SDTM-ETL 4.4 User Manual and Tutorial

Author: Jozef Aerts, XML4Pharma

Last update: 2024-02-06

Ensuring Mapping Completeness

Table of Contents

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:
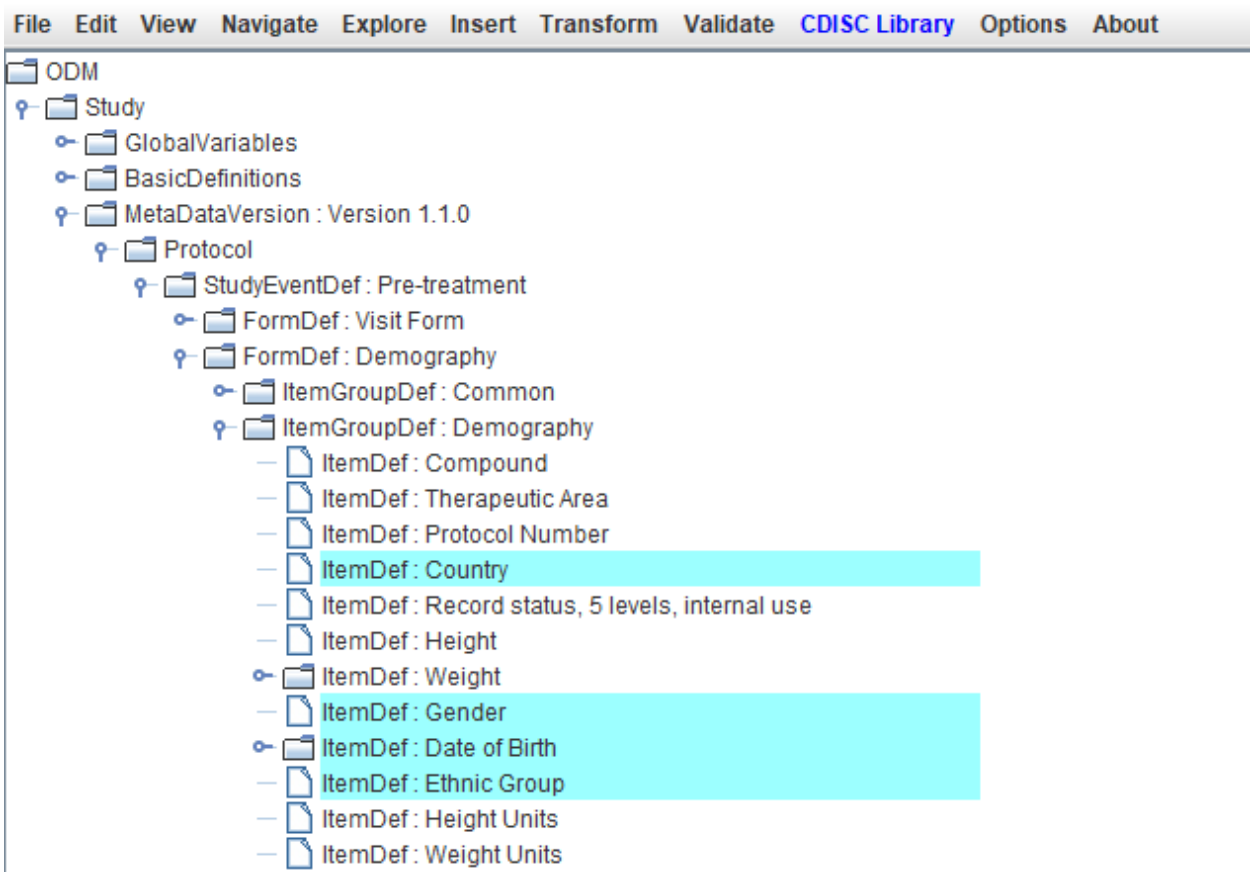
with the items (CRF questions) with a cyan background color, having an **"SDTM annotation"** in the ODM, as very often for items in CDASH forms. The SDTM annotation can be seen when hovering the mouse over the item, for example:
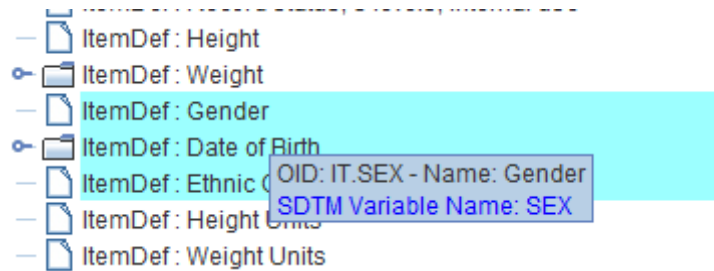
or by using the menu "View - All Item Details", e.g. leading to:

Message

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
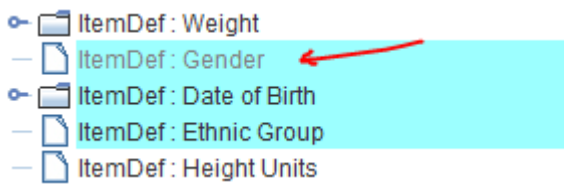
Allowed values (code and decode)

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.SEX[1].

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

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

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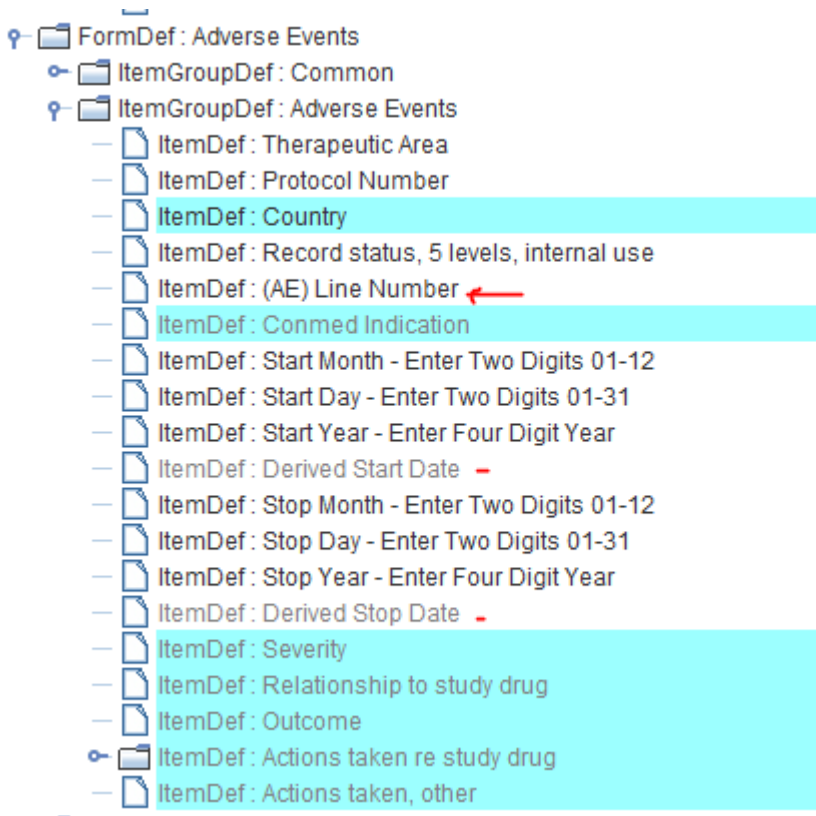
where one sees that "(AE) line number" has not been used in a mapping yet. However, it does not have an SDTM annotation where it should go into in SDTM. Typical usage of "line number" in SDTM is AESPID (Sponsor-defined Identifier), as is also explained in the SDTM-IG:

					single domain for a subject.	
AEREFID	Reference ID	Char		Identifier	Internal or external identifier such as a serial number on an SAE reporting form.	Perm
AESPID	Sponsor-Defined Identifier	Char		Identifier	Sponsor-defined identifier. It may be preprinted on the CRF as an explicit line identifier or defined in the sponsor's operational database. Example: Line number on an Adverse Events page.	Perm
AETERM	Reported Term for	Char		Topic	Verbatim name of the event	Req

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]:
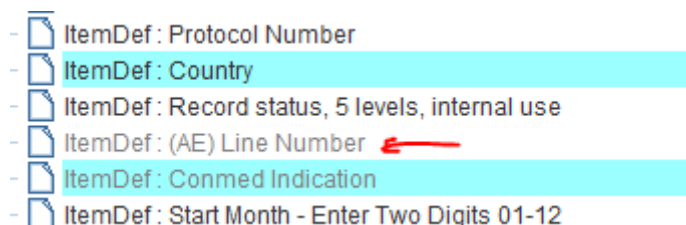
In the example, one also sees that "Start Month...", "Start Day...", "Start Year..." and similar for "End Month..." have not been used. However, "Derived Start Date" and "Derived End Date" have been used. So it is not always so that an item from the ODM must necessarily flow into SDTM. Typical examples 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"

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:

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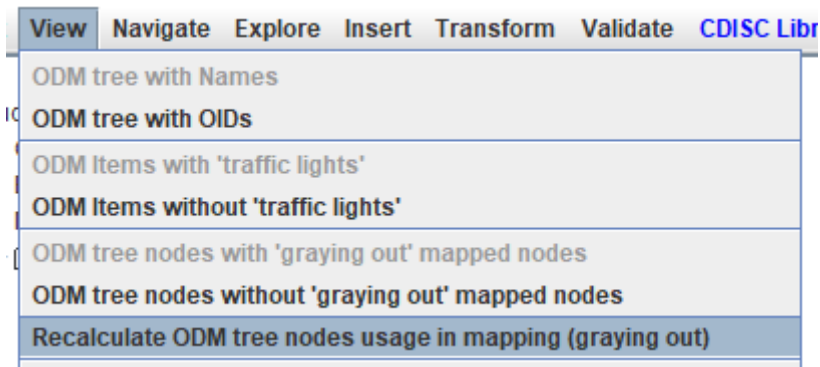
Remark: currently, "graying out" only partially works when doing a "negative selection" during the "generalization", e.g. "exclude all items in the group except for 'Height' and 'Weight'. So, for the moment, we recommend to use "positive selections" (i.e. using "Only for ...") as much as possible.

"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:

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.
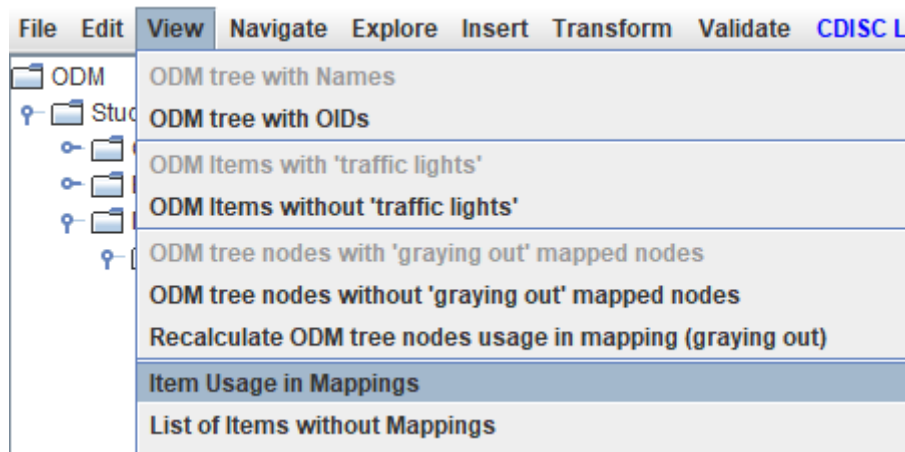
This e.g. leads to:
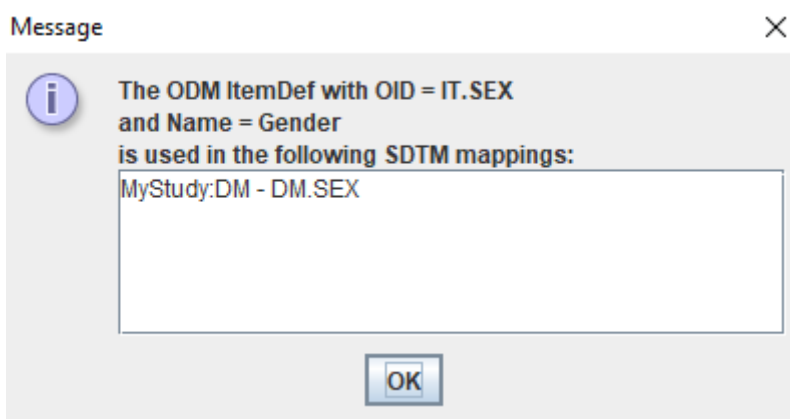
It can of course be that a list is provided when the Item has been used in mappings to different SDTM variables, e.g.:
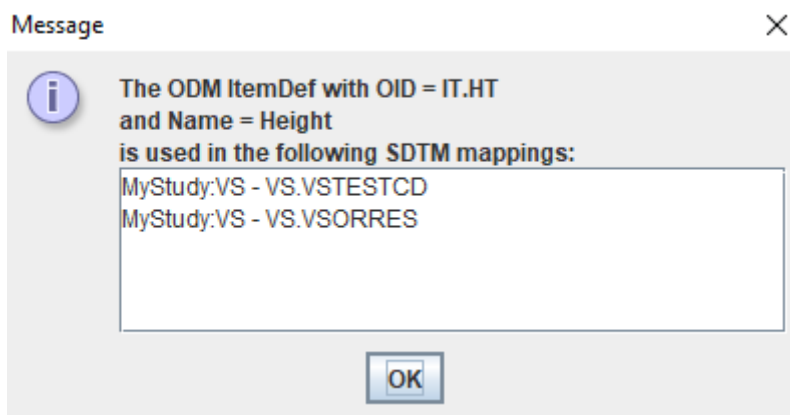
A limitation here still is that when one "reuses" the mapping in another variable, it will not be listed here. For example, when for VSSTRESC (standardized result) one has:

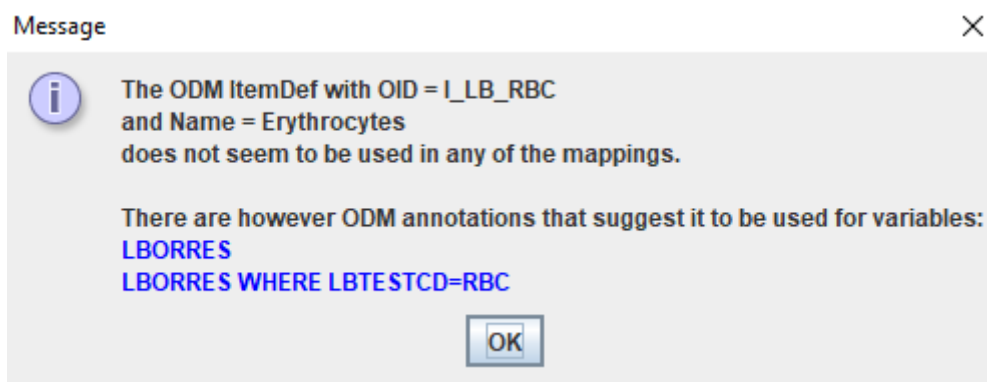VSSTRESC will not appear in the list from "Item Usage in Mappings"[3].

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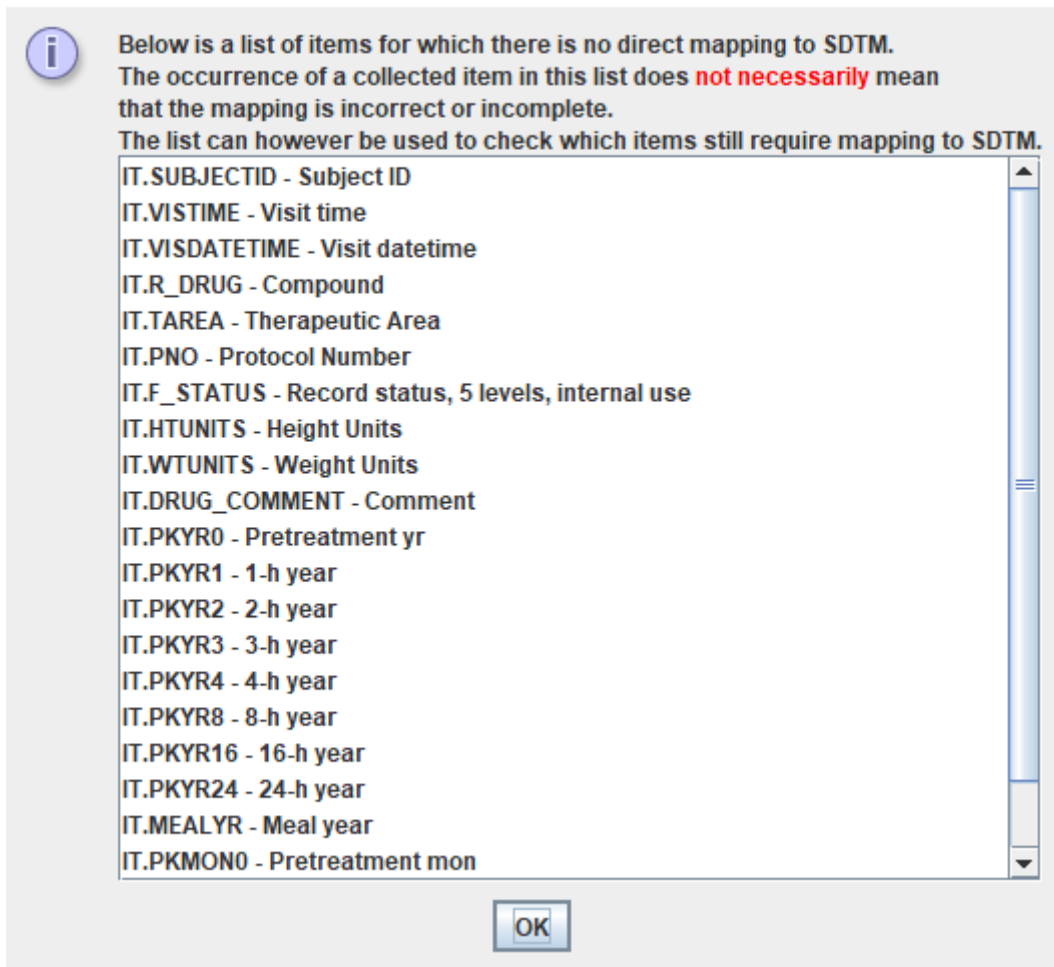
This feature essentially just works the other way around as "View - Item Usage in Mappings".

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 was 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":
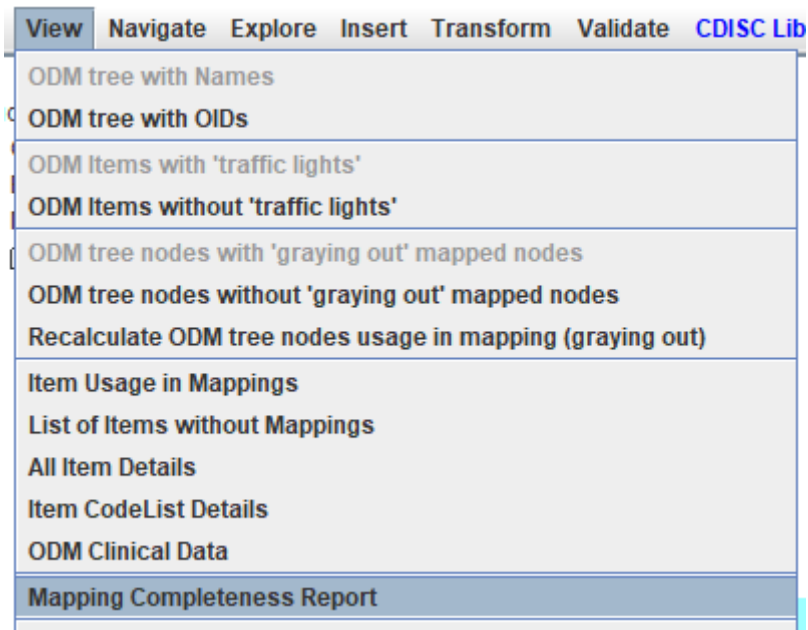
The user is then asked:
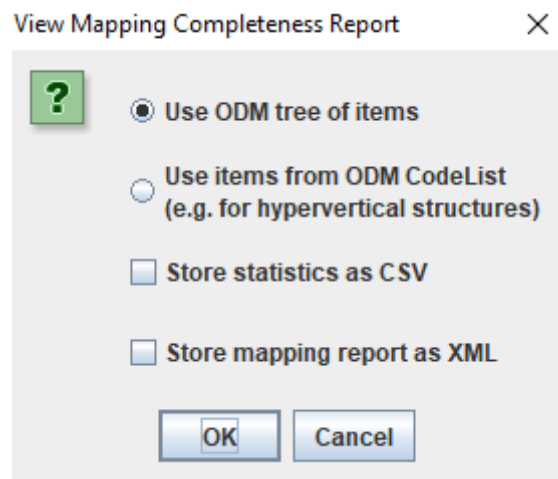
We will explain the choice "Use items from ODM CodeList (e.g. for hypervertical structures) later, in the section "Mapping Completeness for Hypervertical ODM Structures".

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:

Mapping Report

Mapping information is supplied for each ODM Item for which the value of the Value attribute has been used in a mapping

StudyEvent OID (Name)	Form OID (Name)	ItemGroup OID (Name)	Item OID (Name)	(SDTM) Variable OID	Mapping script
SE.VISIT0 (Pre-treatment)	FORM.VISIT (Visit Form)	IG.VISIT (Visit basic data)	IT.SITE (Site ID)	DM.SITEID	# SiteID taken from Visit form of first visit (pre-treatment visit) $DM.SITEID = xpath(/StudyEventData[@StudyEventOID='SE.VISIT0']/FormData[@FormOID='FORM.VISIT']/ItemGroupData[@ItemGroupOID='IG.VISIT']/ItemData[@ItemOID
SE.VISIT0 (Pre-treatment)	FORM.VISIT (Visit Form)	IG.VISIT (Visit basic data)	IT.SUBJECTID (Subject ID)		
SE.VISIT0 (Pre-treatment)	FORM.VISIT (Visit Form)	IG.VISIT (Visit basic data)	IT.VISDATE (Visit date)	DM.RFSTDTC	# Reference date taken as visit date of first visit (Pre-treatment visit) $DM.RFSTDTC = xpath(/StudyEventData[@StudyEventOID='SE.VISIT0']/FormData[@FormOID='FORM.VISIT']/ItemGroupData[@ItemGroupOID='IG.VISIT']/ItemData[@ItemOID Reference end date taken as visit date from last visit (post-treatment visit) $DM.RFENDTC = xpath(/StudyEventData[@StudyEventOID='SE.VISIT1']/FormData[@FormOID='FORM.VISIT']/ItemGroupData[@ItemGroupOID='IG.VISIT']/ItemData[@ItemOID Birthdate taken from Demographics form $BIRTHDATE = xpath(/StudyEventData[@StudyEventOID='SE.VISIT0']/FormData[@FormOID='FORM.DEMOG']/ItemGroupData[@ItemGroupOID='IG.DEMOG']/ItemData[@Item # Date of first visit (reference date) $VISITDATE = xpath(/StudyEventData[@StudyEventOID='SE.VISIT0']/FormData[@FormOID='FORM.VISIT']/ItemGroupData[@ItemGroupOID='IG.VISIT']/ItemData[@ItemOID # Get the difference between the dates as a number of days # datediff gives the date difference as number of days $AGEDAYS = datediff($VISITDATE,$BIRTHDATE) ; # the floor function transforms to an integer $AGE = floor($AGEDAYS / 365.2); $DM.AGE = $AGE;$QS.QSDTC = xpath(/StudyEventData/FormData[@FormOID='FORM.VISIT']/ItemGroupData[@ItemGroupOID='IG.VISIT']/ItemData[@ItemOID='**IT.VISDATE**']/@Value);$SV.S xpath(/StudyEventData/FormData[@FormOID='FORM.VISIT']/ItemGroupData[@ItemGroupOID='IG.VISIT']/ItemData[@ItemOID='**IT.VISDATE**']/@Value);$SV.S xpath(/StudyEventData/FormData[@FormOID='FORM.VISIT']/ItemGroupData[@ItemGroupOID='IG.VISIT']/ItemData[@ItemOID='**IT.VISDATE**']/@Value);# Map ItemData with ItemOID **IT.VISDATE** $PE.PEDTC = xpath(/StudyEventData[@StudyEventOID='SE.VISIT0']/FormData[@FormOID='FORM.VISIT']/ItemGroupData[@ItemGroupOID='IG.VISIT']/ItemData[@ItemOID calculate from first visit date and start of AE start date $FIRSTVISITDATE = xpath(/StudyEventData[@StudyEventOID='SE.VISIT0']/FormData[@FormOID='FORM.VISIT']/ItemGroupData[@ItemGroupOID='IG.VISIT']/ItemData[@ItemOID print 'FIRSTVISITDATE = ' $FIRSTVISITDATE; # For the AE start date, we use the already defined variable !

[<<] [<] [>] [>>] [Save HTML]

[OK]

and further on:

					$AE.AEENDY = ''; }# Mapping using ODM element ItemData with ItemOID **IT.VISDATE** $VS.VSDTC = xpath(/StudyEventData[@StudyEventOID='SE.VISIT0']/FormData[@FormOID='FORM.VISIT']/ItemGroupData[@ItemGroupOID='IG.VISIT']/ItemData[@ItemOID
SE.VISIT1 (Post-treatment)	FORM.VISIT (Visit Form)	IG.VISIT (Visit basic data)	IT.VISTIME (Visit time)		
SE.VISIT1 (Post-treatment)	FORM.VISIT (Visit Form)	IG.VISIT (Visit basic data)	IT.VISDATETIME (Visit datetime)		
SE.VISIT1 (Post-treatment)	FORM.AE (Adverse Events)	IG.COMMON (Common)	IT.SITE (Site ID)	DM.SITEID	# SiteID taken from Visit form of first visit (pre-treatment visit) $DM.SITEID = xpath(/StudyEventData[@StudyEventOID='SE.VISIT0']/FormData[@FormOID='FORM.VISIT']/ItemGroupData[@ItemGroupOID='IG.VISIT']/ItemData[@ItemOID
SE.VISIT1 (Post-treatment)	FORM.AE (Adverse Events)	IG.COMMON (Common)	IT.SUBJECTID (Subject ID)		
SE.VISIT1 (Post-treatment)	FORM.AE (Adverse Events)	IG.AE (Adverse Events)	IT.TAREA (Therapeutic Area)		
SE.VISIT1 (Post-treatment)	FORM.AE (Adverse Events)	IG.AE (Adverse Events)	IT.PNO (Protocol Number)		
SE.VISIT1 (Post-treatment)	FORM.AE (Adverse Events)	IG.AE (Adverse Events)	IT.SCTRY (Country)	DM.COUNTRY	$DM.COUNTRY = xpath(/StudyEventData[@StudyEventOID='SE.VISIT0']/FormData[@FormOID='FORM.DEMOG']/ItemGroupData[@ItemGroupOID='IG.DEMOG']/ItemData[@Item
SE.VISIT1 (Post-treatment)	FORM.AE (Adverse Events)	IG.AE (Adverse Events)	IT.F_STATUS (Record status, 5 levels, internal use)		
SE.VISIT1 (Post-treatment)	FORM.AE (Adverse Events)	IG.AE (Adverse Events)	IT.LINE_NO ((AE) Line Number)	AE.AESPID	# Mapping using ODM element ItemData with ItemOID **IT.LINE_NO** $AE.AESPID = xpath(/StudyEventData[@StudyEventOID='SE.VISIT1']/FormData[@FormOID='FORM.AE']/ItemGroupData[@ItemGroupOID='IG.AE']/ItemData[@ItemOID='**IT.L**
SE.VISIT1 (Post-treatment)	FORM.AE (Adverse Events)	IG.AE (Adverse Events)	IT.AETERM (Conmed Indication)	AE.AETERM	# Mapping using ODM element ItemData with ItemOID **IT.AETERM** # Generalized for all StudyEvents $AE.AETERM = xpath(/StudyEventData/FormData[@FormOID='FORM.AE']/ItemGroupData[@ItemGroupOID='IG.AE']/ItemData[@ItemOID='**IT.AETERM**']/@Val element ItemData with ItemOID **IT.AETERM** # Generalized for all StudyEvents $AE.AEDECOD = xpath(/StudyEventData/FormData[@FormOID='FORM.AE']/ItemGroupData[@ItemGroupOID='IG.AE']/ItemData[@ItemOID='**IT.AETERM**']/@V

The absence of any mapping for "Visit date" and "Visit Time" may trigger the idea that one still needs to develop an SV (Subjects Visits) dataset, where these two items will typically be used.

Also here, the absence of a mapping script for an item does not mean that something is wrong, as another method (e.g. hard-coded, copy from another variable) may have been used to develop the mapping.

This report (it is HTML) can then be saved to file using "Save HTML", e.g. for sharing and discussions with the team.

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:

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

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="HRsup_UpperLimit" Value="100"/>
    <ItemData ItemOID="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..
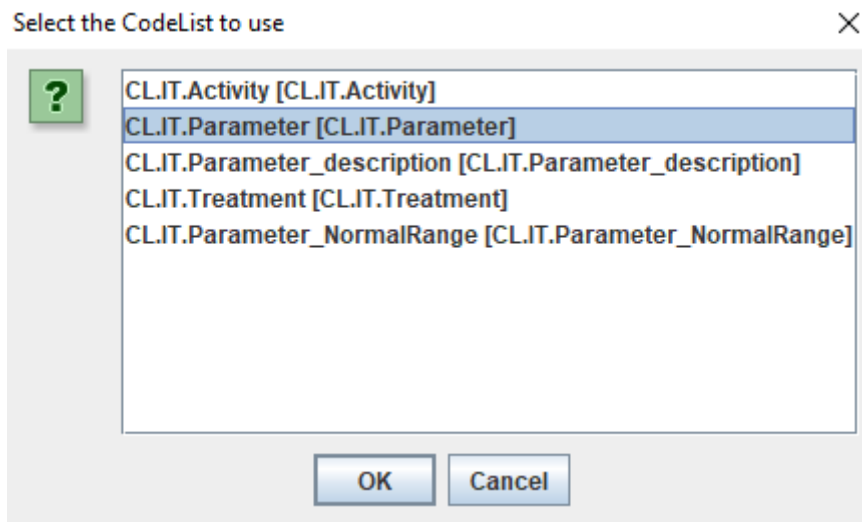
When the ODM data come as "hypervertical", i.e. as "parameter - attributes - value" "records", checking mapping completeness is a bit more difficult.
One can then still make a lot of use of the feature to generate and display a "Mapping Completeness Report". So when using the menu,

and checking the radiobutton "Use Items from ODM CodeList":
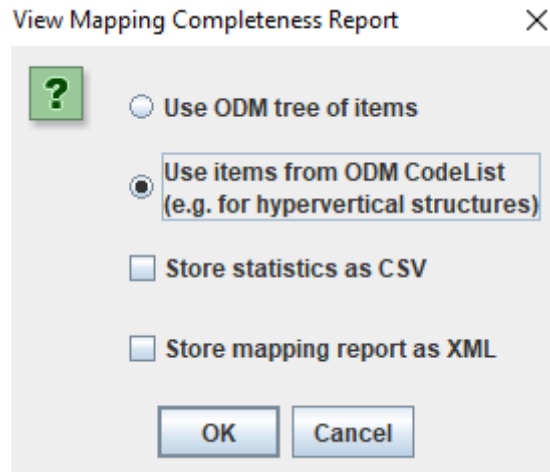
one will then be asked which codelist to use (there may be several), for example:

e.g. leading to:

also here, having a "No mapping found for the coded item" does not necessarily mean that there should be, but one should at least check it.

The Message dialog shows the following entries:

HepC		No mappings found for the coded value
Fibrinogen	LB.LBTESTCD	(mapping code)
INR	LB.LBTESTCD	(mapping code)
LabCoag		No mappings found for the coded value
PT	LB.LBTESTCD	(mapping code)

Fibrinogen – LB.LBTESTCD:

```
$LABTEST = xpath(/StudyEventData/FormData[@FormOID='FO.DEFAULT']/ItemGroupData[@ItemGroupOID='IG.DEFAULT']/ItemData[@ItemOID='IT.Parameter'][@Value='Fibrinogen' or @Value='PT' or @Value='aPTT' or @Value='INR']/@Value);
if($LABTEST = 'Fibrinogen') {
$LB.LBTESTCD = 'FIBRINO';
} elsif($LABTEST = 'PT') {
$LB.LBTESTCD = 'PT';
} elsif($LABTEST = 'aPTT') {
$LB.LBTESTCD = 'APTT';
} elsif($LABTEST = 'INR') {
$LABTESTCD = 'INR';
} else {
$LB.LBTESTCD = 'TODO';
}
```

INR – LB.LBTESTCD:

```
$LABTEST = xpath(/StudyEventData/FormData[@FormOID='FO.DEFAULT']/ItemGroupData[@ItemGroupOID='IG.DEFAULT']/ItemData[@ItemOID='IT.Parameter'][@Value='Fibrinogen' or @Value='PT' or @Value='aPTT' or @Value='INR']/@Value);
if($LABTEST = 'Fibrinogen') {
$LB.LBTESTCD = 'FIBRINO';
} elsif($LABTEST = 'PT') {
$LB.LBTESTCD = 'PT';
} elsif($LABTEST = 'aPTT') {
$LB.LBTESTCD = 'APTT';
} elsif($LABTEST = 'INR') {
$LABTESTCD = 'INR';
} else {
$LB.LBTESTCD = 'TODO';
}
```

PT – LB.LBTESTCD:

```
$LABTEST = xpath(/StudyEventData/FormData[@FormOID='FO.DEFAULT']/ItemGroupData[@ItemGroupOID='IG.DEFAULT']/ItemData[@ItemOID='IT.Parameter'][@Value='Fibrinogen' or @Value='PT' or @Value='aPTT' or @Value='INR']/@Value);
if($LABTEST = 'Fibrinogen') {
$LB.LBTESTCD = 'FIBRINO';
} elsif($LABTEST = 'PT') {
$LB.LBTESTCD = 'PT';
} elsif($LABTEST = 'aPTT') {
$LB.LBTESTCD = 'APTT';
} elsif($LABTEST = 'INR') {
$LABTESTCD = 'INR';
} else {
$LB.LBTESTCD = 'TODO';
}
```

(Dialog buttons: `<<` `<` `>` `>>` Save HTML OK)

Checking completeness on the SDTM side

On the SDTM side, one can always see which variables are, according to the SDTMIG, are "required", "expected" or "permissible, by the color in the template or the study instance, when there is no mapping provided yet:

Domains (ItemGroups)

Domain	Variable	Variable	Variable	Variable	Variable	Variable	Variable	Variable	Variable	Variable
TV	STUDYID	DOMAIN	TV.VISITNUM	TV.VISIT	TV.VISITDY	TV.ARMCD	TV.ARM	TV.TVSTRL	TV.TVENRL	
TS	STUDYID	DOMAIN	TS.TSSEQ	TS.TSGRPID	TS.TSPARMCD	TS.TSPARM	TS.TSVAL	TS.TSVALNF	TS.TSVALCD	TS.TSVCDREF
TD	STUDYID	DOMAIN	TD.TDORDER	TD.TDANCVAR	TD.TDSTOFF	TD.TDTGTPAI	TD.TDMINPAI	TD.TDMAXPAI	TD.TDNUMRPT	
CO	STUDYID	DOMAIN	RDOMAIN	USUBJID	CO.COSEQ	CO.IDVAR	CO.IDVARVAL	CO.COREF	CO.COVAL	CO.COEVAL
SE	STUDYID	DOMAIN	USUBJID	SE.SESEQ	SE.ETCD	SE.ELEMENT	SE.SESTDTC	SE.SEENDTC	SE.TAETORD	SE.EPOCH
SV	STUDYID	DOMAIN	USUBJID	SV.VISITNUM	SV.VISIT	SV.VISITDY	SV.SVSTDTC	SV.SVENDTC	SV.SVSTDY	SV.SVENDY
CM	STUDYID	DOMAIN	USUBJID	CM.CMSEQ	CM.CMGRPID	CM.CMSPID	CM.CMTRT	CM.CMMODIFY	CM.CMDECOD	CM.CMCAT
EC	STUDYID	DOMAIN	USUBJID	EC.ECSEQ	EC.ECGRPID	EC.ECREFID	EC.ECSPID	EC.ECLNKID	EC.ECLNKGRP	EC.ECTRT
EX	STUDYID	DOMAIN	USUBJID	EX.EXSEQ	EX.EXGRPID	EX.EXREFID	EX.EXSPID	EX.EXLNKID	EX.EXLNKGRP	EX.EXTRT
PR	STUDYID	DOMAIN	USUBJID	PR.PRSEQ	PR.PRGRPID	PR.PRSPID	PR.PRLNKID	PR.PRLNKGRP	PR.PRTRT	PR.PRDECOD
SU	STUDYID	DOMAIN	USUBJID	SU.SUSEQ	SU.SUGRPID	SU.SUSPID	SU.SUTRT	SU.SUMODIFY	SU.SUDECOD	SU.SUCAT
AE	STUDYID	DOMAIN	USUBJID	AE.AESEQ	AE.AEGRPID	AE.AEREFID	AE.AESPID	AE.AETERM	AE.AEMODIFY	AE.AELLT
CE	STUDYID	DOMAIN	USUBJID	CE.CESEQ	CE.CEGRPID	CE.CEREFID	CE.CESPID	CE.CETERM	CE.CEDECOD	CE.CECAT
DV	STUDYID	DOMAIN	USUBJID	DV.DVSEQ	DV.DVGRPID	DV.DVSPID	DV.DVTERM	DV.DVDECOD	DV.DVCAT	DV.DVSCAT
HO	STUDYID	DOMAIN	USUBJID	HO.HOSEQ	HO.HOGRPID	HO.HOREFID	HO.HOSPID	HO.HOTERM	HO.HODECOD	HO.HOCAT
DS	STUDYID	DOMAIN	USUBJID	DS.DSSEQ	DS.DSGRPID	DS.DSREFID	DS.DSSPID	DS.DSTERM	DS.DSDECOD	DS.DSCAT
MH	STUDYID	DOMAIN	USUBJID	MH.MHSEQ	MH.MHGRPID	MH.MHREFID	MH.MHSPID	MH.MHTERM	MH.MHMODIFY	MH.MHDECOD
DA	STUDYID	DOMAIN	USUBJID	DA.DASEQ	DA.DAGRPID	DA.DAREFID	DA.DASPID	DA.DATESTCD	DA.DATEST	DA.DACAT
DD	STUDYID	DOMAIN	USUBJID	DD.DDSEQ	DD.DDTESTCD	DD.DDTEST	DD.DDORRES	DD.DDSTRESC	DD.DDRESCAT	DD.DDEVAL
EG	STUDYID	DOMAIN	USUBJID	EG.EGSEQ	EG.EGGRPID	EG.EGREFID	EG.EGSPID	EG.EGTESTCD	EG.EGTEST	EG.EGCAT
IE	STUDYID	DOMAIN	USUBJID	IE.IESEQ	IE.IESPID	IE.IETESTCD	IE.IETEST	IE.IECAT	IE.IESCAT	IE.IEORRES
IS	STUDYID	DOMAIN	USUBJID	IS.ISSEQ	IS.ISGRPID	IS.ISREFID	IS.ISSPID	IS.ISTESTCD	IS.ISTEST	IS.ISCAT
LB	STUDYID	DOMAIN	USUBJID	LB.LBSEQ	LB.LBGRPID	LB.LBREFID	LB.LBSPID	LB.LBTESTCD	LB.LBTEST	LB.LBCAT
MB	STUDYID	DOMAIN	USUBJID	MB.MBSEQ	MB.MBGRPID	MB.MBREFID	MB.MBSPID	MB.MBTESTCD	MB.MBTEST	MB.MBCAT
MS	STUDYID	DOMAIN	USUBJID	MS.MSSEQ	MS.MSGRPID	MS.MSREFID	MS.MSSPID	MS.MSTESTCD	MS.MSTEST	MS.MSCAT
MI	STUDYID	DOMAIN	USUBJID	MI.MISEQ	MI.MIGRPID	MI.MIREFID	MI.MISPID	MI.MITESTCD	MI.MITEST	MI.MITSTDTL
MO	STUDYID	DOMAIN	USUBJID	MO.MOSEQ	MO.MOGRPID	MO.MOREFID	MO.MOSPID	MO.MOLNKID	MO.MOTESTCD	MO.MOTEST
PC	STUDYID	DOMAIN	USUBJID	PC.PCSEQ	PC.PCGRPID	PC.PCREFID	PC.PCSPID	PC.PCTESTCD	PC.PCTEST	PC.PCCAT
PP	STUDYID	DOMAIN	USUBJID	PP.PPSEQ	PP.PPGRPID	PP.PPTESTCD	PP.PPTEST	PP.PPCAT	PP.PPSCAT	PP.PPORRES
PE	STUDYID	DOMAIN	USUBJID	PE.PESEQ	PE.PEGRPID	PE.PESPID	PE.PETESTCD	PE.PETEST	PE.PEMODIFY	PE.PECAT
QS	STUDYID	DOMAIN	USUBJID	QS.QSSEQ	QS.QSGRPID	QS.QSSPID	QS.QSTESTCD	QS.QSTEST	QS.QSCAT	QS.QSSCAT
RP	STUDYID	DOMAIN	USUBJID	RP.RPSEQ	RP.RPGRPID	RP.RPREFID	RP.RPSPID	RP.RPTESTCD	RP.RPTEST	RP.RPCAT
SC	STUDYID	DOMAIN	USUBJID	SC.SCSEQ	SC.SCGRPID	SC.SCSPID	SC.SCTESTCD	SC.SCTEST	SC.SCCAT	SC.SCSCAT
SS	STUDYID	DOMAIN	USUBJID	SS.SSSEQ	SS.SSGRPID	SS.SSSPID	SS.SSTESTCD	SS.SSTEST	SS.SSCAT	SS.SSSCAT
TU	STUDYID	DOMAIN	USUBJID	TU.TUSEQ	TU.TUGRPID	TU.TUREFID	TU.TUSPID	TU.TULNKID	TU.TUTESTCD	TU.TUTEST
TR	STUDYID	DOMAIN	USUBJID	TR.TRSEQ	TR.TRGRPID	TR.TRREFID	TR.TRSPID	TR.TRLNKID	TR.TRLNKGRP	TR.TRTESTCD
RS	STUDYID	DOMAIN	USUBJID	RS.RSSEQ	RS.RSGRPID	RS.RSREFID	RS.RSSPID	RS.RSLNKID	RS.RSLNKGRP	RS.RSTESTCD
VS	STUDYID	DOMAIN	USUBJID	VS.VSSEQ	VS.VSGRPID	VS.VSSPID	VS.VSTESTCD	VS.VSTEST	VS.VSCAT	VS.VSSCAT
FA	STUDYID	DOMAIN	USUBJID	FA.FASEQ	FA.FAGRPID	FA.FASPID	FA.FATESTCD	FA.FATEST	FA.FAOBJ	FA.FACAT

"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 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.