

SDTM-ETL 4.4 User Manual and Tutorial

Author: Jozef Aerts, XML4Pharma

Last update: 2024-12-10

Ensuring Mapping Completeness

Table of Contents

Table of Contents	1
Introduction.....	1
"Graying out" used items from the ODM	1
"Highlighting" of SDTM/SEND cells.....	6
Using the menu "View - Item Usage in Mappings"	7
Using the menu "View - List Items not used in Mappings"	8
Using the "Mapping Completeness Report" for training AI systems	12
Basic information when executing mappings	12
Mapping Completeness for Hypervetical ODM Structures	13
Checking completeness on the SDTM side	16
Conclusion	17

Introduction

Ensuring mapping completeness, i.e. ensuring that all collected (and derived) data that needs to go into the SDTM or SEND, is always a challenge.

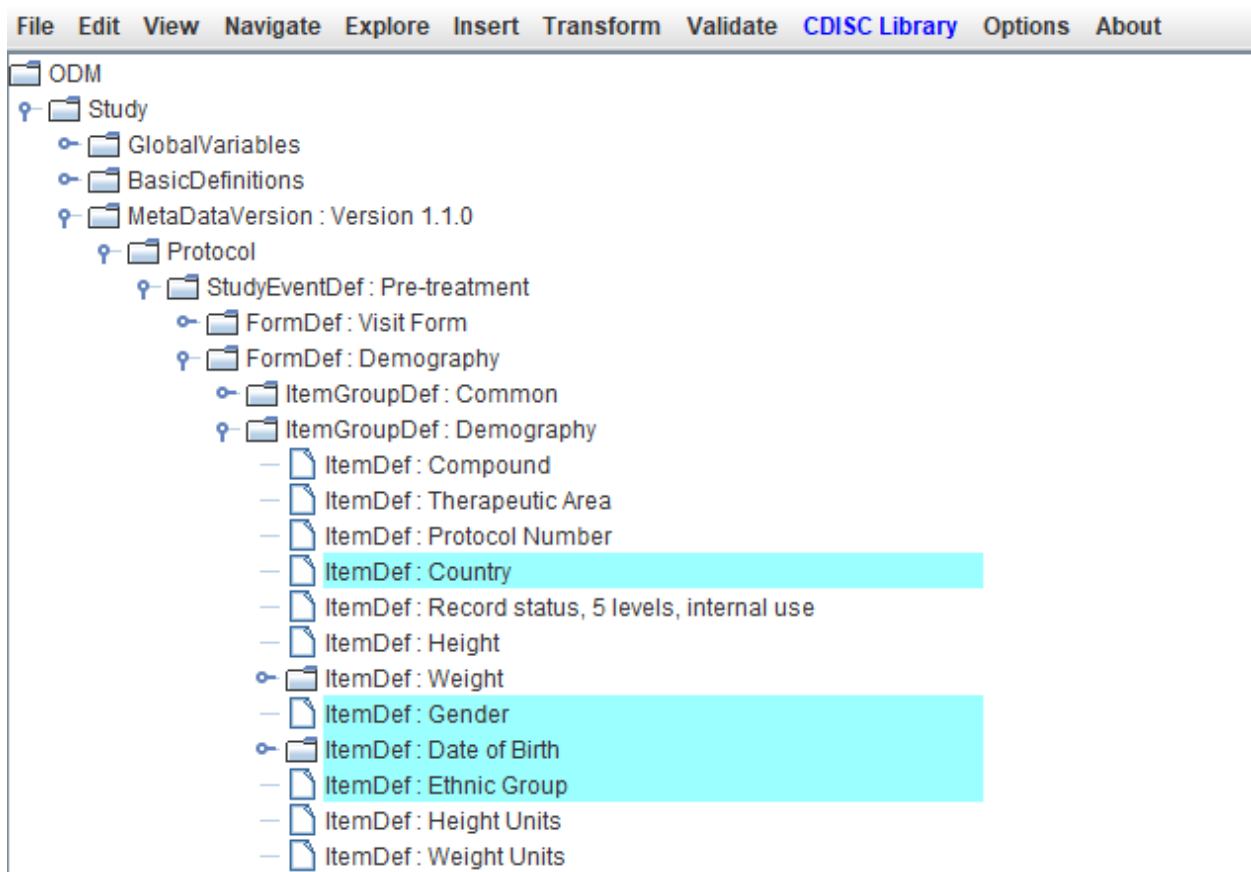
The danger is always that some subjects or visit have been forgotten or skipped, or that some measurements did not make it into the findings datasets. Another typical ones is that measurements for which no results have been obtained, and/or the reason why the measurement did not have results.

The other way around, one may have that one has SDTM/SEND domains and variables for which, based on the input data, one should have mappings for, these domains have not been instantiated as dataset definitions, and/or the variables haven't been provided with a mapping.

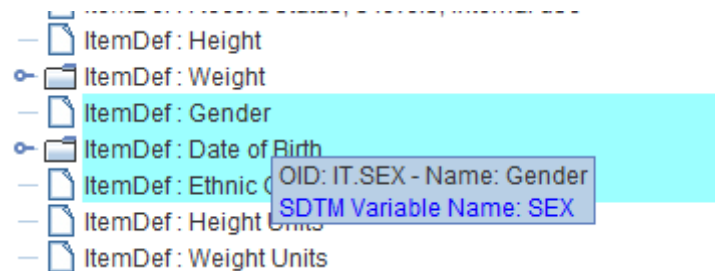
Although SDTM cannot fully guarantee mapping completeness, it has a number of tools allowing the user to check for mapping completeness, and take action when necessary.

"Graying out" used items from the ODM

The loaded ODM defines the metadata of the study, and especially the CRFs (in the case of human clinical trials). For example:



with the items (CRF questions) with a cyan background color, having an "**SDTM annotation**" in the ODM, as very often for items in CDASH forms. The SDTM annotation can be seen when hovering the mouse over the item, for example:



or by using the menu "View - All Item Details", e.g. leading to:



Details for ItemDef with OID IT.SEX with Name 'Gender'

Data type	Mandatory?	Max. Length	Max. Characters after decimal point	SASFieldName	Mapping to SDTM variable
text	No	6		SEX	SEX

CodeList with OID CL.SEXF with Name 'Gender'

Data type: text

Allowed values (code and decode)

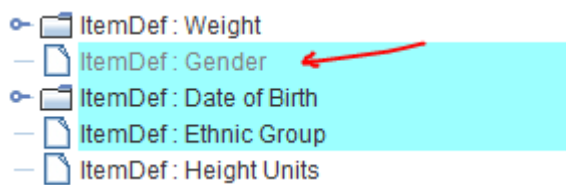
Coded value	NCI Code	Decode language	Decode
0		en	Female
1		en	Male

When then clicking on "Gender" and DM has been loaded, the "SEX" SDTM cell will automatically be highlighted and even selected:

	DM.AGEU	DM.SEX	DM.RACE	DM.ETHNIC

easing drag-and-drop from ODM "Gender" to SDTM DM.SEX¹.

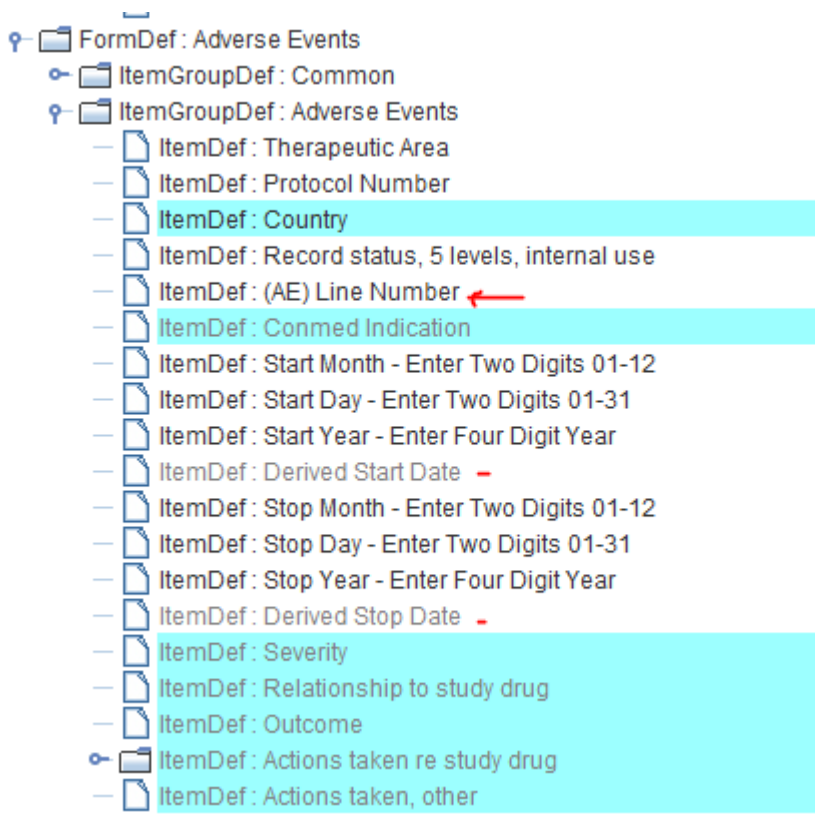
When then having mapped "Gender" to SDTM DM.SEX, one will see that the foreground color changes from black to gray:



i.e. the item "Gender" has been "grayed out", as has being used in an SDTM mapping.

When the mapping has then been advanced, one may e.g. see:

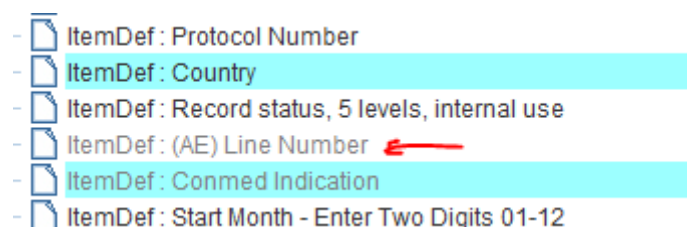
¹ We decided not to automate the drag-and-drop in such a case, as we still want the user to be in charge, and taking all the decisions.



where one sees that "(AE) line number" has not been used in a mapping yet. However, it does not have an SDTM annotation where it should go into in SDTM. Typical usage of "line number" in SDTM is AESPID (Sponsor-defined Identifier), as is also explained in the SDTM-IG:

AEREFID	Reference ID	Char	Identifier	Single domain for a subject.	Perm
AESPID	Sponsor-Defined Identifier	Char	Identifier	Internal or external identifier such as a serial number on an SAE reporting form.	Perm
AFTERM	Reported Term for	Char	Tonic	Sponsor-defined identifier. It may be preprinted on the CRF as an explicit line identifier or defined in the sponsor's operational database. <u>Example: Line number on an Adverse Events page.</u>	Ren

So, when one then drags-and-drops "(AE) Line Number" to the SDTM cell "AESPID" and confirms the mapping, also "(AE) line number" will be "grayed out"²:



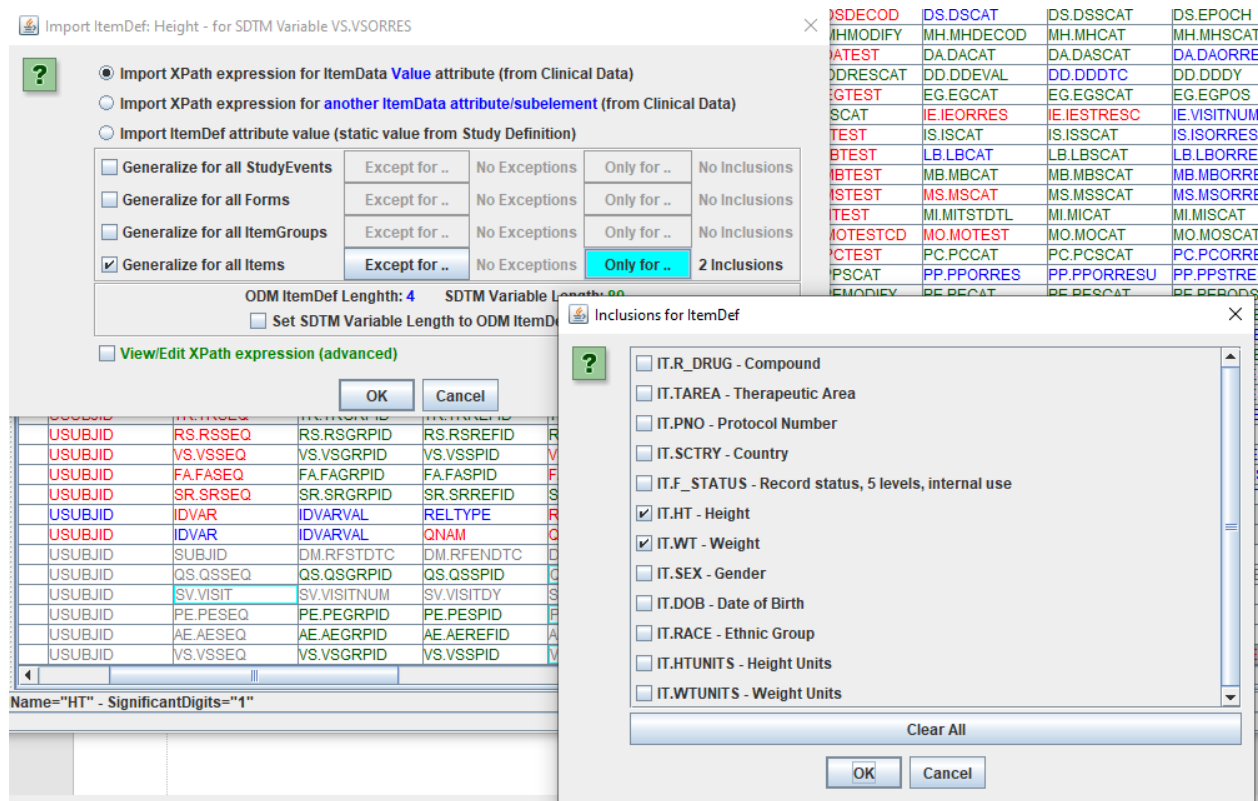
In the example, one also sees that "Start Month...", "Start Day...", "Start Year..." and similar for "End Month..." have not been used. However, "Derived Start Date" and "Derived End Date" have been used. So it is not always so that an item from the ODM must necessarily flow into SDTM. Typical examples are so-called "cleaning aids" like "Did any adverse event occur".

² Remember that "line number", going into AESPID is often use to connect concomitant medications to an adverse event that caused the medication to be given.

This "graying out" mechanism also works when using multiple items in drag-and-drop, the so-called "generalizations". When for example doing a drag-and-drop for "Height"

- ● ItemDef : Height
- ● ItemDef : Weight
- ● ItemDef : Gender
- ● ItemDef : Date of Birth
- ● ItemDef : Ethnic Group

to the cell "VSORRES", and then using the "Generalization" to also include "Weight":



and then doing accepting the mapping, both "Height" and "Weight" will be "grayed out" in the ODM tree:

- ● ItemDef : Record status, 5 levels, internal use
- ● ItemDef : Height
- ● ItemDef : Weight

P.S. Sometimes, this "graying out" may not immediately become visible, and needs another selection with the mouse. One can however also use the menu "View - Recalculate ODM tree nodes usage in mapping":

View	Navigate	Explore	Insert	Transform	Validate	CDISC Lib
ODM tree with Names						
ODM tree with OIDs						
ODM Items with 'traffic lights'						
ODM Items without 'traffic lights'						
ODM tree nodes with 'graying out' mapped nodes						
ODM tree nodes without 'graying out' mapped nodes						
Recalculate ODM tree nodes usage in mapping (graying out)						

Remark: currently, "graying out" only partially works when doing a "negative selection" during the "generalization", e.g. "exclude all items in the group except for 'Height' and 'Weight'". So, for the moment, we recommend to use "positive selections" (i.e. using "Only for ...") as much as possible.

"Highlighting" of SDTM/SEND cells

Once mapped, when selecting an Item in the ODM tree, the system looks up whether the item is used in any of the mappings, and if so, "highlights" the SDTM/SEND cells in which the Item of the ODM tree is used. For example, when clicking "Diastolic BP", the cells "VSTESTCD" and "VSORRES" light up:

The screenshot shows the CDISC Lib interface. On the left, the ODM tree is expanded to 'ItemDef: Diastolic BP'. On the right, a grid of SDTM/SEND cells is displayed. The cells 'VSTESTCD' and 'VSORRES' are highlighted in yellow, indicating they are used in the mapping for the selected item.

Remark that when the "decode()" function has been used for VSTEST, like:

```

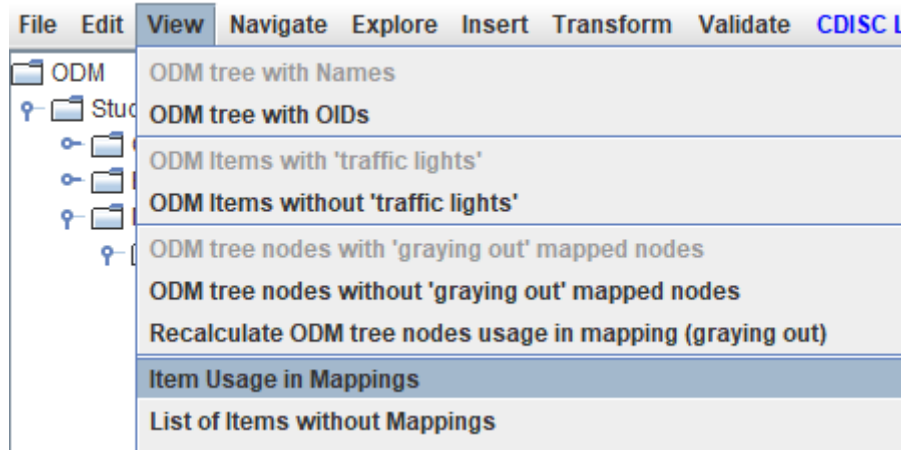
The Transformation Script
1 # Mapping using the decode() function on codelist CL.C66741.VSTESTCD of variable VS.VSTESTCD
2 $VS.VSTEST = decode($VS.VSTESTCD, 'CL.C66741.VSTESTCD', '');

```

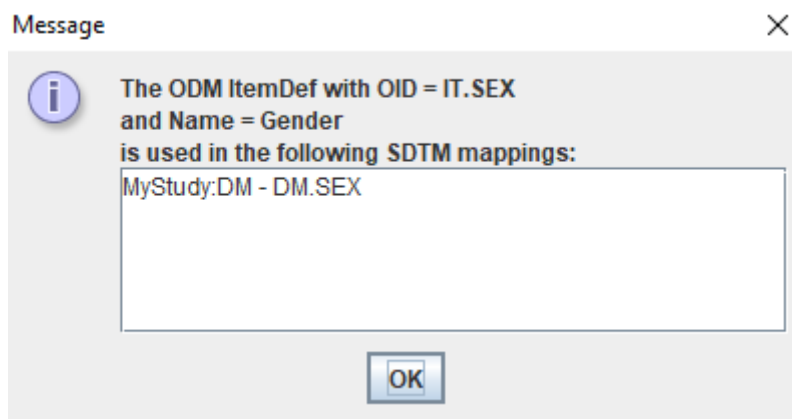
the VSTEST cell will not light up, as there is no "xpath(...)" statement in the mapping script that points back to the ODM tree node.

Using the menu "View - Item Usage in Mappings"

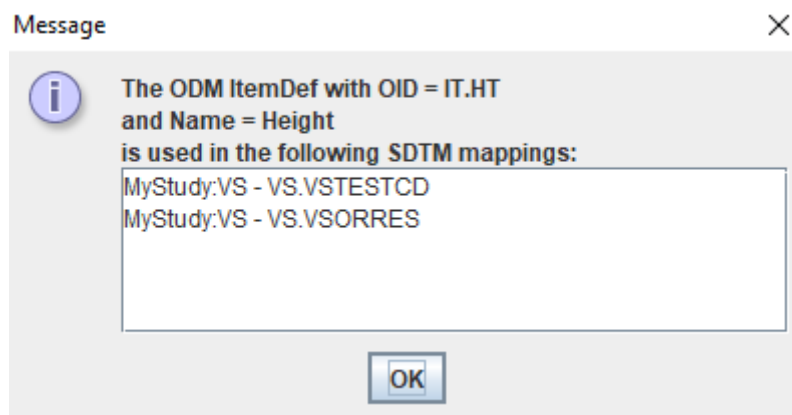
If one wants to find out whether an ODM item (corresponding to a question on the CRF in the case of human clinical studies mapped to SDTM), one can use the menu "View - Item Usage in Mappings" after having selected an Item in the ODM tree.



This e.g. leads to:



It can of course be that a list is provided when the Item has been used in mappings to different SDTM variables, e.g.:



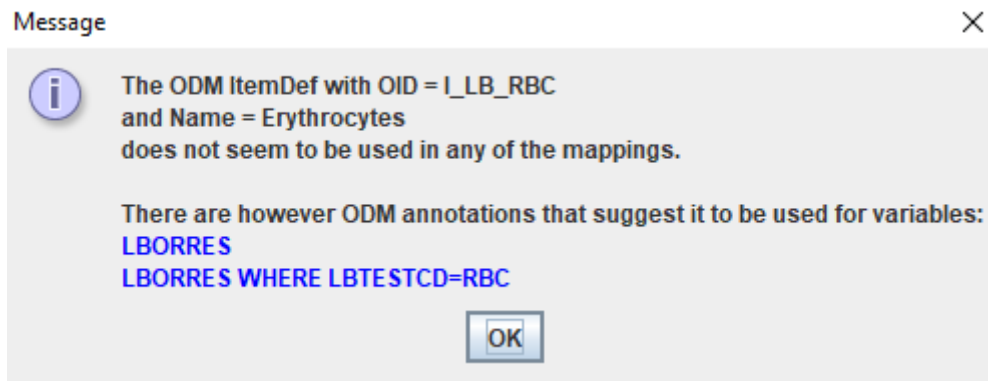
A limitation here still is that when one "reuses" the mapping in another variable, it will not be listed here. For example, when for VSSTRESC (standardized result) one has:

```
The Transformation Script
1 $VS.VSSTRESC = $VS.VSORRES;
```

VSSTRESC will not appear in the list from "Item Usage in Mappings"³.

When no mapping is found the message shown is:

When no mapping is found, but the ODM item has one or more annotations (using the "SDSVarName" attribute and/or "Alias/@Context='SDTM'" child element), the message shown is e.g.:

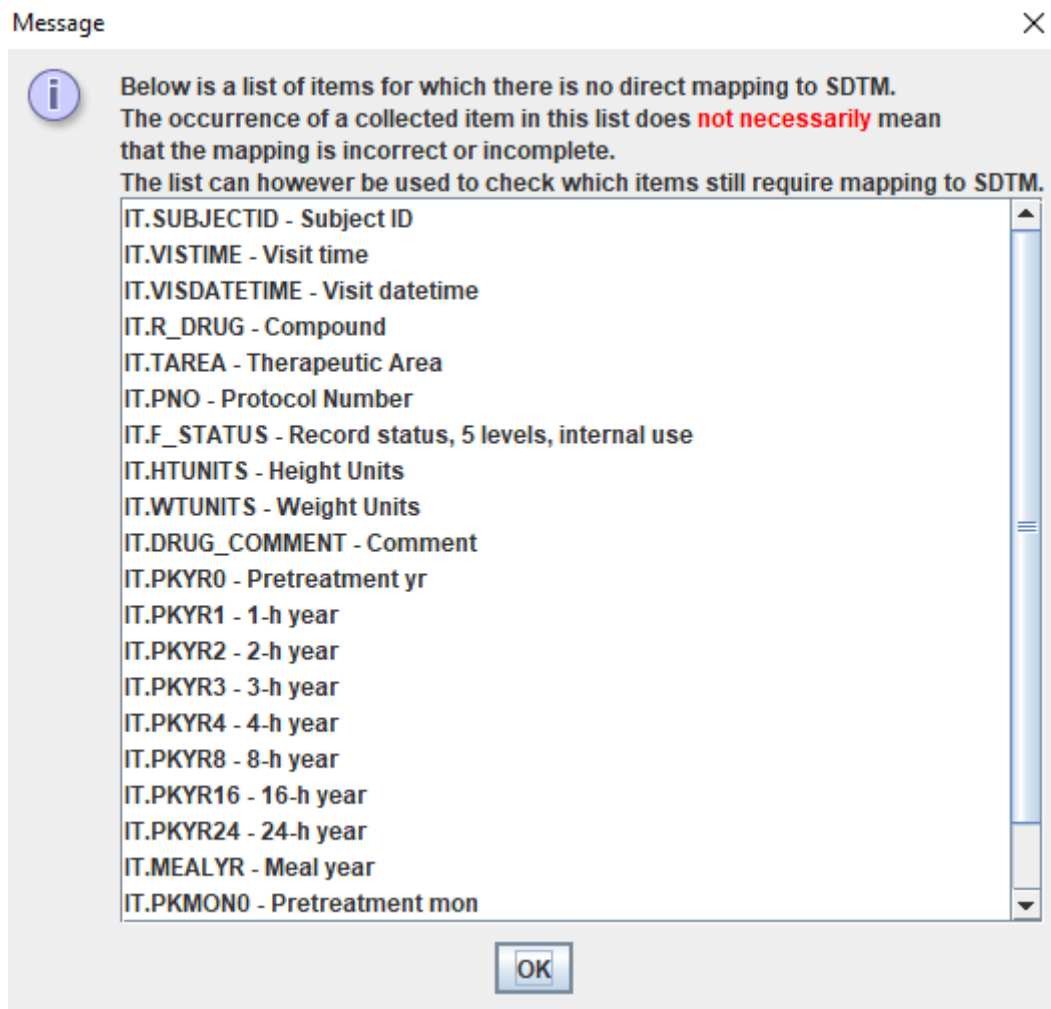


Using the menu "View - List Items not used in Mappings"

This feature essentially just works the other way around as "View - Item Usage in Mappings".

It e.g. leads to the list:

³ The reason is that the algorithm is based on the presence of an "xpath(...)" statement in the mapping script.

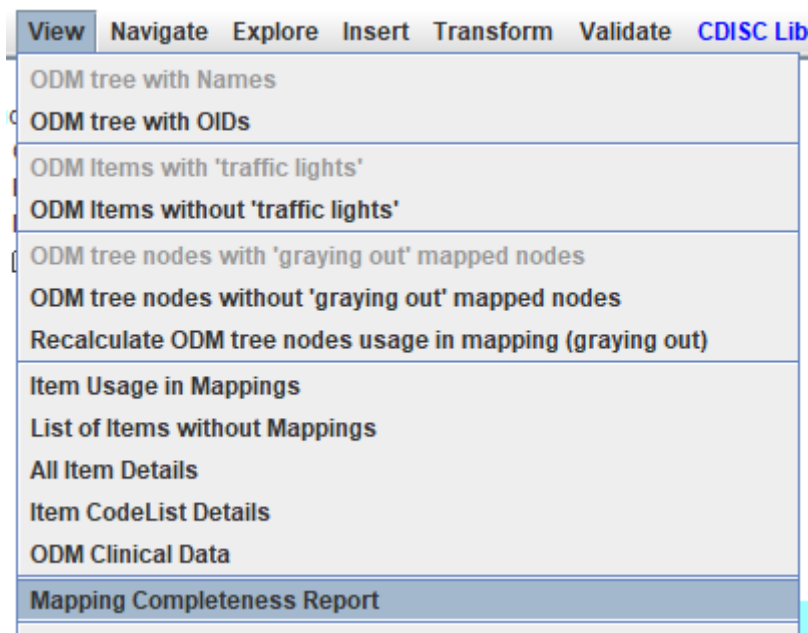


Being in the list does however not mean that the mappings are incomplete or incorrect. For example, we see that "Subject ID" is in the list, but we used the function "usubjid()" in the mappings to retrieve the value of the "SubjectKey" in the ODM clinical data to populate USUBJID. Also, we see that "Height Units" and "Weight Units" is in the list, but these may have been hard-coded, or have been retrieved from the ODM "MeasurementUnit" in the clinical data.

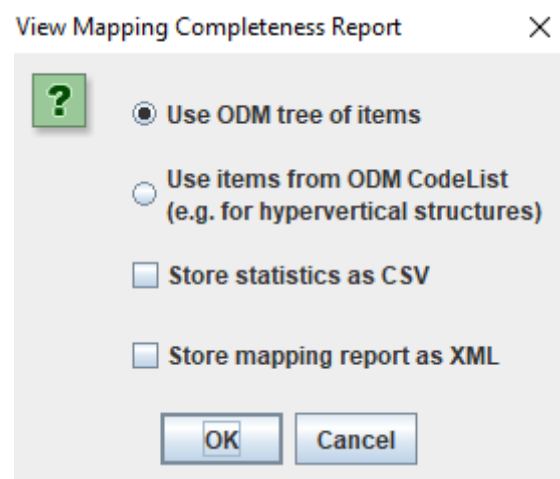
However, it is still always a good idea to go over the list, and ask oneself "do I need this in a mapping, or did I already cover it in another way?"

Generating a full Mapping Completeness Report

Another way to get a good overview of what is already completed and what still needs to be done, is to generate and display a "mapping completeness report", which can be achieved through the menu "View - Mapping Completeness Report":



The user is then asked:



We will explain the choice "Use items from ODM CodeList (e.g. for hypervertical structures) later, in the section "Mapping Completeness for Hypervertical ODM Structures".

The feature "Store statistics as CSV" can e.g. be used to import in an Excel worksheet for discussion with the team. Similarly, the mapping report can be stored as XML for further processing.

The result will e.g. be:

Using the "Mapping Completeness Report" for training AI systems

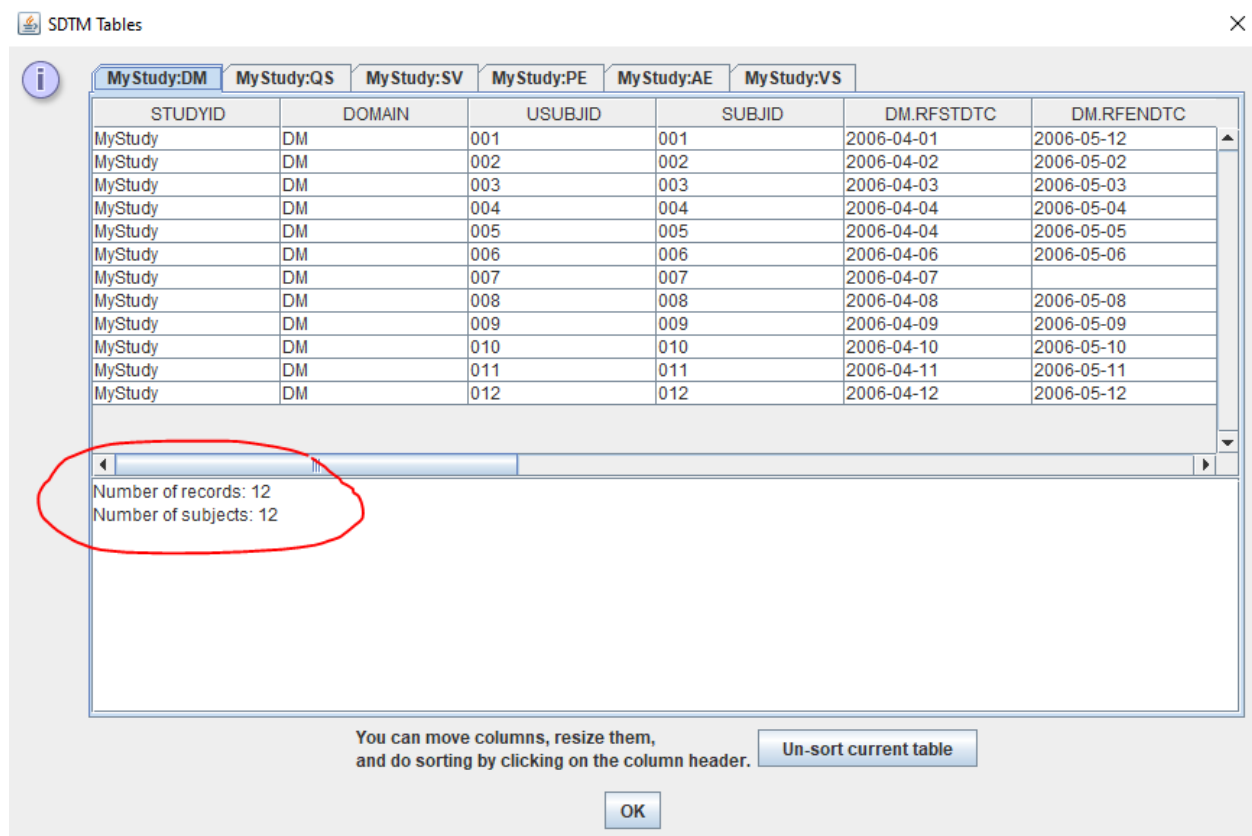
As one can also save the "mapping completeness report" as machine readable XML (metadata + data), it is extremely suited for training AI systems. When one has such a mapping completeness report in XML for several studies, these can be used as the basis for setting up an AI system that can help automating the mapping of new studies.

SDTM-ETL already has an interface with ChatGPT, but the quality of the knowledge of ChatGPT about SDTM (and of SEND) currently is still pretty low. Therefore, we expect that AI systems being trained specifically for SDTM (later maybe also for SEND) are the future. We can then also provide an interface with such AI systems as an add-on.

Basic information when executing mappings

New, and somewhat experimental in SDTM-ETL v.4.4 is that when executing the mappings, one also get basic information about the dataset generated that can be used to check whether the mapping, regarding the scope, is complete.

For example when executing the mappings for our example, one may get for DM:



SDTM Tables

STUDYID	DOMAIN	USUBJID	SUBJID	DM.RFSTDTC	DM.RFENDTC
MyStudy	DM	001	001	2006-04-01	2006-05-12
MyStudy	DM	002	002	2006-04-02	2006-05-02
MyStudy	DM	003	003	2006-04-03	2006-05-03
MyStudy	DM	004	004	2006-04-04	2006-05-04
MyStudy	DM	005	005	2006-04-04	2006-05-05
MyStudy	DM	006	006	2006-04-06	2006-05-06
MyStudy	DM	007	007	2006-04-07	
MyStudy	DM	008	008	2006-04-08	2006-05-08
MyStudy	DM	009	009	2006-04-09	2006-05-09
MyStudy	DM	010	010	2006-04-10	2006-05-10
MyStudy	DM	011	011	2006-04-11	2006-05-11
MyStudy	DM	012	012	2006-04-12	2006-05-12

Number of records: 12
Number of subjects: 12

You can move columns, resize them, and do sorting by clicking on the column header. Un-sort current table

OK

providing information about the number of records and subjects.
And for VS:

SDTM Tables

STUDYID	DOMAIN	USUBJID	VS.VSSEQ	VS.VSTESTCD	VS.VSTEST
MyStudy	VS	001	1	HEIGHT	Height
MyStudy	VS	001	2	WEIGHT	Weight
MyStudy	VS	002	1	HEIGHT	Height
MyStudy	VS	002	2	WEIGHT	Weight
MyStudy	VS	003	1	HEIGHT	Height
MyStudy	VS	003	2	WEIGHT	Weight
MyStudy	VS	004	1	HEIGHT	Height
MyStudy	VS	004	2	WEIGHT	Weight
MyStudy	VS	005	1	HEIGHT	Height
MyStudy	VS	005	2	WEIGHT	Weight
MyStudy	VS	006	1	HEIGHT	Height
MyStudy	VS	006	2	WEIGHT	Weight
MyStudy	VS	007	1	HEIGHT	Height
MyStudy	VS	007	2	WEIGHT	Weight

Number of records: 24
 Number of subjects: 12
 Number of visits: 1
 Number of distinct tests: 2
 Earliest value of VSDTC: 2006-04-01
 Latest value of VSDTC: 2006-04-12

You can move columns, resize them, and do sorting by clicking on the column header.

Un-sort current table

OK

providing information about the number of records, subjects, visits covered, distinct tests (based on VSTESTCD) and earliest and latest value for VSDTC.

This kind of information helps to answer questions one would regularly need to ask oneself:

- are all subjects covered?
- are all visits covered?
- did I include all tests for this domain?
- is the envisaged time span of the trial covered?

Especially not having all visits and tests covered is a usual problem.

For "Interventions", the number of distinct values of the --TRT (treatment), for "Events", the number of distinct values of "--TERM" (Event term) will be provided.

Mapping Completeness for Hypervertical ODM Structures

In some cases, especially when the data does not come from an EDC system, one may have an ODM file having a "hypervertical" structure, i.e. each "record" (ItemGroup) will come as a "parameter - attributes - value" system. For example:

```

<ItemGroupData ItemGroupOID="IG.DEFAULT" ItemGroupRepeatKey="76">
  <ItemData ItemOID="IT.ScreeningNr" Value="3"/>
  <ItemData ItemOID="IT.ExpDelta" Value="-84"/>
  <ItemData ItemOID="IT.ActDelta" Value="-121"/>
  <ItemData ItemOID="IT.ActClock" Value="21NOV22:08:31:00"/>
  <ItemData ItemOID="IT.Activity" Value="Vitals"/>
  <ItemData ItemOID="IT.Parameter" Value="DiastBPsup"/>
  <ItemData ItemOID="IT.Parameter_description" Value="Diastolic blood pressure in supine position"/>
  <ItemData ItemOID="IT.Value" Value="65"/>
  <ItemData ItemOID="IT.Unit" Value="mmHg"/>
  <ItemData ItemOID="IT.Parameter_LowerLimit" Value="50"/>
  <ItemData ItemOID="IT.Parameter_UpperLimit" Value="90"/>
</ItemGroupData>
<ItemGroupData ItemGroupOID="IG.DEFAULT" ItemGroupRepeatKey="77">
  <ItemData ItemOID="IT.ScreeningNr" Value="3"/>
  <ItemData ItemOID="IT.ExpDelta" Value="-84"/>
  <ItemData ItemOID="IT.ActDelta" Value="-121"/>
  <ItemData ItemOID="IT.ActClock" Value="21NOV22:08:31:00"/>
  <ItemData ItemOID="IT.Activity" Value="Vitals"/>
  <ItemData ItemOID="IT.Parameter" Value="HRsup"/>
  <ItemData ItemOID="IT.Parameter_description" Value="Heart rate in supine position"/>
  <ItemData ItemOID="IT.Value" Value="48"/>
  <ItemData ItemOID="IT.Unit" Value="bpm"/>
  <ItemData ItemOID="IT.Parameter_LowerLimit" Value="45"/>
  <ItemData ItemOID="IT.Parameter_UpperLimit" Value="100"/>
</ItemGroupData>

```

providing two measurements, one for diastolic blood pressure and one for heart rate. In "normal" ODM, this would be something like

```

<ItemGroupData ItemGroupOID="Vitals">
  <ItemData ItemOID="IT.DiastBPsup" Value="65"/>
  <ItemData ItemOID="IT.DiastBPsup_LowerLimit" Value="50"/>
  <ItemData ItemOID="IT.DiastBPsup_UpperLimit" Value="90"/>
  <ItemData ItemOID="IT.DiastBPsup_Unit" Value="mmHg"/>
  ...
  <ItemData ItemOID="IT.HRsup" Value="48"/>
  <ItemData ItemOID="IT.HRsup_LowerLimit" Value="45"/>
  <ItemData ItemOID="IT.HRsup_UpperLimit" Value="100"/>
  <ItemData ItemOID="IT.HRsup_Unit" Value="bpm"/>
  ...
</ItemGroupData>

```

and one could use the usual "drag-and-drop" mechanism.

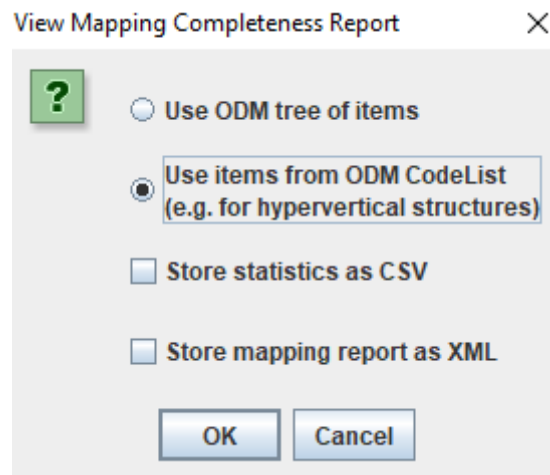
One can of course try to convert one presentation into the other one, but this may be pretty tricky.

Usually, all the parameter values (for "IT.Parameter" and/or for "IT.Parameter_description" are in an ODM codelist. This is e.g. the case when the ODM is generated by our ["ODMGenerator" software](#), which allows to generate ODM files (as well metadata as clinical data) starting from any tabular data such as CSV (possibly extracted from Excel), lab transfers (e.g CDISC Lab), SAS-XPT, etc..

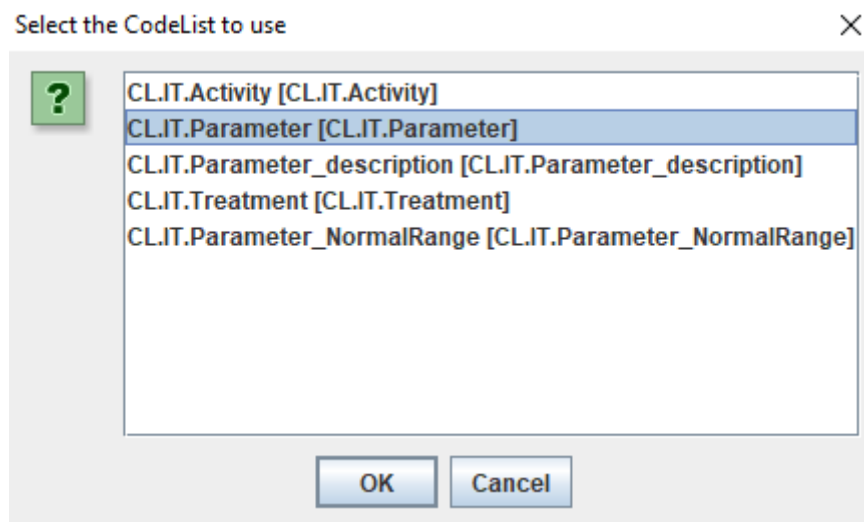
When the ODM data come as "hypervertical", i.e. as "parameter - attributes - value" "records", checking mapping completeness is a bit more difficult.

One can then still make a lot of use of the feature to generate and display a "Mapping Completeness Report". So when using the menu,

and checking the radiobutton "Use Items from ODM CodeList":



one will then be asked which codelist to use (there may be several), for example:



e.g. leading to:

"Required" variables are colored red, "Expected" variables are colored blue, and "Permissible" variables are colored green.

So, one may think "as long as we don't see any "red" variables anymore in our study-specific instance (as these have been 'grayed out'), we are safe ...".

Well, this is a minimum requirement ...

Of course, if one has source data that would need to go into one of the "expected" variables, and even in one of the "permissible" variables, one should provide mappings for these.

And, even then, one may have data in the source that require to add standard variables for the SDTM class ("Findings", "Events", "Interventions"), that are not explicitly in the SDTMIG for that domain. I am thinking about e.g. "time points" (--TPT, --TPTNUM, --TPTREF, --RFTDTC) which are not always mentioned in the SDTMIG for "Findings" domains.

And then there still are the famous "standard non-standard" variables, such as --CLSIG (Clinical Significance) and --REASOC (Reason why the intervention did not occur).

When applicable, one should also look into the ["CDISC Therapeutic Area Guidelines"](#) (TAUGs) for what variables should be included. These tend to mandate, recommend additional variables, or not to use, or use in a specific way, variables from the standard SDTMIG.

Conclusion

Achieving mapping completeness is not easy. Essentially, generating SDTM and SEND datasets is a "categorization", ETL-like exercise. Most of the information from the source will be used in the mappings, but not all of it. Some source data is used in different domains and variables (unfortunately, SDTM has a lot of data redundancy), and will then be used in different mappings.

Fortunately, SDTM-ETL comes with a number of features and tools that make it easier (and more user-friendly) to check for mapping completeness.