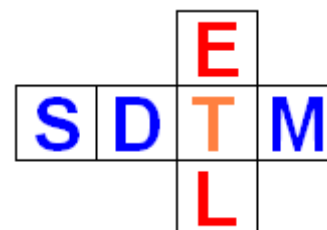


## SDTM-ETL 5.2: Summary of New Features

Author: Jozef Aerts, XML4Pharma

Last update: 2026-01-15



### Summary

This document contains a summary of the most important new features of SDTM-ETL 5.2 and bug fixes.

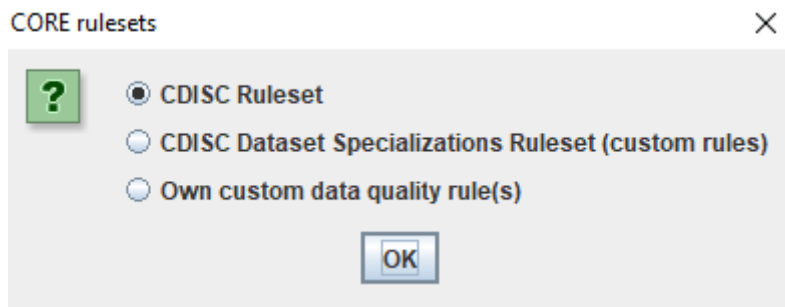
### Table of Contents

Implementation of CORE Validation v.0.14.2 .....	1
XML-Schema validation only of underlying define.xml .....	2
QS with "branching variable" QSCBRFL .....	3
Inform the user to first map the topic variable .....	4
Optional semi-automatic adaption of XPath expressions for complicated selections .....	5
Automatically generate relative XPath for VISITNUM and VISIT variables .....	10
Extended methods for automated retrieval of page numbers from aCRFs .....	13
Features specifically for Apple Macintosh .....	16
Other minor updates .....	17
Initial ODM tree display .....	17
Tooltips on "Except for ..." and "Only for ..." buttons .....	17
Bug fixes .....	18
Identifier variables in generated SQAPxx datasets .....	18
Automated VISITNUM generation for "unscheduled" visits that do not have the --DTC variable present .....	19
Other small fixes .....	19
Other remarks .....	19

## Implementation of CORE Validation v.0.14.2

The newest version of CDISC CORE (CDISC Open Rules Engine) has been added. It contains hundreds of new validation rules, and considerable improvements in the algorithms and reporting.

It also contains 3334 rules that were derived from the latest version of the "[SDTM Dataset Specializations](#)", so that the user now has the choice between the CDISC+FDA rules, data quality rules derived from the SDTM Dataset Specializations and own developed CORE rules in YAML or JSON format.

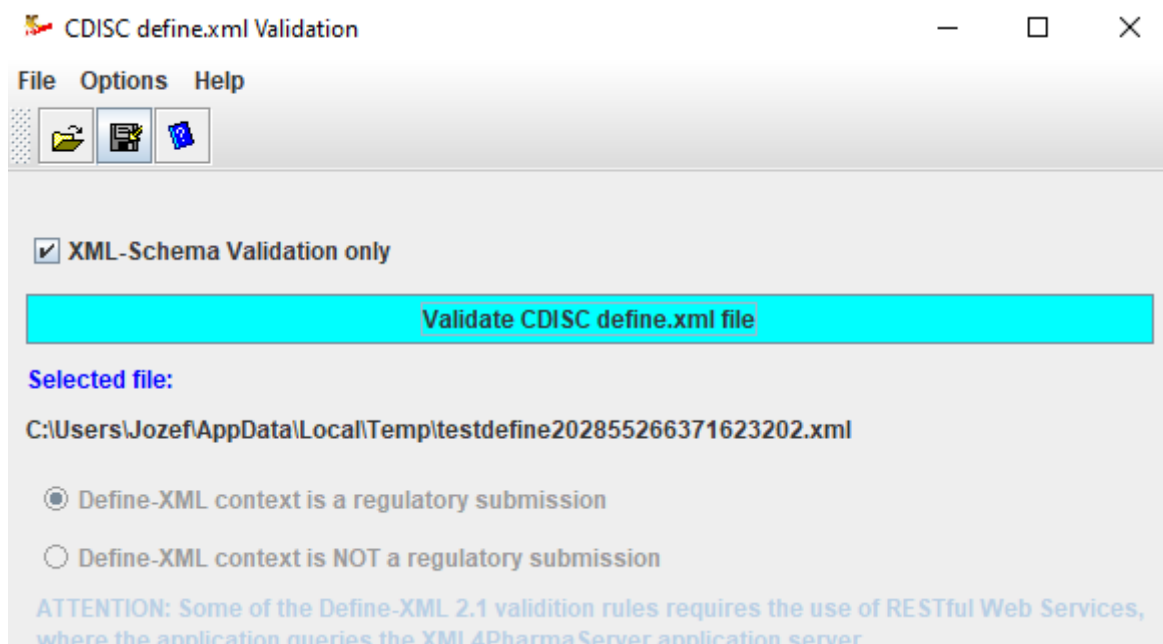


Furthermore, a number of improvements have been made to the CORE GUI for selecting rules and especially the search features.

By default, SDTM-ETL 5.2 is delivered with CORE executables for Windows. If you would like to obtain SDTM-ETL with the CORE executables for either Linux or Mac, please let us know and we will take care of it.

## XML-Schema validation only of underlying define.xml

Essentially, everything about the domains, datasets, variables, codelists, ... in SDTM-ETL is stored in the underlying define.xml, which can be validated using the menu "Validate - Validate define.xml structure". On request of some users, we added the option to do a (fast) "XML-schema only" validation. This resulted in an additional checkbox in the "CDISC define.xml Validation" panel. When it is checked, a set of other options is automatically disabled.



## QS with "branching variable" QSCBRFL

Mid-end 2023, the FDA added a new variable "QSCBRFL" (Conditional Branching Flag) specifically for the QS (Questionnaires) domain. From the FDA "Technical Conformance Guide":

If instructions on how to record and/or score responses to items not done due to conditional branching are available from the instrument developer, then records for items with conditional branching should be included in the submission dataset with the following:

- QSCBRFL = "Y"
- QSORRES, QSSTRESC, and QSSTRESN would be assigned according to the instrument's instructions

If instructions on how to record and/or score responses to items not done due to conditional branching are not available from the instrument developer, then records for items with conditional branching should be included in the submission dataset with the following:

- QSSTAT = "NOT DONE",
- QSCBRFL = "Y" and,
- QSORRES, QSSTRESC, and QSSTRESN all set to null.
- QSREASND = missing or the sponsor-provided reason (e.g., "LOGICALLY SKIPPED ITEM")

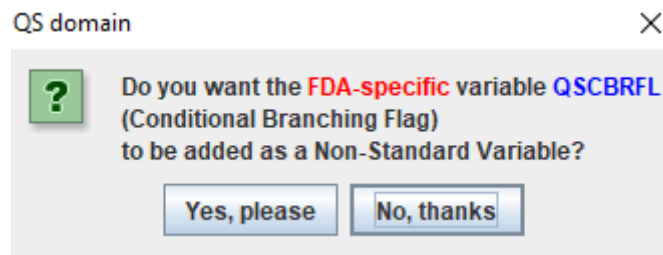
This in addition to the "logically skipped items" rule introduced several years before.

To me, the usage of this variable is not entirely clear, especially the sentence "If instructions on how to record and/or score responses to items not done due to conditional branching are (not) available from the instrument developer, ...". Question is how that can be checked / validated e.g. by a CORE rule. The examples in [some presentations](#) are not very helpful either, especially as they do not contain work flow charts. Essentially, such (metadata) information should not be added to any SDTM table, instead e.g. the workflow in [BPMN](#) (Business Process Model and Notation), for which there is an XML representation available, should be used. It could even be used within the define.xml as an extension. But yes, as usual "if one only has SAS-XPT, everything is a table" ...

Please be aware that QSCBRFL is not a formal variable of even SDTMIG-3.4, and it is totally unclear from the FDA documentation and CDISC presentations whether this variable should be treated as a "non-standard" variable, i.e. "banned" to SUPPQS.

As it is expected that QSCBRFL will become a standard variable in the future SDTMIG version 3.5 (or 4.0) I am afraid that we will need to await the new SDTMIG to get full clarity.

So, for the moment, we have implemented the use of the FDA-specific QSCBRFL variable that when the user sets up a "study-specific instance" of QS, the system asks:



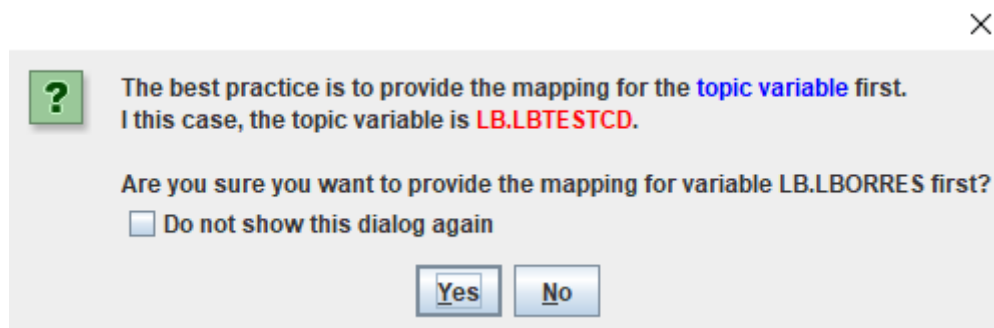
If "Yes, please" is clicked, the variable is added at the end, as a "non-standard variable":

D	LB.LBBLFL	LB.LBFAST	LB.LBDRVFL	LB.LBTOX	LB.LBTC
M	VS.VSELT	VS.VSTPTREF	VS.VSRFTDTC		
F	QS.QSRFTDTC	QS.QSEVLINT	QS.QSCBRFL		

At the moment, you cannot change the "Role" of QSCBRFL yet to make it a "standard variable", if you want to have it that way (but which will generate a lot of validation errors in any validation tool), but you will need to edit the define.xml manually outside the SDTM-ETL software - we do not want to be made responsible for violating the SDTM standard!

## Inform the user to first map the topic variable

It is always advantageous to first generate the mapping for the topic variable: --TRT in the case of "Interventions", --TERM in the case of "Events" and "--TESTCD" in the case of "Findings", as this usually is also the "looping variable", and when executing the mappings, all XPath expressions will be made against the selection in the "looping variable". Although this is well documented in the many tutorials, some users still hesitate "with which variable to start". To help them, we added a new feature that when the user tries to drag-and-drop to a non-topic variable before a mapping for the topic variable has been mapped, a dialog is displayed:



If the user clicks "Yes", the mapping will be continued anyway. When "No" is clicked, the mapping process is stopped and the user is prompted to first do the mapping for the topic variable. The future popping up of the dialog can be avoided by checking the checkbox "Do not show this dialog again".

# Optional semi-automatic adaption of XPath expressions for complicated selections

**IMPORTANT: This is for advanced usage only!**

Some of our users manually adapt the XPath expression for the "looping variable" (which usually is the "topic variable", i.e. --TESTCD, --TRT or --TERM) or by checking the "View/Edit XPath Expression" checkbox immediately after the drag-and-drop, to do some very special things like:

- excluding one or more subjects by ID
- excluding data points collected before or after a specific date

For example, when for LB, having the selection:

Import ItemDef: Erythrocytes - for SDTM Variable LB.LBTESTCD

☐ 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)

ItemOID

CollectionExceptionConditionOID

<input checked="" type="checkbox"/> Generalize for all StudyEvents	Except for ..	No Exceptions	Only for ..	2 Inclusions
<input type="checkbox"/> Generalize for all Forms	Except for ..	No Exceptions	Only for ..	No Inclusions
<input type="checkbox"/> Generalize for all ItemGroups	Except for ..	No Exceptions	Only for ..	No Inclusions
<input checked="" type="checkbox"/> Generalize for all Items	Except for ..	No Exceptions	Only for ..	2 Inclusions

ODM ItemDef Length: 8

SDTM Variable Length: 8

☐ Set SDTM Variable Length to ODM ItemDef Length

☒ View/Edit XPath expression (advanced)

Included items:  
I\_LB\_RBC - Erythrocytes  
I\_LB\_WBC - Leukocytes

OKCancel

where we selected that for the mapping to LBTESTCD, data can come from 2 different visits, for 2 items (in this case "Erythrocytes" and "Leukocytes")<sup>1</sup>. We also want to e.g. exclude subject "002" for some reasons<sup>2</sup>, so we have the checkbox "View/Edit XPath expression" checked.

After then going to the usual procedure, a new window appears:

<sup>1</sup> This is an extremely simplified example, just to show how the feature works.

<sup>2</sup> Remark that "Screen Failure" data should normally always be included. Some of our customers however had cases that they needed to exclude one or more subjects, or have their data treated separately, usually true to inconsistencies in the provided or by the "[ODM Generator](#)" generate data. The better way of course would be to have an ODM in which the specific subject was already be removed in advance. Our users do however not always have the authority to do so.

If you do have high quality ODM data, this should not occur.

View / Edit XPath Conditions



	Condition for StudyEventData:	<input type="checkbox"/> Edit	[@StudyEventOID="WEEK_1" or @StudyEventOID="WEEK_2"]
	Condition for FormData:	<input type="checkbox"/> Edit	[@FormOID="F_LAB"]
	Condition for ItemGroupData:	<input type="checkbox"/> Edit	[@ItemGroupOID="IG_LB_HEMATOLOGY"]
	Condition for ItemData:	<input type="checkbox"/> Edit	[@ItemOID="I_LB_RBC" or @ItemOID="I_LB_WBC"]
	Value selection:	<input type="checkbox"/> Edit	@ItemOID

OK Cancel

If we know something about the ODM tree for the "Clinical Data" part, we do know that the hierarchy is:

### 3.1.4.1 SubjectData

Body:

(AuditRecord?, Signature?, Investigator?)

Attributes:

**SubjectKey**

subjectKey

TransactionType

(Insert | Update | Delete)

ClinicalData  
SubjectData  
StudyEventData  
FormData  
ItemGroupData  
ItemData

and that the identifier for the subject the attribute "SubjectKey" is.

So (and yes, this is "advanced"), we can adapt the XPath for "StudyEventData" to:

View / Edit XPath Conditions



	Condition for StudyEventData:	<input checked="" type="checkbox"/> Edit	[@StudyEventOID="WEEK_1" or @StudyEventOID="WEEK_2"] [not(../@SubjectKey="002")]
	Condition for FormData:	<input type="checkbox"/> Edit	[@FormOID="F_LAB"]
	Condition for ItemGroupData:	<input type="checkbox"/> Edit	[@ItemGroupOID="IG_LB_HEMATOLOGY"]
	Condition for ItemData:	<input type="checkbox"/> Edit	[@ItemOID="I_LB_RBC" or @ItemOID="I_LB_WBC"]
	Value selection:	<input type="checkbox"/> Edit	@ItemOID

OK Cancel

leading to the mapping script:

Mapping Script Editor for SDTM Variable LB.LBTESTCD

```

1 # Mapping using ODM element ItemData with ItemOID I_LB_RBC - value from attribute ItemOID
2 # Generalized for all StudyEvents
3 # Generalized for all Items within the ItemGroup
4 # Mapping for ODM Items (I_LB_RBC, I_LB_WBC) to SDTM CodeList LB.LBTESTCD
5 # with CodeList OID "CL.C65047.LBTESTCD"
6 CODEDVALUE = xpath(/StudyEventData[@StudyEventOID="WEEK_1" or @StudyEventOID="WEEK_2"] [not(../@SubjectKey="002")]/FormData[@FormOID="F_LAB"]/ItemGroupData[@ItemGroupOID="IG_LB_HEMATOLOGY"]/ItemData[@ItemOID="I_LB_RBC" or
7 if ((CODEDVALUE = "RBC")) {
8   $NEWCODEDVALUE = "RBC";
9 } elseif ((CODEDVALUE = "I_LB_WBC")) {
10   $NEWCODEDVALUE = "WBC";
11 } elseif ((CODEDVALUE = "")) {
12   $NEWCODEDVALUE = "";
13 } else {
14   $NEWCODEDVALUE = "NULL";
15 }
16 LB.LBTESTCD = $NEWCODEDVALUE;
17
18

```

and when executing this mapping to:

CES:LB				
STUDYID	DOMAIN	USUBJID	LB.LBSEQ	LB.LBTESTCD
CES	LB	001	1	RBC
CES	LB	001	2	WBC
CES	LB	001	3	RBC
CES	LB	001	4	WBC
CES	LB	001	5	RBC
CES	LB	001	6	WBC
CES	LB	003	1	RBC
CES	LB	003	2	WBC
CES	LB	003	3	RBC
CES	LB	003	4	WBC
CES	LB	003	5	RBC
CES	LB	003	6	WBC
CES	LB	004	1	RBC
CES	LB	004	2	WBC
CES	LB	004	3	RBC
CES	LB	004	4	WBC
CES	LB	004	5	RBC
CES	LB	004	6	WBC
CES	LB	005	1	RBC
CES	LB	005	2	WBC
CES	LB	005	3	RBC
CES	LB	005	4	WBC
CES	LB	005	5	RBC

where we indeed see that subject 002 has been excluded.

Normally, one should then take care that this "exclusion of subject 002" is added for each other variable XPath for that dataset definition. However, this is cumbersome and often even forgotten.

So we added a new feature allowing the system to inform the user when the XPath expression for any other variable is not allowed with that of the "topic variable". To enable this feature, one must however switch it on<sup>3</sup>. This can be done by using the menu "Options - Setting":

<sup>3</sup> This can be a good idea when one unexpectedly gets an empty column for a variable for which the mapping looks OK.

**?**

- ☐ Always hide upper panel in Mapping Script Editor
- ☐ View ODM Items without 'traffic lights'
- ☐ View ODM tree nodes without graying out mapped nodes
- ☐ View ODM tree with OIDs
- ☒ Assume define.xml and ODM files in same directory
- ☒ Allow mapping guidance from ODM annotations
- ☒ Jump to SDTM cell expected to be suitable for mapping
- ☐ Show Animated Icons for 'hot candidates'
- ☐ Hide sticky notes in SDTM/SEND cells
- ☒ Add default mapping descriptions from file 'default\_mapping\_descriptions.txt'
- ☐ Generate/Execute Mappings for user-selected domains/dataset-definitions only
- ☒ Allow to suggest adaptations to the XPath expression to be aligned with the XPath expression for the topic variable

and checking the checkbox "Allow to suggest adaptations to the XPath expression to be aligned with the XPath expression for the topic variable". Please remark that the default for this checkbox is "off".

When we then continue our mappings, and drag-and-drop again to e.g. populate LBORRES, we get the selection dialog again:

**?**

☒ Import XPath expression for ItemData **Value** attribute (from Clinical Data)

☐ Import XPath expression for **another ItemData attribute/subelement** (from Clinical Data)

ItemOID

☐ Import ItemDef attribute value (static value from Study Definition)

CollectionExceptionConditionOID

<input checked="" type="checkbox"/> Generalize for all StudyEvents	Except for ..	No Exceptions	Only for ..	2 Inclusions
<input type="checkbox"/> Generalize for all Forms	Except for ..	No Exceptions	Only for ..	No Inclusions
<input type="checkbox"/> Generalize for all ItemGroups	Except for ..	No Exceptions	Only for ..	No Inclusions
<input checked="" type="checkbox"/> Generalize for all Items	Except for ..	No Exceptions	Only for ..	2 Inclusions

ODM ItemDef Length: 8    SDTM Variable Length: 80

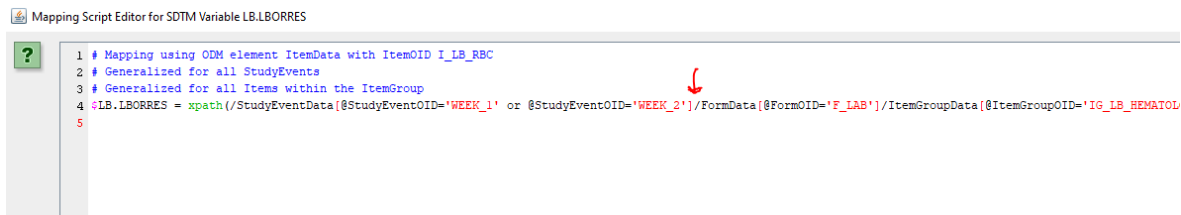
☐ Set SDTM Variable Length to ODM ItemDef Length

☒ View/Edit XPath expression (advanced)

OK Cancel



Especially in case there was a time span between mapping to LBTESTCD and to LBORRES (lunch, coffee/thee break, other day, ...) we will probably forget to check the checkbox "View/Edit XPath expression ...", and the generated mapping script will be:

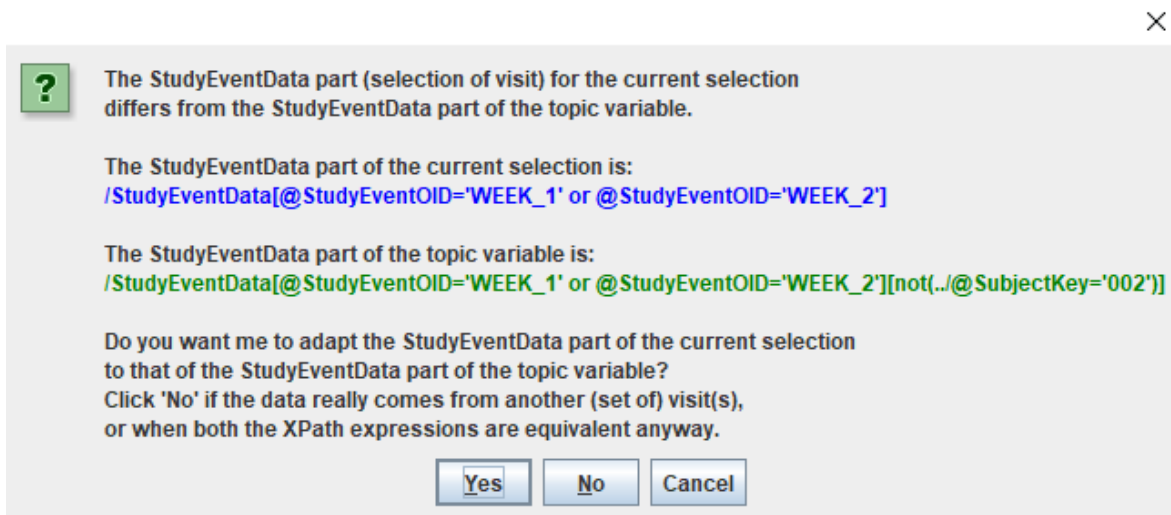


```

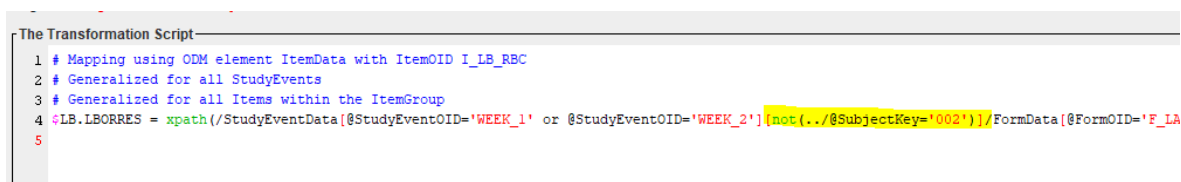
1 # Mapping using ODM element ItemData with ItemOID I_LB_RBC
2 # Generalized for all StudyEvents
3 # Generalized for all Items within the ItemGroup
4 LB.LBORRES = xpath(/StudyEventData[@StudyEventOID='WEEK_1' or @StudyEventOID='WEEK_2']/FormData[@FormOID='F_LAB']/ItemGroupData[@ItemGroupOID='IG_LB_HEMATOL
5

```

where we do not have the additional selection to exclude subject 002. If we would now continue, thinking everything is fine, it is possible that we come to surprises, like an empty LBORRES being produced. As we however have switched on the option to "suggest adaptations to the XPath expression ...", when the then click "OK" in the mapping editor, the system tells us:



suggesting us to have the XPath expression for LBORRES being adapted. If we click "Yes", and then click "Yes", and then double-click the LBORRES cell to edit it again, we see that the system has itself adapted the XPath expression to be aligned with that of the "looping variable" LBTESTCD. For LBORRES:

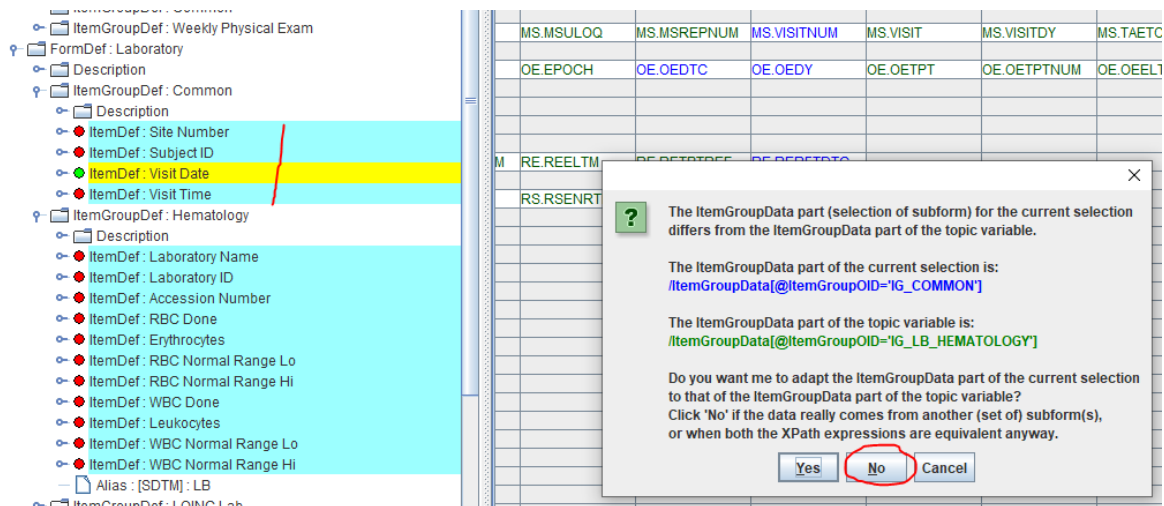


```

1 # Mapping using ODM element ItemData with ItemOID I_LB_RBC
2 # Generalized for all StudyEvents
3 # Generalized for all Items within the ItemGroup
4 LB.LBORRES = xpath(/StudyEventData[@StudyEventOID='WEEK_1' or @StudyEventOID='WEEK_2'][not(../@SubjectKey='002')]/FormData[@FormOID='F_LA
5

```

Remark that all this not only applies to the "StudyEventData" level, but can in principle also be applied to the "FormData" and "ItemGroupData" level. That will however be very seldom. It is also important not to just automatically click "Yes" in the above dialog: think first! In cases that the data point to be selected does not come from the same form or subform, this may lead to an incorrect mapping. For example, when the "collection date" comes from another sub-form (ODM "ItemGroup"), one will not want to have the XPath expression being adapted on the level of "ItemGroupDef" to align with that of the topic variable:

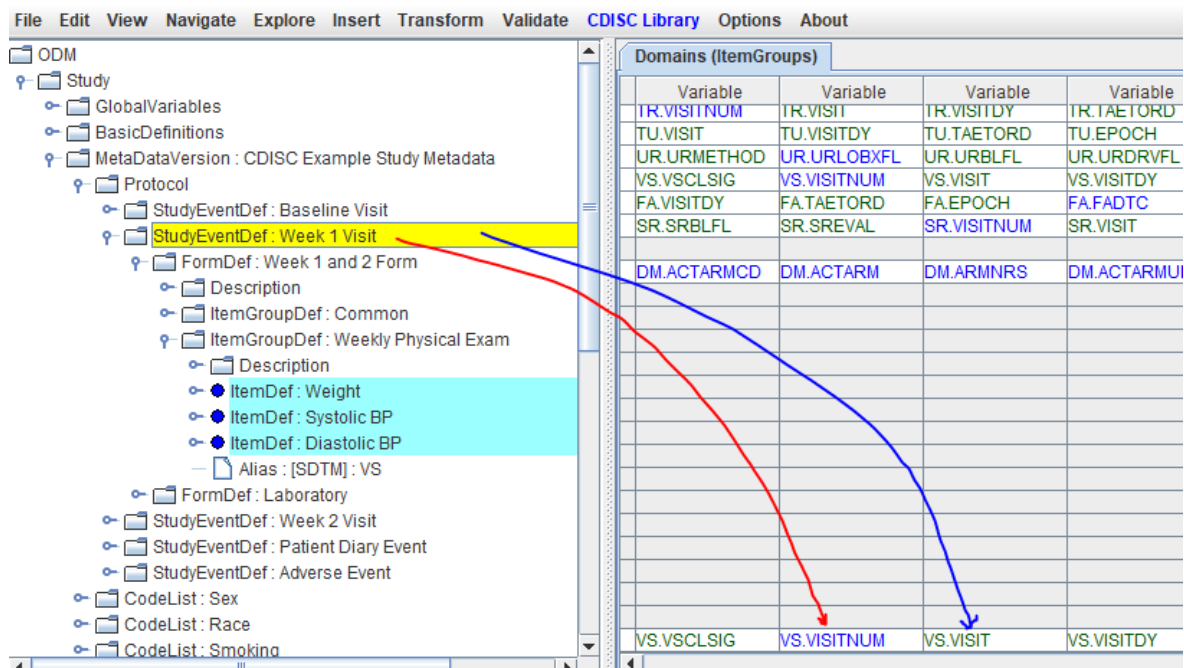


This new feature is very powerful, but may also lead to confusion, so it may be a good idea to only use it when one has such "very special" selections, and only for specific variables like -ORRES. Normally, when one has well-organized data in the ODM file, one will not need this feature at all.

P.S. This feature can be combined with the "xpathfilter" function for the "looping variable". The "xpathfilter" function is described in the tutorial ["Filtering on looping variables"](#).

## Automatically generate relative XPath for VISITNUM and VISIT variables

In most cases, for "Interventions", "Events" and "Findings" domains / dataset-definitions, the mapping will use the value of the OID (identifier) of the ODM "StudyEvent", as that typically represents the visit. So, one will usually start from a "drag-and-drop" from "StudyEventDef" in the ODM tree, like:



and then, using the OID, generate values for "VISITNUM" and "VISIT" from it, often using an if-elsif-else structure.

Usually, this is straightforward, but may go wrong when the topic variable (--TRT, --TERM, --TESTCD) has a very complicated XPath expression. In such a case, using a "relative XPath" will solve the problem in 99.99% of the cases. If the topic variable was mapped from information from the "Item" level, which is usually the case, then one can use the "relative XPath" for starting the mapping of VISITNUM or VISIT using e.g.:

```
$VISITOID = xpath('../..../@StudyEventOID);
```

meaning: "from the Item level, go up three levels, which results in the StudyEvent level, and from that, simply take the value of the "StudyEventOID" attribute".

Some users however are not very familiar with XPath, and asked for an automated solution, when they have such a complicated XPath expression for the topic variable, and the default procedure, just doing drag-and-drop, doesn't lead to a result (typically an empty VISITNUM / VISIT column).

In order to allow this feature to be applied, use the menu "Option - Properties", and look for the checkbox "Allow to suggest the simplified relative XPath expression '../..../@StudyEventOID' for VISITNUM and VISIT variables":

☐ Hide sticky notes in SDTM/SEND cells  
☒ Add default mapping descriptions from file 'default\_mapping\_descriptions.txt'  
☐ Generate/Execute Mappings for user-selected domains/dataset-definitions only  
☐ Allow to suggest adaptations to the XPath expression to be aligned with the XPath expression for the topic variable  
☒ Allow to suggest to use the simplified relative XPath expression **"../../../../@StudyEventOID" for VISITNUM and VISIT variables**  
☐ Add progress messages to XSLT  
☐ Skip display of generated XSLT

and then check this checkbox (the default of it is "off").

Once again: best is to use this feature only on the case one has trouble generating VISITNUM/VISIT using the normal procedure.

When one then does a drag-and-drop from "StudyEventDef" in the ODM tree, one will first get the usual selection panel:

Import StudyEventDef: WEEK\_1 - for SDTM Variable VS.VISITNUM

☒ Import XPath expression for  
☐ Import attribute value (static value) for

OID

☒ Generalize for all StudyEvents    Except for ..    No Exceptions    Only for ..    No Inclusions

☐ View/Edit XPath expression (advanced)

OK Cancel

and depending on the circumstances, also check "Generalize for all StudyEvents", and sometimes using "Except for ..." or "Only for ...". In the next step, a new dialog will then show up:

Simplified XPath

☒ Do you want me to use the simplified XPath expression **../../../../@StudyEventOID** as the base for the mapping for variable **VS.VISITNUM**?

Yes No Cancel

If the user then clicks "Yes", the resulting mapping script will then e.g. be:

The Transformation Script

```

1 # Mapping using ODM element StudyEventData using value from attribute StudyEventOID
2 # Generalized for all StudyEvents
3 $VS.VISITNUM = xpath('../../../../@StudyEventOID');
4

```

but as the OID is not a number, needs to be further developed, e.g. as:

```
The Transformation Script
1 # Mapping using ODM element StudyEventData using value from attribute StudyEventOID
2 # Generalized for all StudyEvents
3 $VISITOID = xpath('../../../../@StudyEventOID');
4 if($VISITOID = 'BASELINE') {
5     $VS.VISITNUM = 0;
6 } elseif($VISITOID = 'WEEK_1') {
7     $VS.VISITNUM = 1;
8 } elseif($VISITOID = 'WEEK_2') {
9     $VS.VISITNUM = 2;
10 } else {
11     $VS.VISITNUM = -999;
12 }
```

with the "-999" assignment just as a "backup" for unexpected cases.

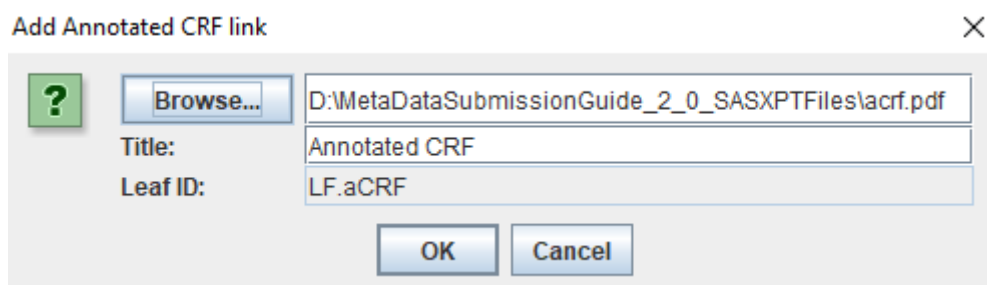
If in the prior dialog, the user has clicked "No" or "Cancel", the usual equivalent XPath expression is just generated:

```
The Transformation Script
1 # Mapping using ODM element StudyEventData using value from attribute StudyEventOID
2 # Generalized for all StudyEvents
3 $VS.VISITNUM = xpath(/StudyEventData/@StudyEventOID/);
4
```

It is principally equivalent, as during execution, all "absolute" XPath expressions (those that start with a "/") will automatically be transformed into a relative XPath expression. As stated, this may go wrong in very special cases, so using "../../../../@StudyEventOID" then is the safer option.

## Extended methods for automated retrieval of page numbers from aCRFs

Already for a longer time, we added the feature to automatically retrieve page numbers from an annotated CRF (aCRF) for use in the assignment in variables with source type "Collected". This requires loading a link to the aCRF first, using the menu "Insert - Link to Annotated CRF". For example:



Add Annotated CRF link

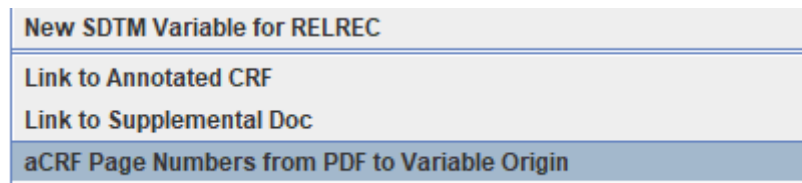
? Browse... D:\MetaDataSubmissionGuide\_2\_0\_SASXPTFiles\acrf.pdf

Title: Annotated CRF

Leaf ID: LF.aCRF

OK Cancel

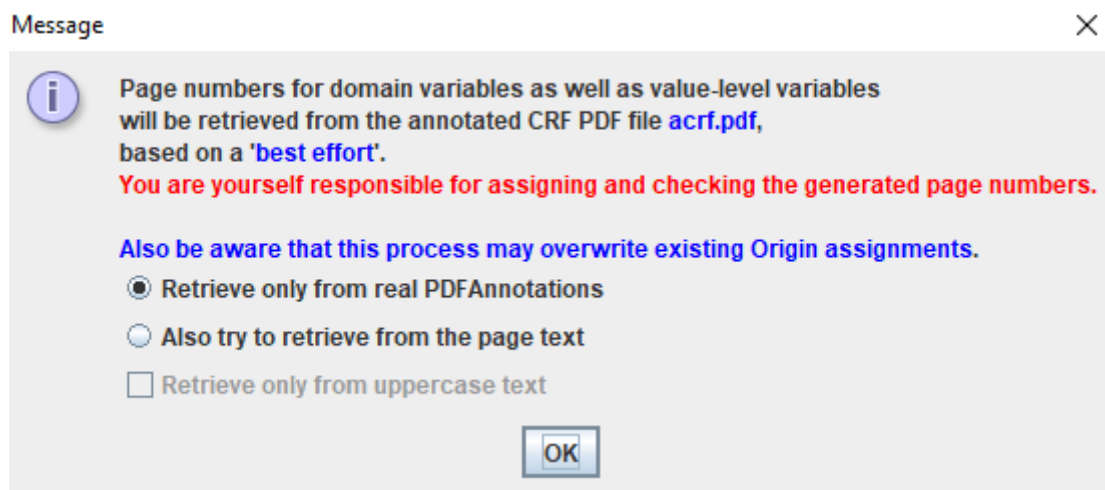
followed by using the menu "Insert - aCRF Page Numbers from PDF to Variable Origin":



We noticed that this does not work for all types of aCRF PDFs. For some it does, for some it doesn't. For example, for the aCRF coming from the CDISC "Metadata Submission Guide 2.0", it did not work. The reason is that there is no single, unique way to generate aCRF PDFs, and the way it is done may well be company-specific.


Therefore we extended the code for the page number retrieval for a number of aCRF-PDFs that we had to our disposal.

When the user then requests to automatically retrieve page numbers for the variables of source type "Collected", the following dialog is displayed:



For the second choice "Also try to retrieve from the page text", one can limit to those pieces of text that are full-uppercase, as SDTM annotations are usually uppercase. This will then avoid that also the question texts are taken into account. For the latter choice, after about 1-2 minutes, depending on the file size of the aCRF (one can follow the progress in the console), then e.g. leads to:

Select for which Items an Origin of Type 'CRF' with page numbers need to be cre... X

 Please select the Items for which an Origin of type 'CRF', containing a page number or page numbers, need to be created. Items in **red** do currently already have an Origin assigned. You can see the Origin assignment by hovering over the Item (tooltip).

<input type="checkbox"/>	Item: <b>BRTHDTC</b> [DM.BRTHDTC] pages = 5 Annotation = BRTHDTC
<input type="checkbox"/>	Item: <b>AGE</b> [DM.AGE] pages = 5 Annotation = AGE
<input type="checkbox"/>	Item: <b>AGEU</b> [DM.AGEU] page = 5 Annotation = AGEU
<input type="checkbox"/>	Item: <b>SEX</b> [DM.SEX] pages = 5 Annotation = SEX
<input type="checkbox"/>	Item: <b>RACE</b> [DM.RACE] pages = 5 Annotation = RACE
<input type="checkbox"/>	Item: <b>VISIT</b> [LB.VISIT] pages = (8,9,10,12,21) Annotation = VISIT
<input type="checkbox"/>	Item: <b>VSPOS</b> [VS.VSPOS] pages = (8,9,10,11,12,13) Annotation = VSPOS
<input type="checkbox"/>	Item: <b>VSORRES</b> [VS.VSORRES] pages = (8,9,10,11,12,13) Annotation = VSORRES
<input type="checkbox"/>	Item: <b>VSORRESU</b> [VS.VSORRESU] pages = (8,9,10,11,12,13) Annotation = VSORRESU
<input type="checkbox"/>	Item: <b>VISIT</b> [VS.VISIT] pages = (8,9,10,12,21) Annotation = VISIT

from which one can then make the selection. Items in blue have not been assigned "Collected" as "Origin" yet, but when selected, will get the correct annotation and page number the following time the properties of the variable is taken care of. For example for "AGEU":

**Origin type:**

☐ Not Available

☐ Assigned

☐ Protocol

☐ Derived

☐ Predecessor

☒ Collected

**Source type:**

☒ Subject

☒ Investigator

☐ Vendor

☐ Sponsor

Origin description

Document (leaf) ID:

LF.aCRF

☐ No page details (yet)

☒ Page list (physical reference)

☐ Named destinations

Page list / List of named destinations

5

☐ Page range: first page - last page

As the system cannot know who filled the CRF (it could also be a Personal Report Outcome - PRO), one then still needs to check the appropriate radiobutton under "Source Type".

We recognize that there are very many ways to produce aCRFs in PDF format<sup>4</sup>, so if you see that your way of annotating CRFs as PDF is not covered, please reach out to use, so that we can extend the software for it as an additional service.

## Features specifically for Apple Macintosh

Drag-and-drop (DnD) works rather different on Apple-Java: upon starting the "drag", MacOS replaces the "drag"-icon with its own default, which is a rectangle. Also, when starting a "drag" from the tree on the left, it requires that the "content" of the tree item is written to file, which then needs to be picked up by the table on the right.

This required that a good part of the "drag-and-drop" code needed to be rewritten to distinguish between the two cases.

When the software starts, it automatically finds out whether the operating system is a MacOS version, or is Windows or Linux, so that during "drag-and-drop", the correct code is executed for the applicable operating system.

Mac users can however override this by adding a line in the "properties.dat" file:

<sup>4</sup> The better way would be that regulatory authorities require a real electronic aCRF, e.g. in the form of the ODM file with the study metadata, containing all the mapping targets as "SDTM Alias" annotations.



```
odmdefineinsamefolder=true  
# Addition v.5.2 2026-01-14: whether on Mac, default Java behavior should be used  
usedefaultdraganddroponmac=true
```

when "usedefaultdraganddroponmac" is set to "true", the system will use the default Java behavior, which is much better tested, and also often used on Mac anyway.

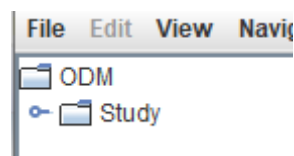
The default for this parameter is "false", so on Mac, it will then try to use the new code especially created for Mac.

This parameter is ignored by the system on Windows and Linux computers.

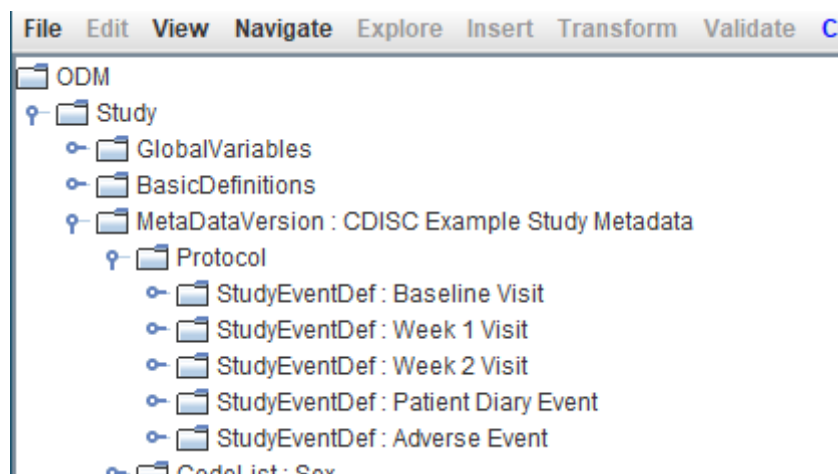
## Other minor updates

### Initial ODM tree display

A few users found the initial display of the "ODM tree" on the left side not very intuitive, as to start working, one first needs to expand the "Study" node in the tree.



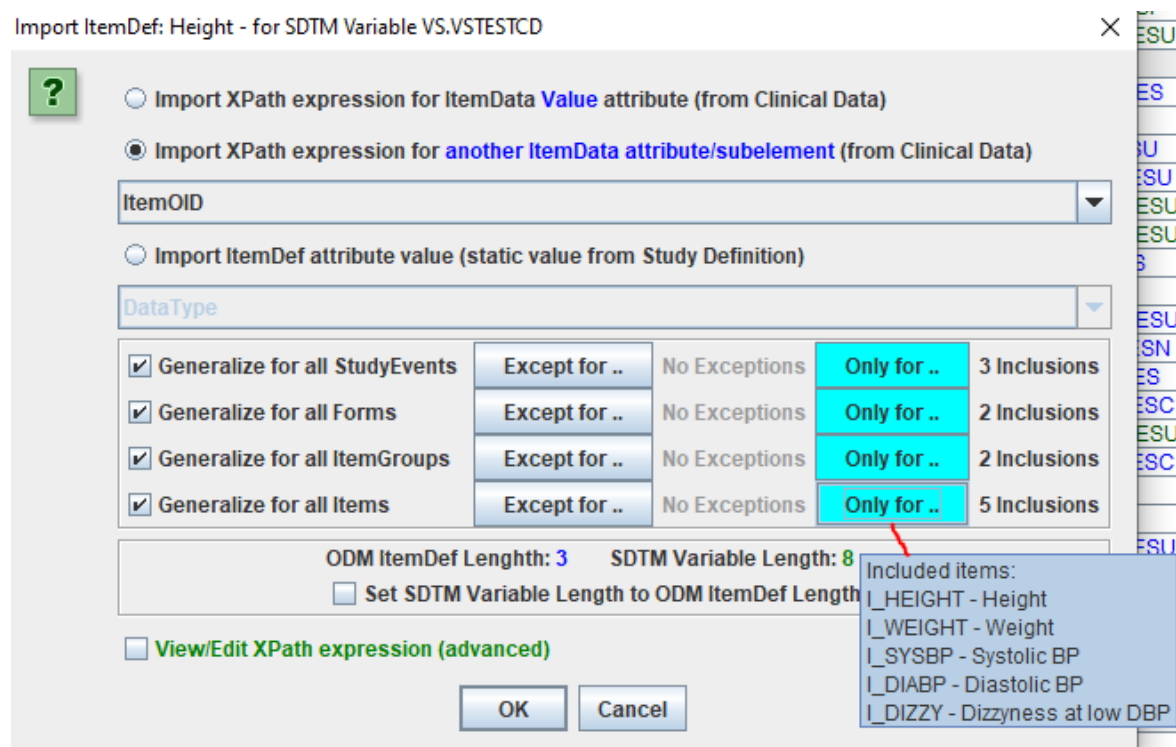
Therefore, we changed the initial view, immediately after loading, to one where all visits (ODM "StudyEvent") are visible by expanding the tree to them:



During loading the template file for a specific version of SDTM or SEND, one can then start navigating for specific items (using the "Navigate" menu), inspect their codelist (when any) and the clinical data for them etc..

### Tooltips on "Except for ..." and "Only for ..." buttons

Especially when one does not do all the mappings for a single domain in a single run, it may be useful to quickly see what the prior selections on visit, form, subform and items for that domain mapping were. This could already be done by clicking the "Only for ..." or "Except for ..." button (again), but also obtaining the information when simply hovering the mouse over the button (or the label adjacent to it) is also very useful. So we added tooltips to these. They are automatically updated when the user changes something in the selection. For example:



## Bug fixes

### Identifier variables in generated SQAPxx datasets

When having an "Associated Persons" dataset like APMH (Associated Persons Medical History) with one or more "non-standard variables", and selecting the checkbox "Move non-standard variables to SUPP--", a SQAPxx<sup>5</sup> (e.g. SQAPMH) is automatically generated. In such cases, the identifier in the SQAPxx dataset was incorrectly provided as "USUBJID", whereas it must be "APID" with the label "Associated Persons Identifier". This has now been corrected.

It has now also been implemented that in the case of SAS-XPT, the APID variable in the SQAPxx dataset is assigned the same length as in the parent APxx dataset.

<sup>5</sup> The reason such datasets are not getting a name like "SUPPAPMH" is that SAS-XPT does not allow dataset names to be longer than 8 characters. If one would then need to have a "split dataset" like SUPPAPMHRE, the dataset name would exceed this limit. So the prefix "SUPP" has been replaced by "SQ" for this kind of datasets.

## **Automated VISITNUM generation for "unscheduled" visits that do not have the --DTC variable present**

For most of the Events and Intervention datasets, there is no --DTC (Date/Time of Collection) variable, but there is a --STDTC (Start Date/Time) variable. Some of them however can have both. Thus caused problems in the automated assignment of VISITNUM for an unscheduled visit when no --DTC variable could be found or was not populated.

The algorithm has now been changed in such a way that for these classes, the system will always first try to use --DTC (Date/Time of Collection) and if it is not found, or not populated, it will try to use the value of --STDTC (Start Date/Time).

## **Other small fixes**

## **Other remarks**

These new features have only partially been tested with other operating systems than Windows, such as on Linux. This is work in progress.

Several of the new features have not been tested yet with batch execution.