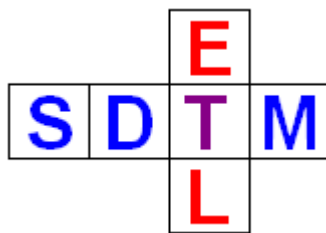


# SDTM-ETL™



## New features in version 1.6

Author: Jozef Aerts – XML4Pharma  
July 2011

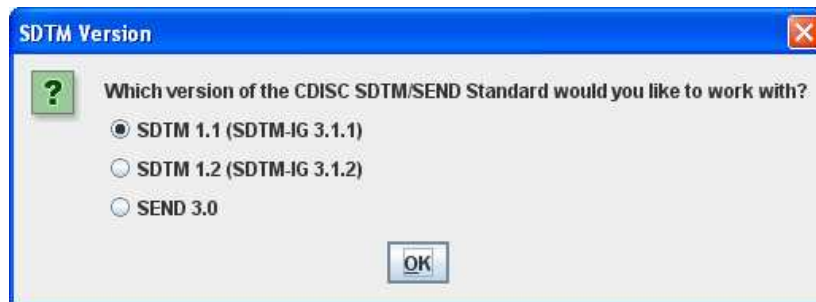
## Table of Contents

Implementation of SEND v.3.0 final .....	3
Automated creation of the RELREC dataset and records.....	4
Subject Global Variables.....	8
Searching function in scripting editor.....	10
Latest SDTM, SEND, ADaM and CDASH controlled terminology.....	11
Better compliance with OpenCDISC validation.....	11
Sticky notes for SDTM variables.....	12
“Required” / “Expected” / “Permissible” shown in “CDISC Notes”.....	13
Tailoring the log level per XML file.....	13
Changes in the XSLT that is generated.....	14
Faster drag-and-drop.....	14
Offline execution of existing mappings.....	14
SDTM-ETL Light.....	14

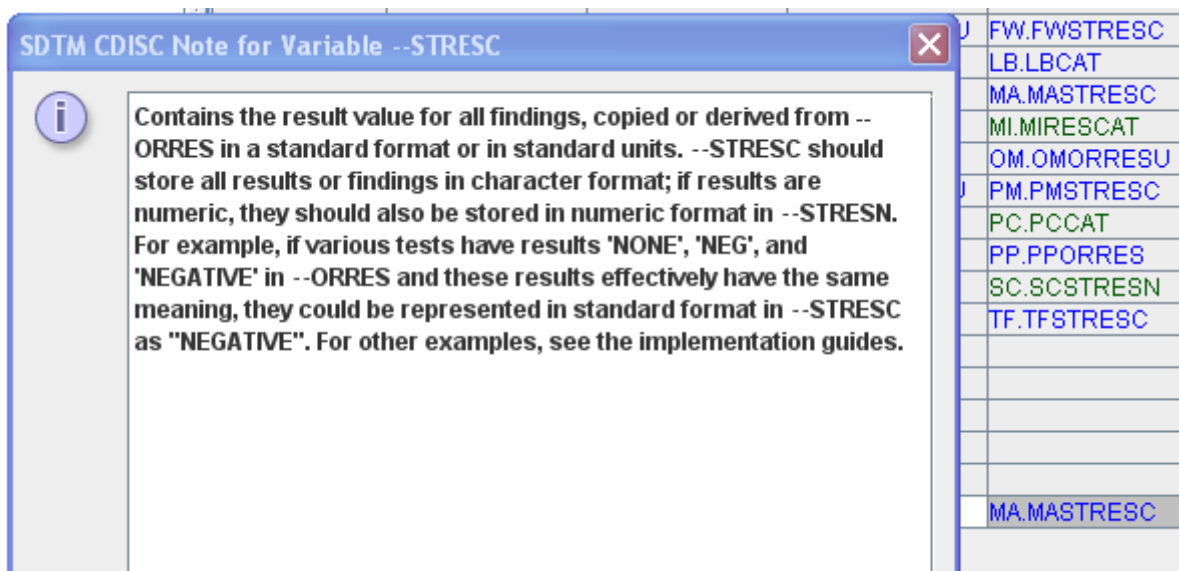
## Implementation of SEND v.3.0 final

The final version of the SEND standard v.3.0 has now been implemented.

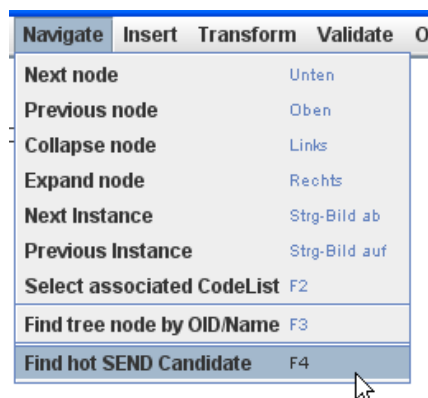
When starting on a new set of mappings, the user has now the choice between SDTM v.1.1 (SDTM-IG 3.1.1), SDTM 1.2 (SDTM-IG 3.1.2) and SEND v.3.0:



When SEND 3.0 is chosen, the define.xml template for SEND 3.0 is loaded, including all the necessary information from the SEND 3.0 draft standard, such as “CDISC notes”:



Everything else works just like when working with one of the SDTM standards, except that the titles of some menus and dialogs are adapted. For example:



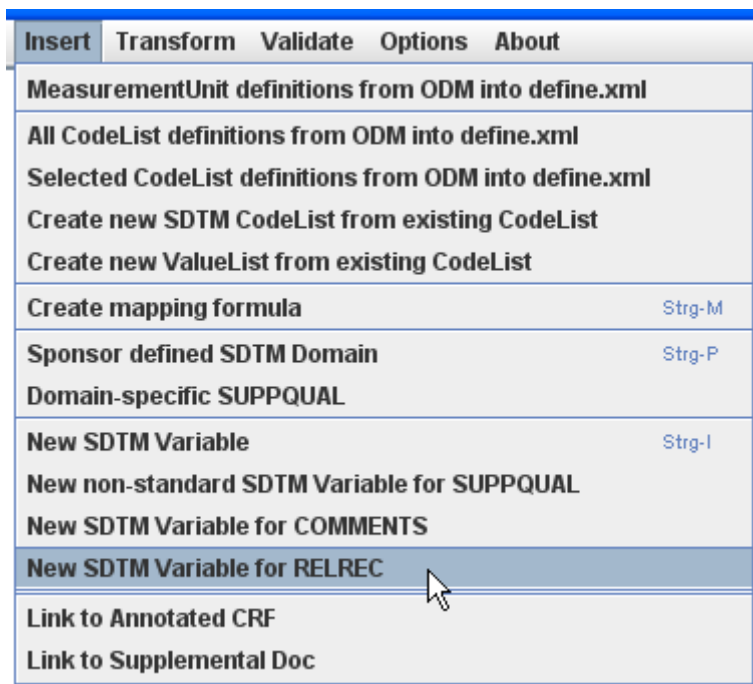
## Automated creation of the RELREC dataset and records

The SDTM has a very weird way of defining records that are related between datasets. Very probably, this is due to the limitations of the SAS Transport 5 format.

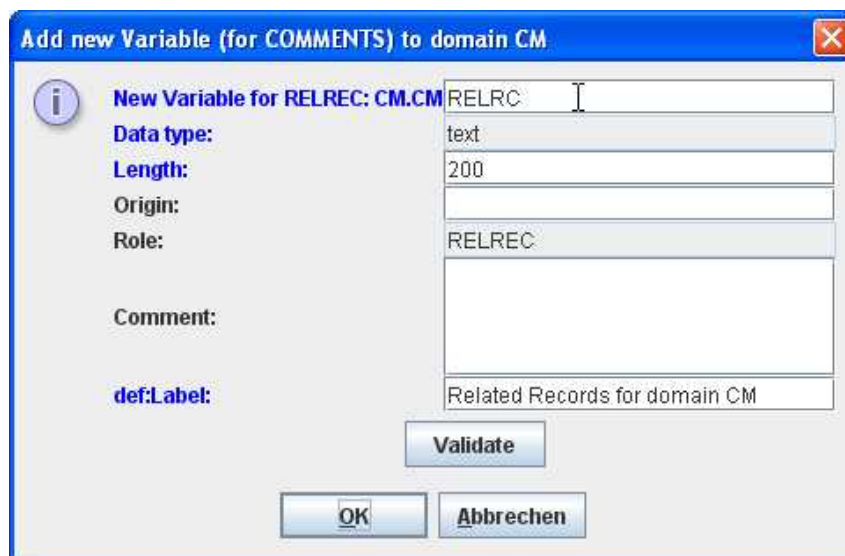
In XML, such a relation could easily be defined by a simple XPath expression.

Version 1.6 of SDTM-ETL does now also allow to automatically create the RELREC dataset and records from mappings.

In order to do so, create a special variable, using the menu “Insert -> SDTM Variable for RELREC”:



A dialog then shows up:



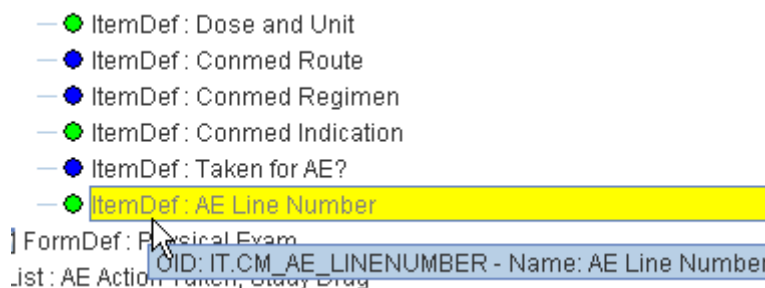
allowing to set a name, and value for the “Length”, “Origin”, “def:Label” and a comment for this special variable. Remark that the “Role” has been fixed to “RELREC” which will later enable to split of this variable and create a separate RELREC dataset.

After clicking OK, the new variable is added to the table (at the end of the row), and colored red, to indicate it is a non-standard variable:

AE.AESTDTC	AE.AEENDTC	AE.AESTDY	AE.AEENDY
CM.CMSTRTPT	CM.CMSTTPT	CM.CMENRTPT	CM.CMRELRC

This special variable will usually be created at the “end point” or “target” of the relationship. For example, if there are one or more concomitant medications for each adverse event, we will create the special variable in the CM domain, and then provide a mapping that points to the AE record.

In our case, the concomitant medication is related to the AE line number, so we can drag and drop from that in the ODM tree structure from the CM form:



which results in the following code being generated automatically:

```

The Transformation Script
# Mapping using ODM element ItemData with ItemOID IT.CM_AE_LINENUMBER
$TEMP = xpath(/StudyEventData[@StudyEventOID='SE.VISIT1']/FormData[@FormOID='FORM.CONMED']/ItemGroupData[
]

The Transformation Script
.'FORM.CONMED']/ItemGroupData[@ItemGroupOID='IG.CONMED']/ItemData[@ItemOID='IT.CM_AE_LINENUMBER']/@Value);

```

(we changed the variable name into \$TEMP for further elaboration).

We can now use the new, special “relrec” function to define the relationship. Therefore we add:

```
$CM.CMRELRC = relrec('AE', 'AESPID', $TEMP);
```

meaning: the variable CM.CMRELRC value takes the value of \$TEMP (i.e. the reference to the AE line number) as related record to AESPID in related domain AE, and generates a related record from that. The parameters for these new function are (in SDTM jargon):

```
relrec($RDOMAIN, $IDVAR, $IDVARVAL)
```

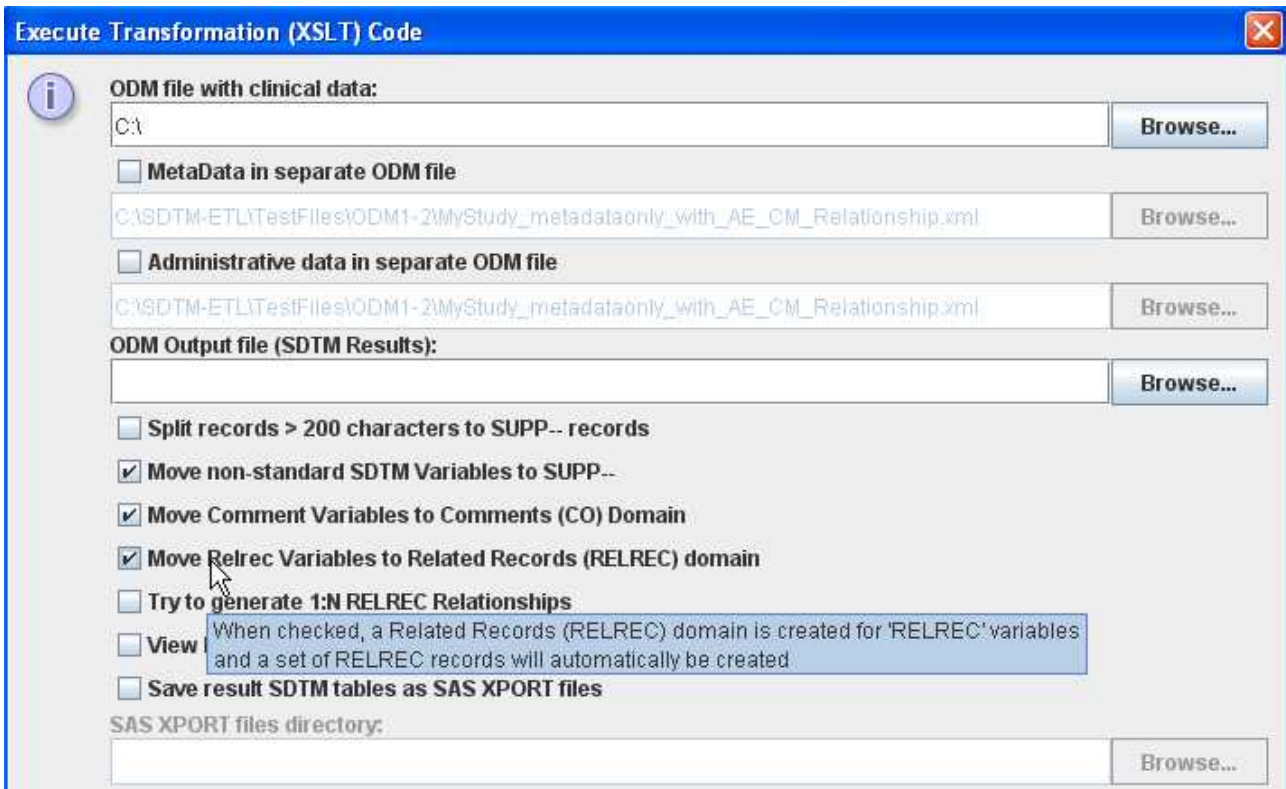
This is correct, as the same XPath expression was used to generate AESPID in the AE domain.

Please remark that it is not allowed to use 'AESEQ' as the second parameter of the function, as the sequence number is only assigned in the last step of the transformation, so we do not know yet what its value will be.

So the complete mapping is:

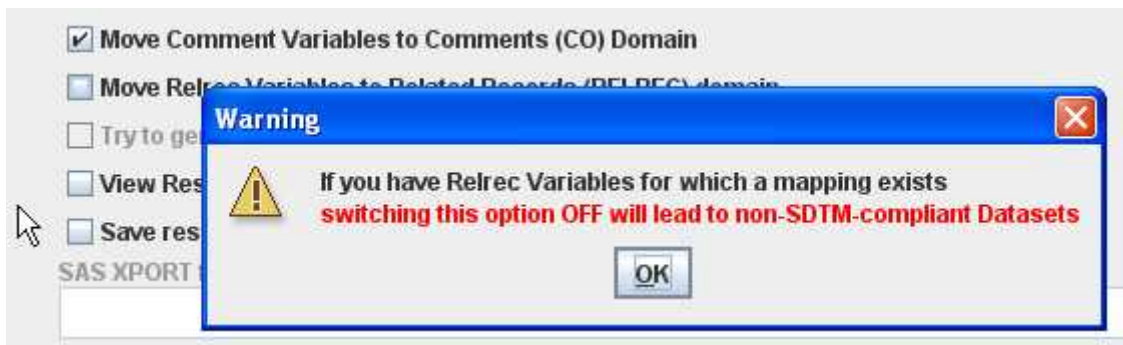
```
The Transformation Script
# Mapping using ODM element ItemData with ItemOID IT.CM_AE_LINENUMBER
$TEMP = xpath(/StudyEventData[@StudyEventOID='SE.VISIT1']/FormData[@FormOID='FORM.CONMED']/ItemGroupData[
$CM.CMRELRC = relrec('AE','AESPID',$TEMP);
```

We can now execute the mapping on the clinical data, using the menu “Transform -> Generate Transformation (XSLT) Code”, followed by “Execute Transformation (XSLT) Code”:



We see that there is an additional checkbox “Move Relrec Variables to Related Records (RELREC) domain. By default it is checked on.

When checked off, a warning message will appear:



saying that this will probably lead to non-compliant SDTM datasets.

A second checkbox can be found immediately after it:



When checked, the software will try to generate 1:N RELREC relationships. When unchecked, all relationships will be of the 1:1 type, i.e. there will be two rows (with the same RELID) for each combination of concomitant medication and AE line number.

In most cases, one will however want to generate 1:N relationships. In our example this means that there will be N+1 rows with the same RELID in the RELREC dataset, the first being the AE line number, followed by a row for each concomitant medication that is related to that AE line number.

When now executing the mapping, the result will, for the 1:N case, be:

SDTM Tables							
MyStudy:AE MyStudy:CM MyStudy:RELREC							
STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	RELTYPE	RELID	
MyStudy	AE	001	AESPID	3		1	
MyStudy	CM	001	CMSEQ	1		1	
MyStudy	CM	001	CMSEQ	2		1	
MyStudy	AE	008	AESPID	1		2	
MyStudy	CM	008	CMSEQ	1		2	
MyStudy	CM	008	CMSEQ	2		2	
MyStudy	CM	008	CMSEQ	3		2	

saying that the for subject 001, the AE record with AESPID=3, is related to the CM records with CMSEQ=1 and CMSEQ=2 (same RELID).

Similarly, for subject 008, the AE record with AESPID=1, is related to the three CM records with CMSEQ=1, 2 and 3.

In case we choose for 1:1 relationships, we get:

SDTM Tables							
MyStudy:AE		MyStudy:CM		MyStudy:RELREC			
STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	RELTYPE	RELID	
MyStudy	CM	001	CMSEQ	1		1	
MyStudy	AE	001	AESPID	3		1	
MyStudy	CM	001	CMSEQ	2		2	
MyStudy	AE	001	AESPID	3		2	
MyStudy	CM	008	CMSEQ	1		3	
MyStudy	AE	008	AESPID	1		3	
MyStudy	CM	008	CMSEQ	2		4	
MyStudy	AE	008	AESPID	1		4	
MyStudy	CM	008	CMSEQ	3		5	
MyStudy	AE	008	AESPID	1		5	

Showing all the 1:1 pairs.

The automatically generated SAS XPT file “RELREC.xpt” for the 1:N case then looks like:

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

The distribution contains a sample define.xml file “MyStudy\_AE\_CM\_Domains\_define\_with\_RELREC.xml” which can be used in combination with the “MyStudy\_metadataonly\_with\_AE\_CM\_Relationship.xml” file (ODM metadata) and the file “MyStudy\_ClinicalData\_only\_with\_AE\_CM\_Relationship.xml” (ODM clinical data).

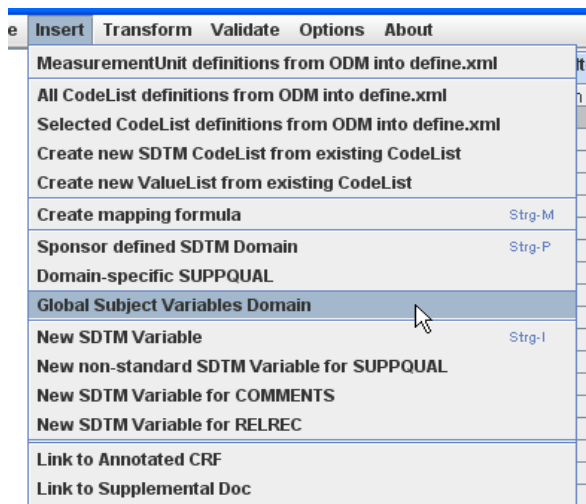
### **Subject Global Variables**

Some variables are used in different domains for calculations. Typical examples are the “Reference Start Date” (DM.RFSTDTC) and the “Reference End Date” (DM.RFENDTC). They are e.g. used for the calculation of durations (e.g. AE), visit days (VISITDY) and the “End Relative to Reference Period” (--ENRF) variables.

Until v.1.5, the mapping for variables such as RFSTDTC and RFENDTC need to be generated over again, or copied-and-pasted from the DM domain.

In version 1.6, it is possible to define “subject global variables” and provide mappings for them, and then use these variables in any other domain. The naming “subject global” has been chosen to reflect that such a variable is global within the scope of a subject, as e.g. the reference start date can be different for each subject.

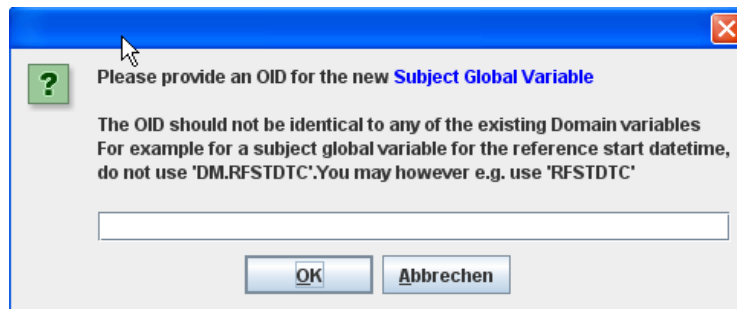
In order to create “subject global variables”, use the menu “Insert -> Global Subject Variables Domain”.



RELREC	STUDYID	RDOMAIN
SUPPQUAL	STUDYID	RDOMAIN
MyStudy:GLOBAL		
MyStudy:DM	STUDYID	DOMAIN
MyStudy:QS	STUDYID	DOMAIN
MyStudy:SV	STUDYID	DOMAIN
MyStudy:PE	STUDYID	DOMAIN

This creates an empty domain with the name “GLOBAL”. This special domain cannot be copied (only one instance is allowed), and should come as the first domain in the list of study-specific domains (the software takes care of that). It can also not be moved.

One can then define a set of subject global variables within this “special” domain, using the menu “Insert -> New SDTM Variable” or simply by a double-click in an empty cell,



and use drag-and-drop or the variable editor to generate the mappings. It is important that such “global” variables are not given an OID that is already used in any of the other domains. For example, it is not a good idea to use “DM.RFSTDTC”, “DM.RFENDTC”, or “TEMP” as OID for such a variable. Instead one can however e.g. use “RFSTDTC” and “ENDTC” or “GLOBAL.RFSTDTC” and “GLOBAL.RFENDTC”.

RELREC	STUDYID	RDOMAIN	USUBJID	IDV#
SUPPQUAL	STUDYID	RDOMAIN	USUBJID	IDV#
MyStudy:GLOBAL	RFSTDTC	RFENDTC		
MyStudy:DM	STUDYID	DOMAIN	USUBJID	SUE
MyStudy:QS	STUDYID	DOMAIN	RFENDTC	QS.#
MyStudy:SV	STUDYID	DOMAIN	Mandatory: No	SV.\
MyStudy:PE	STUDYID	DOMAIN	OrderNumber: 2	PE.F
MyStudy:AE	STUDYID	DOMAIN	Role: GLOBAL	AE.#
MyStudy:VS	STUDYID	DOMAIN		VS.V

The “subject global” variables can then be referenced in any other domain mapping. For example for the variable VS.VSDY:

```

The Transformation Script
$VS.VSDY = datediff($VS.VSDTC,$RFSTDTC) + 1;

```

The “subject global” \$RFSTDTC variable is used in the calculation for the date difference between the VS “date/time of collection” (VS.VSDTC - which was already defined in the same domain) and the “reference start date/time”. One day has to be added as SDTM defines the first day of the study as day “1” (instead of day “0”).

Remark that local testing on real clinical data in mappings where “subject global” variables are used is not possible, as the variable is not “in scope”.

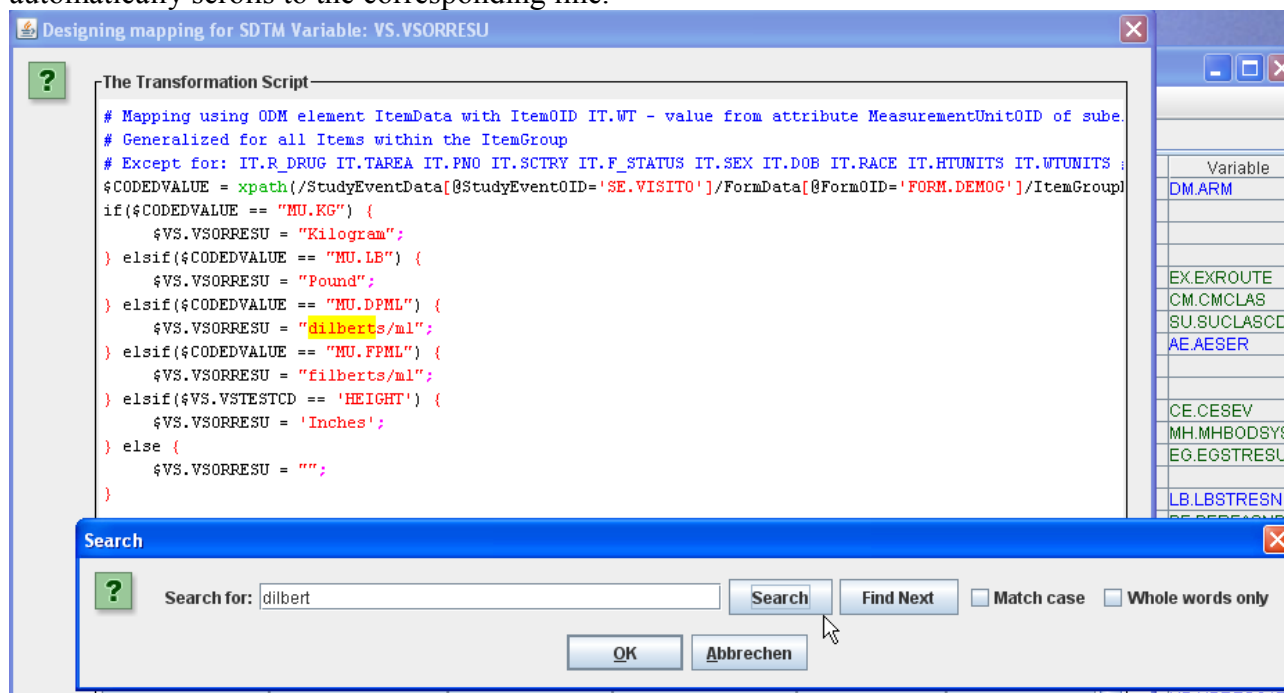
The “Subject Global” domain is removed when doing a “cleaned” export of the define.xml, and is omitted when validating the underlying define.xml structure using OpenCDISC or using the SDTM-ETL's own validator.

The distribution contains a sample file “MyStudy\_AE\_Domain\_with\_GLOBAL\_define.xml” demonstrating this new feature.

### **Searching function in scripting editor**

A searching function has been introduced in the scripting editor, which can be invoked using the keystroke combination CTRL-F. A separate dialog then appears:

When the “search” button is then clicked, the hit is highlighted in the editor, and the system automatically scrolls to the corresponding line.



### **Latest SDTM, SEND, ADaM and CDASH controlled terminology**

Version 1.6 comes with the latest versions of CDISC controlled terminology for SDTM, SEND, ADaM and CDASH. This is important, as it is clear now that the FDA expects more and more that CDISC controlled terminology is used for electronic submissions.

Remark that this new controlled terminology has not been implemented in the template for the old SDTM 1.1 (SDTM-IG 3.1.1) as the latter only has a very limited set of controlled terminology.

The new codelists can easily be recognized as they have an NCI number in the OID, e.g. "CL.C66781.AGEU".

One major difference with the old template files for controlled terminology is that the empty string is not regarded anymore as an allowed coded value. Earlier versions of the CDISC controlled terminology still stated the empty string as a possible coded value.

### **Better compliance with OpenCDISC validation**

Although the validation rules of OpenCDISC were not developed by, nor in cooperation with CDISC and the FDA<sup>1</sup>, we see that the latter is (stunningly) accepting these validation rules as "being the truth". By using the latest controlled terminology (see previous section), the number of errors and warning will however be reduced considerably.

Also, one must treat the OpenCDISC warnings for what they are: warnings. Many people incorrectly believe that they will not be able to do a successful FDA submission for datasets that generate warnings by OpenCDISC validation software.

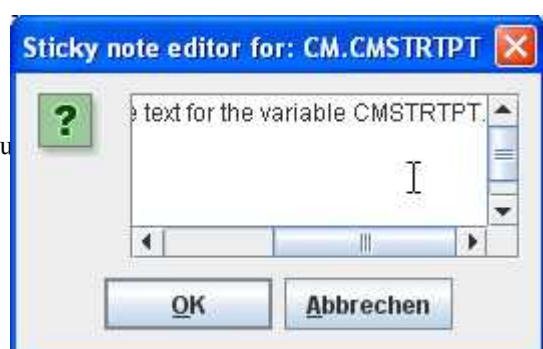
It is however a very good idea to document warnings thrown by OpenCDISC in the "reviewers guide" explaining why the warning was generated (i.e. because a non-CDISC codelist was used in the study, or because the rule itself is incorrect), and discuss these with the FDA.

### **Sticky notes for SDTM variables**

It is now possible to add "sticky notes" to SDTM/SEND variables (not to SDTM/SEND domains). This is especially useful e.g. for marking SDTM variables for which the mapping still needs to be completed.

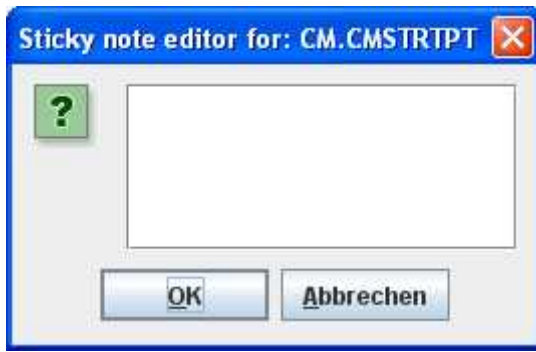
In order to add a sticky note to an SDTM variable, use the menu "Edit -> Sticky Note on SDTM Variable" after having selected an SDTM variable from a study-specific instance domain (so not from the template).

A small text field will then appear allowing to add some text to the sticky note:



---

<sup>1</sup> Some of the rules in OpenCDISC are even inherently wrong, see the following statement.



The number of characters and lines is in principle unlimited.

When an SDTM has a sticky note attached, it is marked by a “pin” in the table, e.g.

GR	AE.AESTDTC	AE
IRF	 CM.CMSTRPT	CM

This “pin” icon can however also be hidden by using the menu “Options -> Settings” and checking the checkbox “Hide sticky notes in SDTM/SEND cells”:

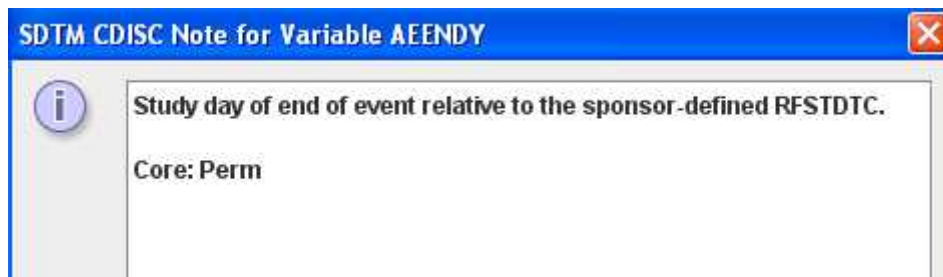


In order to completely remove a sticky note from an SDTM variable, use the menu “Edit -> Remove Sticky Note from SDTM Variable”.

Please remark that all sticky notes are automatically removed when generated a “cleaned “ define.xml. They are however retained when saving the project define.xml file for later use or reuse.

**“Required” / “Expected” / “Permissible” shown in “CDISC Notes”**

When using the menu “View -> CDISC Notes”, it is now also shown whether the SDTM variable is “required”, “expected” or “permissible”. For example:



### **Tailoring the log level per XML file**

The file “properties.dat” that is read by the software at startup, allows to set the logging level of the logging system. Allowed values so far were: “INFO”, “DEBUG”, “WARN”, “ERROR”, and “FATAL”.

Most usual, the log level will be set to “INFO”, which will reduce the number of messages produced (only information, warning, and error messages) and thus increase the overall speed. “DEBUG” will usually be used for debugging purposes, but has the disadvantage that it generates a very high of messages from all classes in the software. As of version 1.6, one can also use “FILE”, meaning that the system will try to read a set of configured logging instructions from a file “log4j.xml” which should then be present.

This functionality is reserved to users that have experience with [Apache log4j](#) configuration.

A sample log4j\_sample.xml file is however delivered with the software, allowing to at least start from an example file.

### **Changes in the XSLT that is generated**

The XSLT that is being automatically generated looks a bit different than the one that was generated in earlier versions of the SDTM-ETL software.

Most notably is that the “my:” as prefix for the ODM namespace has been replaced by the “odm:” prefix. This is an aesthetic change only, and should change nothing in the executability of the XSLT files.

It however means that one should not manually merge XSLT files from previous versions with one of the current version.

### **Faster drag-and-drop**

With very large mappings (i.e. when having a good amount of instances of the same (e.g. LB) domain), and complicated study designs (resulting in large ODM trees), drag-and-drop from the ODM tree to the SDTM structure could become relatively slow.

This could already be improved by switching off the “traffic lights” in the ODM tree, using the menu “Options -> Settings” and then checking the checkbox “View ODM Items without traffic lights”.

We made some changes in the code however so that drag-and-drop has become faster in general.

## **Offline execution of existing mappings**

Once a set of mappings is ready, it can be advantageous to be able to execute the mappings offline, i.e. from a batch file, so without needing to start the SDTM-ETL graphical interface.

This is now possible in version 1.6, but only on machines for which there is already a license available for the SDTM-ETL software.

For execution of the mappings to generate the SDTM datasets on a server, a separate server license (execution-only license) will be necessary.

For further technical details see the manual “Executing SDTM-ETL in batch mode”.

## **SDTM-ETL Light**

Similarly, some of our customers have asked for a “light” version of SDTM-ETL that only executes mappings for which there is already a define.xml file with mappings available. This would e.g. allow them to send a define.xml with mappings to their customers, who can then execute the mappings on their datasets.

The SDTM-ETL Light software has a graphical interface which is very similar to the one in SDTM-ETL, but which does not allow editing of mappings, nor to define new domains or variables.

Only generation of the XSLT from the loaded define.xml and executing that XSLT on clinical data is allowed.

Please remark that SDTM-ETL Light is a separate product for which a separate license may be required.

