

SDTM-ETL Version 5.3: New Features in development

Author: Jozef Aerts

Date last update: 2026-08-12

This document contains a list of new features of the SDTM-ETL v.5.3 mapping software that are currently developed or already made it into the beta version.

If, as a customer or potential customer would like to see any other new features, please let us know so that we can implement them. Just send a mail to info@XML4Pharma.com.

Feature	Status
Implementation of SDTMIG-4.0 / SENDIG-3.0 The draft publications of SDTMIG-4.0 and SENDIG-3.0 have been implemented, including the new NS-- dataset structure for non-standard variables (NSVs).	To be finalized once the final SDTMIG-4.0 and SENDIG-3.0 have been published.
Automated generation of ValueList-s in the define.xml based on CDISC "SDTM Dataset Specializations".	In development.
After drag-and-drop, ask user to immediately provide Source and Origin and add these to the underlying define.xml. If a file named "Default_Source_Origin.csv" is present, the proposed Source and Origin will be taken from this file. This feature can be switched off.	Realized.
Perform CDISC CORE validation on single table in local "table view". CORE validation (including rules from the "SDTM Dataset Specializations" can be executed immediately on a single table in the "table view". Table cells that have a violation are colored orange. This enables to do a fast CORE validation, and have the cells having violations marked within the local table itself without needing to generate SAS-XPT or Dataset-JSON files.	Realized.
Faster Dataset-JSON generation starting from a local "table view"	Realized.
"SDTM Dataset Specialization" checking in in the "mapping script editor". For SDTM, the mapping script can immediately be checked against the CDISC "SDTM Dataset Specializations". For example when there is a line: if(\$VS.VSTESTCD == 'DIABP') { \$VS.VSORRESU = 'cm'; this will then be marked as a violation.	Realized.
Direct checking against assigned CDISC Controlled Terminology in the "mapping script editor". Within the mapping script editor, the assignments can immediately be checked against the assigned CDISC Controlled Terminology (subsetted codelist or original codelist).	Realized.
New algorithm for setting up "CodeList-CodeList mapping wizard" in the case of large set of Controlled Terminology. Setting up the "CodeList-CodeList mapping wizard" could take 1-2 minutes in the case of a (very) large CDISC-CT associated codelist (e.g. for LBTESTCD, LBTEST, UNIT). Due to the new algorithm, this is reduced to a few seconds.	Realized.
Right-click in "mapping script editor" shows values of the assigned CDISC Controlled Terminology and allows a direct insert into the mapping script text.	Realized.
Many improvements in the "Mapping Completeness Report".	Realized.
Contents of "Mapping Completeness Report" can be exported into an Excel or CSV file, with our without the mapping scripts themselves. This allows the "Mapping Completeness Report" to be shared with colleagues in the team for discussion. An example can be found at: https://www.xml4pharma.com/SDTM-ETL/CES_mapping_completeness_report.xlsx	Realized.
Redesign of the "ODM Clinical Data Viewer". When using the menu "View - ODM Clinical Data", a new window with the results and options is displayed. The layout of this window has been redesigned, including that when it is enlarged, the table with the results is also enlarged. This makes it easier to review the contents in the case of long texts.	Realized.

Parse character-delimited text files (like CSV-files) for direct use in the mappings. In some cases, it makes sense to be able to import data from external files with assignments (e.g. from third parties) without needing to convert them to ODM first. For example, a CSV file containing the MedDRA assignments of AEDECOD, AELLT, AELLTCD, etc., that comes from an external source. A mapping script function is being developed to enable such a direct import. For example: <code>\$AE.AEDECOD = <i>parse-external-file</i>('D:\temp\meddra.txt','',\$USUBJID,1,\$AE.AESPID,2,3)</code> with the first argument being the file from where the information is taken from, the second argument being the delimiter character between the fields in the file, the third argument passing the value of USUBJID (usually the third variable in the domain), the fourth argument being the field number of the field with the subject identifier (corresponding to USUBJID) in the SDTM, the fifth argument passing the value of the second identifier (this will often be a --SPID variable, but can e.g. also be a --CAT or --SCAT variable), the sixth argument being the field number in the text file corresponding to the second identifier, and the last argument being the field number the information needs to be taken from. In the case of AEDECOD, this will usually be the field with the "MedDRA Preferred Term"	In test phase