

CLI and Batch / SDTM-ETL Light execution of SDTM-ETL v.4.2

Last update: May 1, 2023

This document describes how to run the transformations for generating SDTM from the command line using the CLI (Command Line Interface) or in batch, and how to use the "SDTM-ETL Light" version of the software, for executing mappings and generate SDTM files, but without working on the mappings themselves.

Running SDTM-ETL execution in batch mode / CLI mode

It is also possible to execute the mappings, and generate SDTM datasets from existing mappings, in line command or batch mode, e.g. on a server, using commands or scripts that are executed using the CLI (Command Line Interface).

Although it is easily possible to execute the software for generating SDTM/SEND datasets by typing the command from the command line, one will usually store the command in a file (".bat" file on Windows, ".sh" file on Linux/Unix) and run that file. Such files are usually designated as "batch file".

The contents of such a batch file will be very similar to the SDTM-ETL.bat file, but there are also a good number of differences, as the execution command has a good number of additional parameters.

First of all, the batch file should set the libraries for executing the software, for example:

The base for executing in batch mode is given by:

```
java -Xms256M -Xmx1024M -jar SDTM-ETL_Executor.jar  
com.xml4pharma.sdtmetl.light.SDTMETLExecutor ...
```

The -Xms256M and -Xmx1024M parameters indicate the amount of memory the java VM is allowed to claim, the first being the amount at start, and the latter being the maximum allowed at all. 1024MB will be sufficient in most cases, but the user may decide to either a lower value or increase the amount of memory claimed by the Java Virtual Machine (VM), depending on the amount of mappings and the sizes of the files to be transformed.

It is advised never to allow more than about 50% of the available physical memory, as the operating system usually also uses a lot of memory.

The command is followed by the set of keywords and parameters. For example, in the case SAS-XPT must be generated:

```
java -Xms256M -Xmx1024M -jar SDTM-ETL_Executor.jar  
com.xml4pharma.sdtmetl.light.SDTMETLExecutor -DEFINEVERSION 2.0  
-GENERATECOMMENTSDOMAIN  
-DEFINEFILELOCATION C:\SDTM-ETL\TestFiles\MyStudy_Six_Domains_define_2_0.xml  
-CLINICALDATAFILELOCATION C:\SDTM-ETL\TestFiles\ODM1-3\MyStudy_ODM_1_3.xml  
-STUDYID MyStudy  
-ODMVERSION 1.3  
-METADATAFILELOCATION C:\SDTM-ETL\TestFiles\ODM1-3\MyStudy_ODM_1_3.xml  
-GENERATESASXPT  
-SASXPTDIRECTORYLOCATION C:\SASXPTSamples\  
-GENERATERELRECDOMAIN  
-GENERATECOMMENTSDOMAIN  
-GENERATESUPQUALDOMAINS
```

All these should be in a single line (no carriage return).

Or when Dataset-XML must be generated:

```
java -Xms256M -Xmx1024M -jar SDTM-ETL_Executor.jar
com.xml4pharma.sdtmetl.light.SDTMETLExecutor
-DEFINEVERSION 2.0 -GENERATECOMMENTSDOMAIN
-DEFINEFILELOCATION C:\SDTM-ETL\TestFiles\MyStudy_Six_Domains_define_2_0.xml
-CLINICALDATAFILELOCATION C:\SDTM-ETL\TestFiles\ODM1-3\MyStudy_ODM_1_3.xml
-STUDYID MyStudy -ODMVERSION 1.3
-METADATAFILELOCATION C:\SDTM-ETL\TestFiles\ODM1-3\MyStudyNew_ODM_1_3.xml
-GENERATEDATASETXML
-DATASETXMLDIRECTORYLOCATION C:\temp\ -GENERATERELRECDOMAIN
-GENERATECOMMENTSDOMAIN
-GENERATESUPQUALDOMAINS
```

I.e. each keyword starts with a dash ("-") and is, depending on the keyword itself, followed by a parameter value.

The list of keywords and parameters is given in the following table. Those given in bold-italic are *mandatory*. If one of these is absent, the program will issue a warning and stop.

All parameters/keywords may be given either in lower case or mixed case or uppercase. They will however be translated to uppercase before interpretation by the program.

| Parameter / keyword | Parameter value | Explanation |
|---|-----------------|---|
| <i>-DEFINEFILELOCATION</i> | file path | absolute or relative path of the define.xml file containing the mappings. See also remark on file paths. |
| <i>-CLINICALDATAFILELOCATION</i> | file path | absolute or relative path of the file containing the clinical data. See also remark on file paths. |
| -METADATAFILELOCATION | file path | only necessary when the mappings use data from the metadata section of the study, and this section is not included in the file with clinical data. The former is essentially the case when the SDTM-ETL functions "name()" or "question()" are used in one or more mappings. See also remark on file paths. |
| -ADMINDATAFILELOCATION | file path | only necessary when the mappings use data from the administrative data section of the study, and this section is not included in the file with clinical data. The former is essentially the case when the SDTM-ETL functions "sitename()" is used in one or more mappings. See also remark on file paths. |
| -XMLOUTPUTFILELOCATION | file path | absolute or relative path of the file where the SDTM/SEND data in XML format will be written to. Required when the output is in Dataset-XML format. See also remark on file paths. |

| Parameter / keyword | Parameter value | Explanation |
|--------------------------|---|--|
| -STUDYID | OID of the study | The OID of the study, as given in the Study/@OID attribute of the file with metadata (or ClinicalData/@StudyOID in the file with clinical data). |
| -ODMVERSION | version of the ODM standard used for the study metadata and clinical data | Default is "1.2". Can be either "1.2", "1.3" or "1.3.1". |
| -DEFINEVERSION | Version of the define.xml file containing the mappings | Can be "1.0" or "2.0" or "2.1" Remark that Define-XML v.1.0 is outdated and has limited support in SDTM-ETL 4.0. |
| -GENERATESUPPQUALDOMAINS | none | keyword to indicate that supplemental qualifier datasets should be generated for: a) automated splitting of records with more than 200 characters (usually necessary when also SAS XPT datasets need to be generated) b) non-standard SDTM variables (NSVs) that need to be moved to the corresponding SUPP--domain. Remark that when this keyword is omitted, no splitting is done, which may be problematic when output is in SAS XPT format. |
| -GENERATERELRECDOMAIN | none | keyword to indicate that a RELREC domain needs to be automatically created for "RELREC" variables (see user manual for further details). |
| -GENERATE_1_N_RELREC | none | keyword to indicate that the software should attempt to generate 1:n relationships in the automatically generated RELREC domain. Only to be used in combination with the -GENERATERELRECDOMAIN keyword. |
| -GENERATECOMMENTSDOMAIN | none | keyword to indicate that a Comments (CO) domain needs to be automatically generated from "Comment" variables (see user manual for further details). |
| -GENERATESASXPT | none | keyword to indicate that SAS XPT SDTM/SEND datasets should be generated. It is recommended to then also use the keyword "-GENERATESUPPQUALDOMAINS" as otherwise records of more than 200 characters will be cut after the 200th character. |

| Parameter / keyword | Parameter value | Explanation |
|-------------------------------|-----------------|--|
| -SASXPTDIRECTORYLOCATION | file path | absolute or relative path of the directory where the SAS XPT files need to be written to. Only to be used in combination with the "-GENERATESASXPT" keyword |
| -GENERATEDATASETJSON | none | |
| -DATASETJSONDIRECTORYLOCATION | file path | |
| -GENERATEDATASETXML | none | keyword to indicate that modern CDISC Dataset-XML datasets should be generated. Either "-GENERATESASXPT" <u>or</u> "-GENERATEDATASETXML" <u>or</u> "-GENERATEUTF8CSV" must be provided |
| -DATASETXMLDIRECTORYLOCATION | file path | absolute or relative path of the directory where the Dataset-XML files need to be written to. Only to be used in combination with the "-GENERATEDATASETXML" keyword |
| -GENERATEUTF8CSV | none | keyword to indicate that UTF-8 encoded CSV should be generated. Either "-GENERATESASXPT" <u>or</u> "-GENERATEDATASETXML" <u>or</u> "-GENERATEUTF8CSV" must be provided |
| -UTF8CSVDIRECTORYLOCATION | file path | absolute or relative path of the directory where the UTF-8 encoded files need to be written to. Only to be used in combination with the "-GENERATEUTF8CSV" keyword |
| -UTF8CSVADDHEADERLINES | none | Add 2 header lines to the UTF-8 encoded CSV file, one with the variable names, and one with the variable labels |
| -USETYPEDITEMDATA | none | keyword to indicate that the clinical data use "typed ItemData" elements (e.g. ItemDataInteger". May only be used when ODMVERSION is either 1.3 or 1.3.1 (so not for ODM version 1.2) |
| -ISOPENCLINICAVERSION | none | keyword to indicate that support needs to be provided for the OpenClinica 3.x extensions to the ODM standard. Can only be used with the "OpenClinica version" of the software. |

Using "batch" execution in production

When having a process in place for doing the batch execution in production mode, it may be wise to use a define.xml file with the template removed, i.e. only keeping the study-specific domain/dataset definitions, this as loading the template takes considerable time.

One can generate such a "template-free" define.xml in SDTM-ETL using the menu "File - Save cleaned define.xml", but taking care that all mappings ("MethodDef" elements in the define.xml) are retained.

Of course, one should always test the batch execution carefully before taking it into production.

Example batch files

Example batch files can be found in the distribution. These can be used as start for your own files, but need to be adapted for your own use case, especially for the file paths. Following files are available:

| | |
|---|--|
| Batch_example_SDTCM-ETL_SASXPT.bat | Example file for generation of SDTCM files in SAS XPT files |
| Batch_example_SDTCM-ETL_DatasetJSON.bat | Example file for generation of SDTCM files in modern Dataset-JSON format. |
| Batch_example_SDTCM-ETL_DatasetXML.bat | Example file for generation of SDTCM files in modern Dataset-XML format |
| Batch_example_SDTCM-ETL_SASXPT_define_2-1.bat | Example file for generation of SDTCM files in SAS XPT files from a Define-XML 2.1 file with mappings |
| Batch_example_SDTCM-ETL_SASXPT_RWS.bat | Test example file using mappings that use RESTful web services for generating SDTCM records |

These example batch files use example ODM and Define-XML files from the "TestFiles" folder that is included in the distribution.

Remark concerning file paths

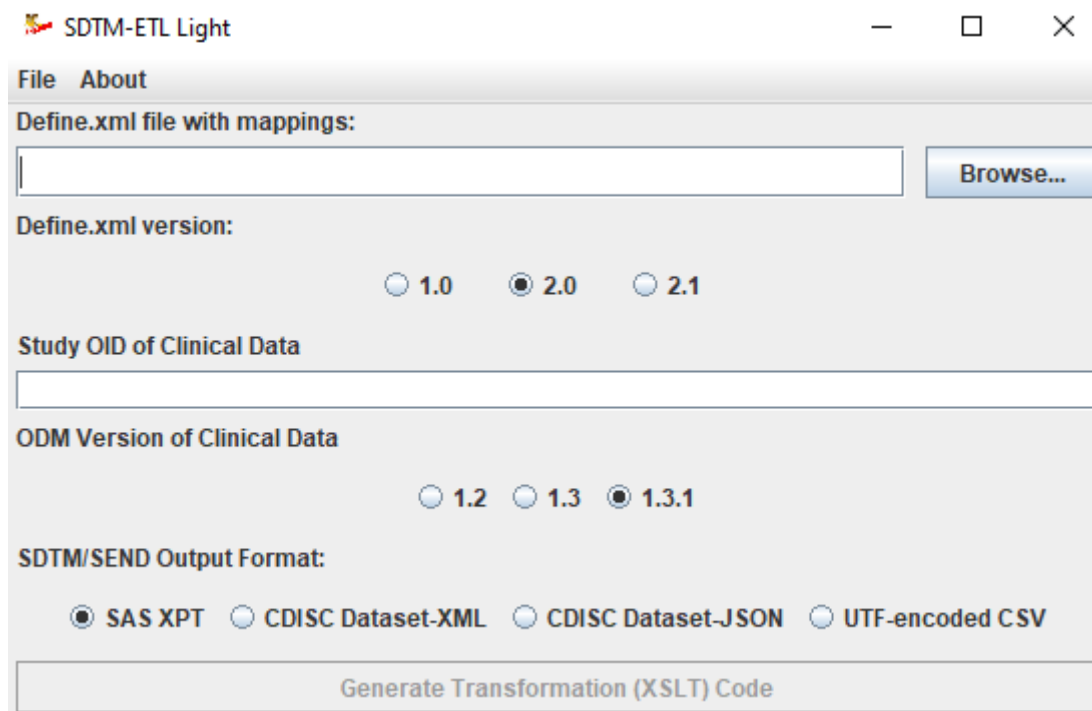
File paths can be absolute or relative with respect to the directory from which the batch execution is run. In case the file path contains one or more blanks, the whole path should be embedded in double quotes. For example:

```
... -DEFINEFILELOCATION "C:\Documents and Settings\My Define Files\MyStudyDefine.xml"
```

Starting transformation-execution-only from a simple graphical user interface ("SDTM-ETL Light")

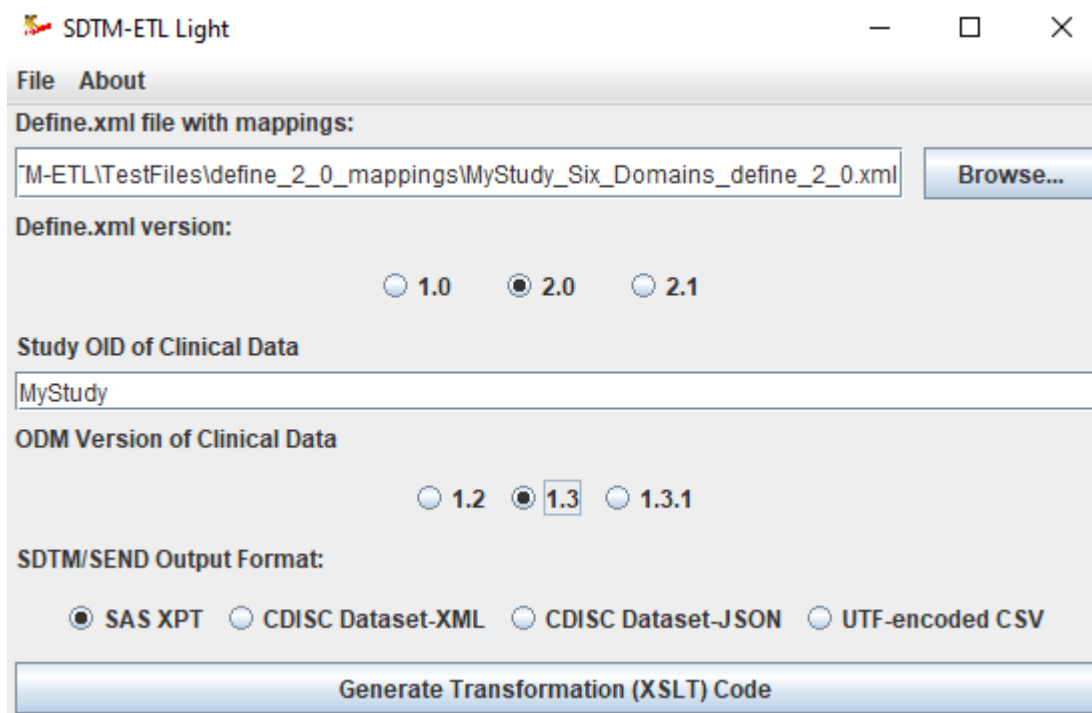
Whereas the instructions provided above are ideal for executing the transformations on a server, e.g. at regular times using a [cron job scheduler](#), some people may also want to execute the mappings on a normal PC or workstation, but without working on the mappings themselves, i.e. just executing them. For this use case, we developed a simple graphical interface. This "light" version can be started using the file "*SDTM-ETL_execution_only_GUI.bat*".

When started, the following graphical user interface is shown:



The screenshot shows the SDTM-ETL Light GUI window. The title bar reads "SDTM-ETL Light". The menu bar contains "File" and "About". The main area is titled "Define.xml file with mappings:" and contains an empty text box and a "Browse..." button. Below this is the "Define.xml version:" section with radio buttons for "1.0", "2.0" (selected), and "2.1". The "Study OID of Clinical Data" section has an empty text box. The "ODM Version of Clinical Data" section has radio buttons for "1.2", "1.3", and "1.3.1" (selected). The "SDTM/SEND Output Format:" section has radio buttons for "SAS XPT" (selected), "CDISC Dataset-XML", "CDISC Dataset-JSON", and "UTF-encoded CSV". At the bottom is a large button labeled "Generate Transformation (XSLT) Code".

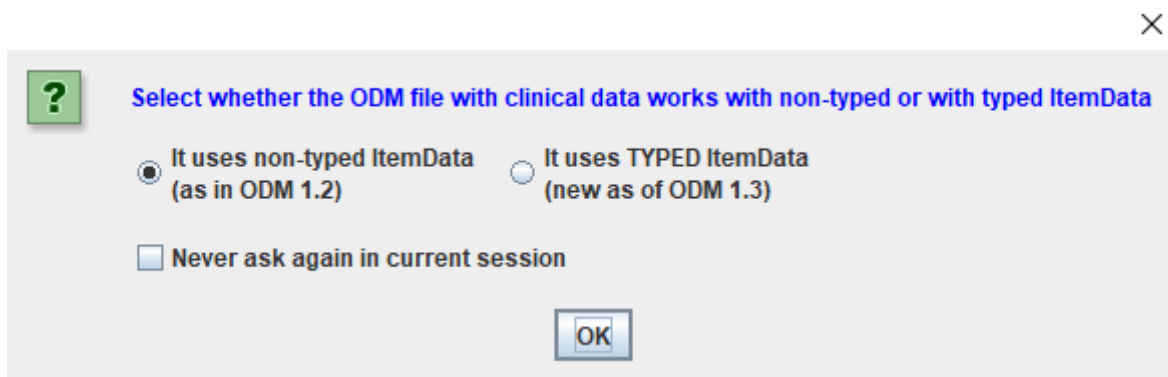
One should then fill in all fields and select the appropriate checkboxes, e.g.:



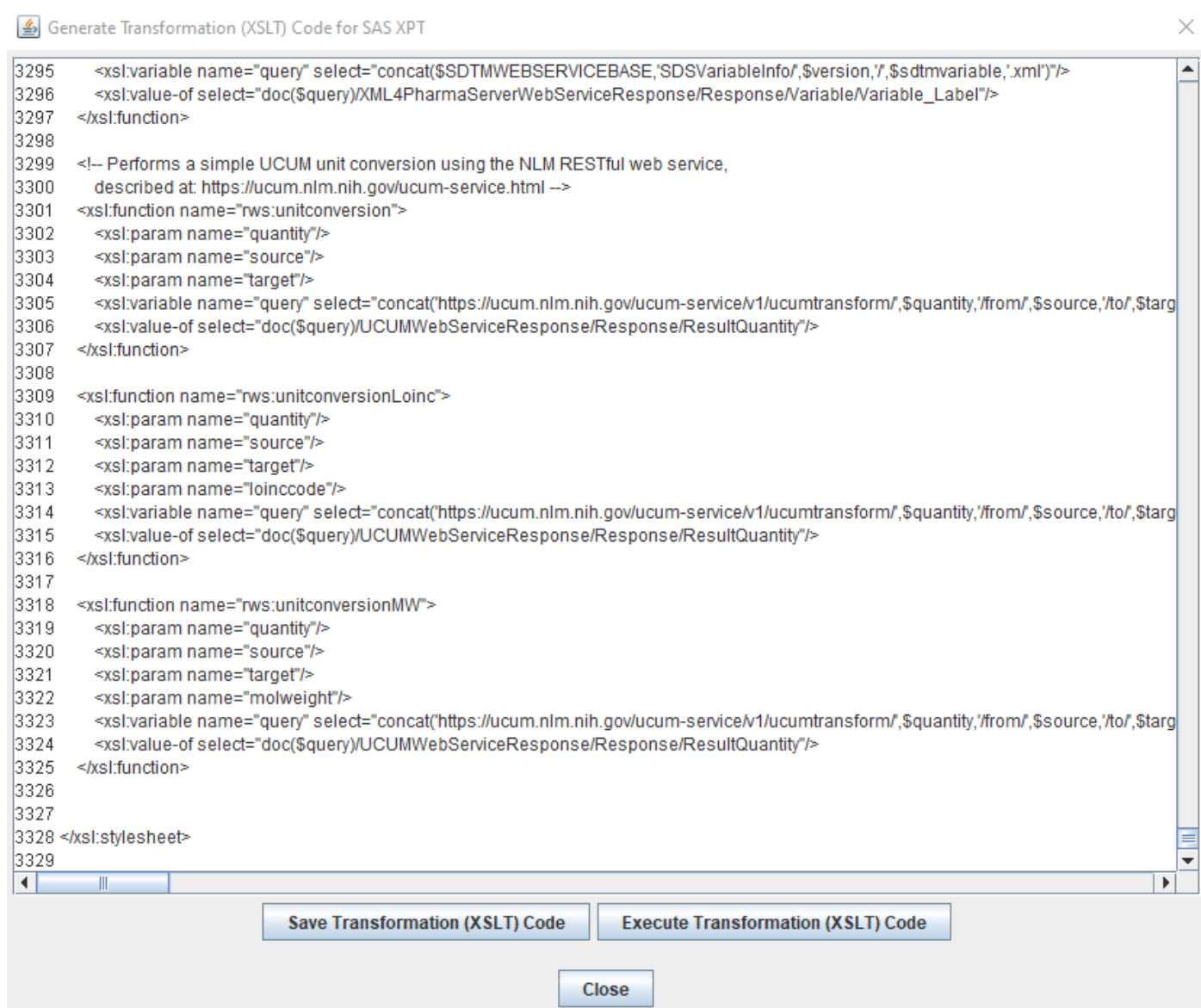
The screenshot shows the SDTM-ETL Light GUI window with the fields filled and options selected. The title bar reads "SDTM-ETL Light". The menu bar contains "File" and "About". The main area is titled "Define.xml file with mappings:" and contains a text box with the path "M-ETL\TestFiles\define_2_0_mappings\MyStudy_Six_Domains_define_2_0.xml" and a "Browse..." button. Below this is the "Define.xml version:" section with radio buttons for "1.0", "2.0" (selected), and "2.1". The "Study OID of Clinical Data" section has a text box containing "MyStudy". The "ODM Version of Clinical Data" section has radio buttons for "1.2", "1.3" (selected), and "1.3.1". The "SDTM/SEND Output Format:" section has radio buttons for "SAS XPT" (selected), "CDISC Dataset-XML", "CDISC Dataset-JSON", and "UTF-encoded CSV". At the bottom is a large button labeled "Generate Transformation (XSLT) Code".

And can then choose between "SAS XPT" (SAS Transport 5), CDISC's own Dataset-XML or Dataset-JSON format, and UTF-8 encoded CSV for the output format.

When the clicking the "Generate Transformation (XSLT) Code, the dialogs that we already know from the full version pop up:



followed by:



Upon clicking the "Execute Transformation (XSLT) Code" button, the classic dialog for entering source and target and the different options is displayed:

Execute Transformation (XSLT) Code for SAS-XPT

ODM file with clinical data:
 D:\SDTM-ETL\TestFiles\ODM1-3\MyStudy_ODM_1_3.xml Browse...

MetaData in separate ODM file Browse...

Administrative data in separate ODM file Browse...

Save output XML to file Browse...

Perform post-processing for assigning --LOBXFL

Split records > 200 characters to SUPP-- records

Move non-standard SDTM Variables to SUPP-- Move Comment Variables to Comments (CO) Domain

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 Re-sort records using define.xml keys

Save Result SDTM tables as SAS XPORT files Perform CDISC CORE validation on generated SAS XPORT files

SAS XPORT files directory:
 D:\SASXPTSamples

Add location of SAS XPORT files to define.xml Store link as relative path

Additionally generate a merged dataset for 'split' domain datasets

Messages and error messages:

Execute Transformation on Clinical Data

Close

As usual, one can select several "special" options, such as optimizing the XPT files for file size¹, using the checkbox "Adapt Variable Length for longest value", generation of "NOT DONE" records in the case of QS datasets, and generation of "1:N" RELREC relationships".

New in SDTM-ETL 4.2 is that one can ask the system to re-sort the records according to the "keys" ("KeySequence" attributes) in the define.xml, which should only be necessary when the data in the source ODM is not in chronological order). The checkbox for this option will only be made visible when the system detects such keys in the define.xml file from which the mapping execution is started.

Another new feature in SDTM-ETL is to perform [CDISC CORE validation](#) on the generated datasets, essentially replacing the "crappy" Pinnacle21 validation engine.

See the separate documents "[SDTM-ETL 4.2 new features overview](#)" and "[Validating SDTM/SEND datasets using CDISC-CORE](#)" on the SDTM-ETL website.

The final step is then to click the button "Execute Transformation on Clinical Data" which starts the transformation execution.

For further details about the last steps, see the tutorials on the SDTM-ETL website.

¹ This is due to that the outdated SAS-XPT is a highly inefficient format, where values, shorter than the variable length, are padded with blanks until the given length is reached.