml   Messages and error messages:     Image: Comparison of Clinical Data	Generate 'NOT DONE' records for QS datasets		
C:\temp Browse Add location of SAS XPORT files to define.xml Store link as relative path Messages and error messages: Execute Transformation on Clinical Data			
Add location of SAS XPORT files to define.xml  Messages and error messages:  Execute Transformation on Clinical Data	-		
Messages and error messages:  Execute Transformation on Clinical Data			
Execute Transformation on Clinical Data		Store link a	as relative path
	messages and error messages.		
Close	Execute Transform	ation on Clinical Data	
	CI	ose	

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:

	STUDYID	DOMAIN	RDOMAIN	USUBJID	COSEQ	IDVAR	IDVARVAL	COREF	COVAL	COVAL1	COE
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 COVAL1, 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).