The Define.xml Designer 2022

User Manual for designing and developing Define.xml files

Last update: 2022-09-07

Introduction

This user manual describes the features for designing and developing define.xml 2.0 and 2.1 files using the "Define.xml Designer" software package. For information about designing ODM Study design files, please see the separate user manual for the "ODM Designer" software.

Background

CDISC SDTM, SEND and ADaM electronic submissions to regulatory authorities such as the US FDA, the Japanese PMDA and Chinese NMPA, require a "define.xml" file to accompany the submission files. The define.xml file contains the metadata (data about data) of the information in the submission files. As such, a well designed define.xml file is of utmost importance for explaining the reviewers about what the data is about and how it was generated. The define.xml file is an XML file that is as well human-readable as well as machine-readable. The human readability is ensured by a so called "stylesheet" (XSLT stylesheet), that transforms the XML into information that is interpretable by any modern browser (HTML+CSS).

It is however essentially the define.xml file itself (i.e. the XML) that carries the information - what is seen in the browser is just a human understandable presentation of that data¹.

Although XML is very easy to learn, many people desire to use a good software tool to design or develop a define.xml file describing the contents of their electronic submission, which does not require them to write XML. The "Define.xml Designer 2022" is such a tool.

Ideally, the define.xml file is already developed before study start or in the early stages of a clinical study. The reason for this is that the most important goal of a (commercial) clinical study is to get a marketing authorization for the drug or therapy that is tested. So it is of utmost importance to already test whether the during the study captured data <u>can</u> be transformed to SDTM, SEND or ADaM (for the latter from SDTM). If it is found that this is not possible at the end of the study, then the whole submission is endangered.

Many pharma companies already develop their mapping to SDTM together with the corresponding define.xml even before study start. The ideal tool to do so is XML4Pharma's <u>SDTM-ETL</u> software, allowing s users to develop the mapping to SDTM starting from an ODM file with the study design. The SDTM-ETL software is also used a lot for SEND submissions. In that case, when no ODM file is available, it can be generated from "flat" files such as CSV files using the "<u>ODM Generator</u>" software.

There are however cases where such a synchronized (SDTM + define.xml) approach is not possible. This is especially the case for legacy studies where the SDTM files have already been generated and where the define.xml must be generated "post-SDTM". This often also applies to SEND and ADaM files.

¹A stylesheet could even be used to manipulate the information as seen in the browser. Therefore, the "truth" is what is in the define.xml file, not what is seen in the browser.

Another case is when the user wants to start from a specific SDTM-IG and corresponding define.xml template file, and then add/remove variables and add all other information, adapting the define.xml to the needs of a specific study or submission. For example, many sponsors already do develop a define.xml as a specification to their third party service provider, containing the information how the sponsor wants to obtain the SDTM data files for a specific study at the end of the study. A third use case is that the user obtains a define.xml that is incomplete, is not correct, or does not well explain the submission to the regulatory reviewer.

These use cases are covered by the "Define.xml Designer 2022". This includes the almostautomated generation of a define.xml file starting from a set of SAS Transport 5 ("XPT") files.

This manual describes how the "Define.xml Designer 2022" is used for these three and other use cases. No knowledge of XML is required. It is however recommended that the user has a basic understanding of the goals and structure of define.xml. This can be obtained by reading the define.xml specification (a copy is included with the distribution) and/or attending a one-day CDISC define.xml training. The latter will make you a Define-XML expert, allowing you (when using this software) to develop jewel define.xml files that very well explain the regulatory reviewers what your data is about, how it was generated, and how it is organized.

It is recommended to not use "black box" tools, i.e. tools that let you start e.g. from an Excel worksheet, and "magically" generate a define.xml file from your information in the worksheet. This kind of tools force you to re-generate the define.xml file each time something is not perfect in the result (as displayed by the browser!). Furthermore, they give you no insight into the define.xml that is generated, so essentially, you are not understanding what you are doing. This will usually lead to disaster...

Table of Contents

Introduction	1
Background	1
Starting the software and main principles	4
Creating a new define.xml	13
Creating a define.xml starting from an SDTM/SEND/ADaM template	15
Creating a define.xml starting from an existing set of SAS-XPT files	26
Ordering the dataset definitions	44
Adding and subsetting CDISC Controlled Terminology	46
Adding and subsetting codelists from SAS-XPT files	59
Transforming an "EnumeratedItem" CodeList into a "CodeListItem" CodeList	67
Displaying your define.xml as HTML in a browser using a stylesheet	71
Creating and populating Sponsor-defined Codelists	72
Cleaning up your define.xml	79
Saving your work to file	81
Creating ValueLists	84
Automatically assigning WhereClauses to ValueList entries	98
Annotated CRF and Supplemental Documents	103
Assigning valuelists to SDTM/SEND/ADaM variables	107
Assigning an Origin to variables	110
Retrieving PDF Page Numbers from the annotated CRF	117

Starting the software and main principles

ODM Designer

If you are a windows user, double-click the "Define-XML_Designer.bat" icon or entry in your file explorer. If you are a Linux or MacIntosh user, use the Define-XML_Designer.sh" entry. First, license information will be displayed, followed by the dialog:



The software still supports define.xml 1.0, but it is **strongly recommended** to use define.xml 2.0 or 2.1 as this allows you to much better explain the contents and structure of your submission to the regulatory reviewers. Define.xml 1.0 is also not accepted by the FDA anymore.

Also, most of the new features have not been implemented for Define-XML 1.0 anymore.

If you also want to implement the "Analysis Result Metadata" extension, select the raiobutton "Define.xml 2.1 ADaM with Analysis Results Metadata". This will work with Define-XML 2.1.

After having clicked the "OK" button, the software starts reading the corresponding Define-XML XML-Schema, and generates the graphical user interface elements automatically from the XML-Schema. This e.g. results in:

See Define.xml Designer 2022 by XML4Pharma									
File Edit Add Tra	ansform Validate	View Extra Options	Help						
🚔 📑 🧕									
				Global Study Variables	Study Metadata				
Study Name Stu	dy Description P	rotocol Name							

The "Study Name", "Study Description" and "Protocol Name" already allow you to add some mandatory information. Whereas the "Study Name" and "Study Description" items are self-explaining, the "Protocol Name" is expected the contain the "*The sponsor's internal name assigned to the Study*" which usually is the title of the protocol.

Let us however first have a look at the "Options". To do so, use the menu "Options - Settings".

,	Extra	Options	Help
		Settings	Strg-T
		Dialogs s	Sizes Strg-D

This opens a new dialog:

Options a	and Settings ×	<
Options a	and Settings × ✓ Remember last used directoy ✓ ✓ Show value as tooltip in table cells Use classic mechanism for TranslatedText wizard (only necessary in case of ODM-extensions to TranslatedText) ✓ ✓ Use text window for TranslatedText content × Number of minutes between define.xml autosave 5 ✓ Use Schematron for local validation □ □ Disable drag-and-drop ×]
	Disable define-XML WhereClause Wizard	
	Disable define-XML PDFPageRef Wizard	
	OK Cancel	

We will now discuss the most important options and settings.

- Usually you will want to have the software remember the last used import (file read) or export (file write) directory. You can disable this feature by unchecking the checkbox "Remember last used directory"
- The feature "Show value as tooltip in table cells" is very useful when there are many columns, and you don't want to start dragging column borders to see the complete value. The feature is enabled by default, but can be switched off.
- The wizard for "TranslatedText" (i.e. for support of multiple languages) can be disabled using the checkbox "Use classic mechanism for TranslatedText wizard". You probably however want to use the wizard...
- When adding text that goes within "TranslatedText", a separate small text window shows up for adding the text. If the user does not want this, but wants to add the text directly in the table cell, then the checkbox "Use text window for TranslatedText content" should be unchecked.
- The following option is a very important one: the software will save the full study or define.xml design to an define.xml or ODM file each 5 minutes in the directory "autosave". This allows you to pick up an earlier version of your design. The files in this directory use a naming convention for allowing the user to see when it has been created. For example:

Name	Änderungsdatum	Тур	Größe
DDM_2022_8_30_19-4-15.xml	30.08.2022 19:04	XML-Datei	6.858 KB
DDM_2022_8_30_18-59-15.xml	30.08.2022 18:59	XML-Datei	6.858 KB
DDM_2022_8_30_18-54-15.xml	30.08.2022 18:54	XML-Datei	6.858 KB
DDM_2022_8_30_18-49-15.xml	30.08.2022 18:49	XML-Datei	6.858 KB
DDM_2022_8_30_17-33-40.xml	30.08.2022 17:33	XML-Datei	6.858 KB
DDM_2022_8_30_17-28-40.xml	30.08.2022 17:28	XML-Datei	6.858 KB
DDM_2022_8_30_17-23-40.xml	30.08.2022 17:23	XML-Datei	6.858 KB
DDM_2022_8_30_17-18-40.xml	30.08.2022 17:18	XML-Datei	6.858 KB
DDM_2022_8_30_17-13-40.xml	30.08.2022 17:13	XML-Datei	6.858 KB
DDM_2022_8_30_17-8-40.xml	30.08.2022 17:08	XML-Datei	6.858 KB
DDM_2022_8_30_17-3-40.xml	30.08.2022 17:03	XML-Datei	6.858 KB
DDM_2022_8_30_16-58-40.xml	30.08.2022 16:58	XML-Datei	6.858 KB
DDM_2022_8_30_16-53-40.xml	30.08.2022 16:53	XML-Datei	6.858 KB
IDM_2022_8_30_16-48-40.xml	30.08.2022 16:48	XML-Datei	6.858 KB

The time periods between an "autosave" can be changed by using the textfield after "Number of minutes between define.xml autosave". Only positive numbers can be entered. Setting the value to "0" means that no autosaving is done.

Please regularly have a look at the "autosave" directory and clean it up.

- The designer uses schematron technology during validation. This is the methodology of choice for validating XML files. The software comes with a set of schematron files for as well ODM as for define.xml. If you want to limit validation to XML-Schema validation and hardcoded rules when doing "local" validation (see further), uncheck the checkbox "Use schematron for local validation". You will also be able to make a choice when you start a "local" validation (explained further on).
- Most people like to use "drag and drop". There are however some users that don't. If you are one of these users, check the checkbox "disable drag-and-drop". The latter mechanism is then replaced by a "select and click" mechanism.
- The last two checkboxes enable you to disable the wizards for the define.xml "WhereClause" and "PDFPageRef". If you disable these, you will need to enter some information manually (i.e. by typing). This requires a very good knowledge of define.xml. Use with care!

Options a	nd Settings
?	 Remember last used directoy Show value as tooltip in table cells Use classic mechanism for TranslatedText wizard (only necessary in case of ODM-extensions to TranslatedText) Use text window for TranslatedText content Number of minutes between ODM autosave 5 Use Schematron for local validation Disable drag-and-drop Disable define-XML WhereClause Wizard Disable define-XML PDFPageRef Wizard
	OK Cancel

Let us now go back to the main window that is displayed after the XML-Schema has been read and the graphical user elements have been generated.

For example, for "Global Study Variables", you will find 3 tabs:

ا 🏎	⊱ Define.xml Designer 2022 by XML4Pharma										
File	Edit	Add	Transform	Validate	View	Extra	Options	Help			
	6	1									
Stu	dy Nar	ne	Study Descri	iption F	rotocol	Name					
R									_		

The currently selected tab is always colored yellow, a green tab means that it represents information that must be entered, i.e. is mandatory.

Usually, a table is displayed below the tab name. In this case, there is only 1 row, as only 1 "StudyName" element is allowed in the define.xml file.

"Study Name" is a "text content only" element in XML. In order to enter that text, click the "+" icon (the most left one):



This will result in a new window being displayed, allowing you to enter the study name. For example:

⊱ Extra in	nformation for: StudyName	
?	CDISC Example Study	
	OK Cancel	

After clicking "OK" (or hitting the Enter" key on the keyboard), the text also is displayed in the table:

	Study Name	Study Description	Protocol Name								
	CDISC Example Study										
ł											

If the text is longer than the cell in the row allows, you can inspect the text by clicking on the "magnifying glass" icon (second icon). This will show a dialog window:



In more complicated cases, such as when the ODM or define.xml element has attributes and/or child elements, much more information will be displayed in this dialog window.

You can now add similar information for the tabs "Study Description" and "Protocol Name".

Let us now see how this works for an SDTM or SEND or ADaM variable. These, and all other metadata of the submission can be added/edited after clicking the "Study Metadata" button. This results in:

⊱ Define.xm	nl Designer 2022 by XI	ML4Pharma										-		>
File Edit A	dd Transform Va	alidate View <mark>Extra</mark> C	Options Help											
🚔 📑	1													
	Global Study Variables Study Metadata													
Standards	Annotated CRFs	Supplemental Docum	ents ValueList Definitions	Where	eClause Definitions	Dataset Definitions	Variable Definitions	Codelists	Method Definitions	Comment Definition	s Document links			
	OID		Name		Ty	pe	PublishingSe	t	Versio	n	Status	def:Comment	OID	
														_

(here for Define-XML 2.1).

To add an SDTM/SEND/ADaM variable, click the tab "Variable Definitions". This then displays the following panel:

					Global Study V	ariables Study M	etadata
Standards	Annotated CRFs	Supplemental Documents	ValueList Definitions	WhereClause Definitions	Dataset Definitions	Variable Definitions	Codelists M
OID	Name	DataType	Length	SignificantDigits	SASFieldName	SDSVarName	Origin
E S							
C1							
							

Initially, 10 rows are generated, but you can add additional rows (and you probably will) by clicking the "Add Row" button.

In the table, each row corresponds to a single SDTM/SEND/ADaM variable, and each column to an attribute for it. Adding child element information will be done by clicking the "+" icon on the left of each row, and inspecting all the contents can be done by clicking the "magnifying glass" icon (second from the left).

For a variable "BRTHDTC" (birthdate) in DM, a variable declaration will typically look like (for define.xml 2.1):

Standards Annotated C	RFs Supplemental I	Documents ValueLis	st Definitions Wher	eClause Definitions	Dataset Definitions	Variable De
OID	Name	DataType	Length	SignificantDigits	SASFieldName	SDSVarNa
	BRTHDTC	date	10		BRTHDTC	
R.Q.						
RQ.						

Remark that "SASFieldName" is optional, and does not make sense when one does not use SAS to generate the datasets.

We also deliberately added some invalid information, as a value for "Length" is not allowed when the datatype is "date".

The "DataType" value is enumerated, so when clicking that cell, a list appears from which one needs to select a single one from:

DataType		Length	S
date	•		
integer			
float			
date			
datetime			
time			
text			
string			
double	-		

The "Validate" button (in the lower part of the panel, on the right) can then be used to validate the local contents. In our case this will lead to:

Validation	Results				
row = 1: - ItemDef[1]: Length attribute must not be present for ItemDef with OID 'IT.BRTHDTC' with DataType 'date' - ItemDef[1]: INFO: Rule #168: In the context of a regulatory submission, the child element Description must be present for					
	OK				

Two violations² of the define.xml standard (2.1) have been found:

a) No "Description" child element was found. It represents the "label" of the variable

b) The "Length" attribute is not allowed when the DataType has the value "date"

We just "empty" the "Length" cell by selecting its contents and using the "delete" key on the keyboard. The "Description", which is a child element of "ItemDef" (representing the SDTM/SEND/ADaM variable label) can then be added by using the "+" icon:

Extra information for: ItemDef, with OID = IT.BRTHDTC

RangeCheck CodeLi	st Reference Role	Alias Origin	ValueList Reference	
Description	Question	Extern	alQuestion	MeasurementUnitRef
	Language			Translated Text

As one sees, this table has several rows, as a description in several languages can be added. For a

×

²Remark that we do not speak of "Errors" and "Warnings" as such an assignment is completely arbitrary. We will only speak about an "Error" when it is an XML-Schema error.

submission to the FDA, at least an English description must be provided, which must match³ the "label" from the SDTM-IG or SEND-IG or ADaM-IG which is "Date/Time of Birth" in this case.

Remark that this feature also allows to generate a define.xml with non-English labels, which may become a requirement by e.g. the PMDA and NMPA in future.

We first want to add the language. Clicking in the first "Language" cell pops up a dialog:

Description Question ExternalQuestion MeasurementUnitRe Language Translated Text Image:	RangeCheck	CodeList Refe	erence Role	Alias Origin	ValueList Re	eference		
Language Translated Text	Descriptio	on	Question Extern		nalQuestion Meas		Measureme	ntUnitRe
Choose a language using a button or the selector (You can use the autocompletion facilities of the selector) English Spanish Portugese French German Arabic Chinese Japanese Russian Korean English Other: Set Selected Language:		La	nguage			Tra	anslated Text	
Choose a language using a button or the selector (You can use the autocompletion facilities of the selector) English Spanish Portugese French German Arabic Chinese Japanese Russian Korean English Cother: Set Selected Language:								
Choose a language using a button or the selector (You can use the autocompletion facilities of the selector)								
Choose a language using a button or the selector (You can use the autocompletion facilities of the selector) English Spanish Portugese French German Arabic Chinese Japanese Russian Korean English • Other: Set Selected Language:		<u></u>						×
English Spanish Portugese French German Arabic Chinese Japanese Russian Korean English Other: Set Set	_	Choose (You of	se a language usir can use the autoco	g a button or the ompletion facilitie	selector s of the select	tor)		
Other: Set			English Arabic English	Spanish Chinese	Portugese Japanese	French Russian	German Korean	
Selected Language:			Other:				Set	
			Selected L	anguage:				

Extra information for: ItemDef, with OID = IT.BRTHDTC

from we click the "English" button and then click OK⁴. Resulting in:

	Language
en	

Then clicking the cell under "TranslatedText" for the same row displays a text entry dialog:

³In future, we intent an option that automatically looks up the variable label for a given variable name and SDS version, either locally (from a file that comes with the software), by a RESTful web service, or by a query to the <u>CDISC</u> <u>Library</u>.

⁴This wizard was created so that users do not need to know the ISO language codes.

Se Transla	atedText text content	X
?		•
		•
	OK Cancel	

where we add the "SDTM label", in this case "Date/Time of Birth". After clicking "OK" leading to:

RangeCheck C	odeList Reference	Role Alias	s Origin	ValueList Reference]
Description	Que	stion	Exterr	alQuestion	MeasurementUnitRef
	Language				Translated Text
n				Date/Time of Birth	

We could now add the description in other languages⁵, but this is usually not necessary in the case of submissions to the FDA.

By now, you will probably have asked yourself the question: "*will I really have to add all my variable definitions manually for my SDTM/SEND/ADaM" submission?*". The answer is "No", as we will later learn how to start from an SDTMIG, SENDIG or ADAMIG template to do all this for you.

After clicking the "OK" button:

OID	Name
	BRTHDTC

we see that the "+" icon is now with a yellow background color, meaning that additional

⁵This is the reason why "Description" is not a field in the table for "Variable Definition", as it would then only be possible to add a description in only one language.

information has been added. The background color may also become red after validation, meaning that the additionally added information does not conform to the Define-XML standard.

Clicking the "validate" button again leads to:

Stan	dards	Annotated C	CRFs Supplemental	Documents Val	lueList Definitions	Where	eClause Definitions	Dataset Definitions	Variable Definitions
	DID		Name	DataType	Length		SignificantDigits	SASFieldName	SDSVarName
BR	T.BRTHD	rc	BRTHDTC	date				BRTHDTC	
	Validatio	n Results							×
	- (i)	No validat	tion errors found						
EQ.									
	-								
L4 ~	-								
					OK				
						_			

Essentially, this does not mean yet that everything is fully compliant. For example, at this moment, the software does not know yet whether the variable "BRTHDTC" is a value-level variable or not, so that for example the rule that the "Origin" must be declared either on the variable level or at the value-level, cannot be checked yet.

So, the essentials again:

- Variable attributes are added by clicking the appropriate cell. In case the value is enumerated, a "selection list" will be displayed. In some cases, a wizard will pop up.
- Additional information that is stored in child elements are added by clicking the "+" icon, selecting a tab, and adding the information just like in the main panel. In some cases, one will want to "drill down" into deeper levels and sub-dialogs. Examples will be provided later on.
- "Local" validation can be done using the "validate" button. A short report will then be displayed. In case a mandatory attribute is not populated, the field will be colored red.
- In case additional information has been added, the "+" icon is colored orange. It can turn to
 red after validation in case the added information violates the Define-XML standard

Creating a new define.xml

Everything has a beginning...

Though designing define.xml files from scratch is easily possible, one may in many cases either start from either an SDTM or SEND or even ADaM template⁶, or from a set of existing SAS-XPT SDTM, SEND or ADaM files. The ideal is of course to have a process in place to develop the mapping between operational data and SDTM/SEND data that is synchronized with the development of the define.xml such as in the <u>SDTM-ETL software</u>.

In order to start the development of a new define.xml either from an SDTM or SEND template, or

⁶ In the case of ADaM, only tables for the basic datasets, such as ADSL, will be generated. The user can then add additional dataset definitions and variables in the usual way.

from a set of existing SAS-XPT files with SDTM or SEND data, use the menu "File - New define.xml". The following dialog is presented:

Jetine-XML version: 2.1.0	
I want to start from a CDISC SDTM/SEND/ADaM template	
Jefine_template_ADaMIG_1_3.xml	
Jefine_template_SDTMIG_3.1.2_SDTM_1.2.xml	
lefine_template_SDTMIG_3.1.2_SDTM_1.2_oncology_draft.xml	
Jefine_template_SDTMIG_3.1.2_SDTM_1.2_PGx_new.xml	
Jefine_template_SDTMIG_3.1.3_Med_Devices.xml	
Jefine_template_SDTMIG_3.1.3_SDTM_1.3.xml	
lefine_template_SDTMIG_3.1.3_SDTM_1.3_Non_Subject_Data.xml	
Jefine_template_SDTMIG_3.2_AssociatedPersons.xml	
I want to start from a set of SAS-XPT files	
SDTM Browse SAS-XPT	
I want to load by CDISC published Controlled Terminology	
ADaM_Terminology_2021-12-17.xml	
ADaM_Terminology_2022-06-24.xml	
COA_Terminology_2014-12-19.xml	
COA_Terminology_2015-03-27.xml	
QRS_Terminology_2015-06-26.xml	
QRS_Terminology_2015-09-25.xml	
QS-FT_Terminology_2014-06-26.xml	
QS-FT_Terminology_2014-09-26.xml	
Generate Define-XML Variable DataType, Length and SignificantDigits from XPT content	1
Try to create subset CodeLists from XPT content and selected Controlled Terminology	
Try to create sponsor-defined CodeLists from definitions in a 'sponsorcodelistvariables	.dat'
Try to create Valuelists for Supplemental Qualifier datasets from XPT content	
Try to create Valuelists from definitions in a 'valuelistvariables.dat' file	
itudy OID (required)	
itudy Name (required)	
itudy Description (required)	

Either click the checkbox "I want to start from a CDISC SDTM/SEND/ADaM template" or the checkbox "I want to start from a set of SAS-XPT files". Also create a study OID (this is the Define-XML identifier for the study, so one can use a "mnemonic" like "CES" for "CDISC Example Study", or use an identifier that is used within your company), a study short name, a study description and the protocol name or title.

Creating a define.xml starting from an SDTM/SEND/ADaM template

In case you checked the checkbox "I want to start from a CDISC SDTM/SEND/ADaM template", the following fields become visible and enabled:

⊱ New Study Metadata	\times
Define-XML version: 2.1.0	
✓ I want to start from a CDISC SDTM/SEND/ADaM template	
define_template_SDTMIG_3.2_AssociatedPersons.xml	-
define_template_SDTMIG_3.2_SDTM_1.4.xml	
define_template_SDTMIG_3.3_SDTM_1.7.xml	
define_template_SDTMIG_3.4_SDTM_2.0.xml	
define_template_SENDIG_3.0.xml	
define_template_SENDIG_3.1.1.xml	=
define_template_SENDIG_3.1.xml	
define_template_SENDIG_DART_1.0.xml	-

I want to start from a sot of SAS VDT files

One can then select a template from the list. It contains all necessary information from the SDTM-IG or SEND-IG. One can later than still merge with another template. For example, if one has selected "define_template_SDTMIG_3.2.xml", then one can later still merge with domains from the "define_template_SDTM_3.2_AssociatedPersons" or even with templates for newly developed SDTM domains published by CDISC. It is also very easy to develop own templates and then merge them with CDISC-IG templates.

All these templates come as files with the software, and the content of the directory where they reside is inspected by the software. This also means that once CDISC publishes a new Implementation Guide, we will deliver the template file or files for it, and adding it to the directory will suffice to be able to use them - no software update will be necessary.

Suppose we would like to use the latest SDTM version, so we select

"define_template_SDTMIG_3.4.xml". It is then also extremely useful to already add the latest by CDISC published controlled terminology⁷. In order to do so, check the checkbox "I want to load by CDISC published Controlled Terminology" and then select a version (probably the latest) from the list. Also here, when new controlled terminology is published by CDISC, we will provide it as a file, and adding it to the directory where the controlled terminology files reside will be sufficient to let appear it in the list in the software.

Additional CDISC controlled terminology can also be loaded separately (see further on) and merged later.

⁷In earlier versions of SDTM, the controlled terminology was coupled to the SDTM version. This is no longer the case, and one can select any more recent version of the controlled terminology. It is usually advised to use the latest published controlled terminology, and then to stick to that version during the generation of the SDTM and the define.xml.

⊱ New Study Metadata

Define-XML version: 2.1.0							
I want to start from a CDISC SDTM/SEND/ADaM template							
define_template_SDTMIG_3.2_AssociatedPersons.xml	•						
define_template_SDTMIG_3.2_SDTM_1.4.xml							
define_template_\$DTMIG_3.3_\$DTM_1.7.xml							
define_template_\$DTMIG_3.4_\$DTM_2.0.xml							
define_template_SENDIG_3.0.xml							
define_template_SENDIG_3.1.1.xml	=						
define_template_SENDIG_3.1.xml							
define_template_SENDIG_DART_1.0.xml	•						
I want to start from a set of SAS XDT files							

After clicking "OK", the system will start loading the templates and the controlled terminology, and will ask for additional information:

⊱ New MetaDataVersion	\times
MetaDataVersion OID (required)	
MDV.SDTMIG.3.4	
MetaDataVersion Name (required)	
Template define.xml generated using the CDISC Library API	
MetaDataVersion Description (required)	
OK Cancel	

A proposal is made for the "MetaDataVersion OID" and the "MetaDataVersion Name". These are arbitrary - they are currently not really used as only one "MetaDataVersion" is currently allowed in a define.xml⁸.

For the "MetaDataVersion Description", we provide useful information, e.g.:

⊱ New MetaDataVersion	\times
MetaDataVersion OID (required)	
MDV.SDTMIG.3.4	
MetaDataVersion Name (required)	
MetaDataVersion for SDTMIG-3.4	
MetaDataVersion Description (required)	
MetaDataVersion for SDTMIG-3.4 for the CDISC Example Study (CES)	
OK Cancel	

⁸This might change in future when the regulatory authorities allow to have updates on define.xml content.

p. 16

After clicking OK, the templates and CDISC controlled terminology are loaded. The following message appears:

Message

You decided to use the template define.xml file define_template_SDTMIG_3.4_SDTM_2.0.xml Please be aware that you will still need to adapt the information for each dataset, and for each SDTM/SEND variable.
 You have loaded CDISC controlled terminology, but you will often still need to subset it (depending on the tests executed or foreseen in the protocol, or the available choices in the CRFs). Also ValueLists will need to be developed. You can generate ValueLists starting from existing codelists using the menu 'Extra' - 'Generate ValueList from CodeList'.

Remark that the whole CDISC controlled terminology has been loaded, and that you will need to subset many of the codelists. For example, the "LBTESTCD" contains over different 1,500 codes for lab tests, but you will probably have considerably less in your study. We will later learn how to subset a codelist.

Also, you will later have to develop the necessary "value-lists" (value-level metadata). The software has a wizard for this, allowing you to generate a "ValueList" from an existing codelist.

										Global Study V	/ariables	Study I	Metadat
Star	ndards Annotate	ed CRFs	Supplemen	ntal Documents	ValueL	ist Definitions	Wher	eClause Definitions	Dat	aset Definitions	Variable	Definitions	Cod
	OID	Name	e	DataType		Length		SignificantDigits	SAS	FieldName	SDSVarN	ame	Origin
R a	MH MHENTPT	MHEN	VTPT	text		80			MHE	NTPT			
R Q	BS BSSEO	BSSE	0	integer		8			BSS	FO			
RQ	BS BSGRPID	BSGE	RPID	text		80			BSG	RPID			
RQ	BS BSREEID	BSRE	FID	text		80			BSR	FFID			
RQ	BS BSSPID	BSSE	2 D	text		80			BSS	PID			
<u>n</u> a	BS.BSTESTCD	BSTE	STCD	text		8			BST	ESTCD			
n q	BS.BSTEST	BSTE	ST	text		40			BST	EST			
n q	BS.BSCAT	BSCA	AT .	text		80			BSC	AT			
DQ.	BS.BSSCAT	BSSC	CAT	text		80			BSS	CAT			
DQ	BS.BSORRES	BSOF	RRES	text		80			BSO	RRES			
DQ.	BS.BSORRESU	BSOF	RRESU	text		80			BSO	RRESU			
DQ.	BS.BSSTRESC	BSST	RESC	text		80			BSS	TRESC			
	BS.BSSTRESN	BSST	RESN	float		8		2	BSS	TRESN			
DQ.	BS.BSSTRESU	BSST	RESU	text		80			BSS	TRESU			
D Q	BS.BSSTAT	BSST	AT	text		8			BSS	TAT			
D Q	BS.BSREASND	BSRE	EASND	text		80			BSR	EASND			
D Q	BS.BSNAM	BSNA	M	text		80			BSN	AM			
D Q	BS.BSSPEC	BSSF	PEC .	text		80			BSS	PEC			
	BS.BSANTREG	BSAN	ITREG	text		80			BSA	NTREG			
	BS.BSSPCCND	BSSF	CCND	text		80			BSS	PCCND			
	BS.BSMETHOD	BSME	THOD	text		80			BSM	ETHOD			
BQ	BS.BSBLFL	BSBL	.FL	text		1			BSB	LFL			
BQ	BS.BSDTC	BSDT	TC	datetime					BSD	TC			
D Q	BS.BSDY	BSDY	(integer		8			BSD	Y			
BQ	BS.BSTPT	BSTP	т	text		80			BST	PT			
D Q	BS.BSTPTNUM	BSTP	TNUM	text		80			BST	PTNUM			
BQ	BS.BSELTM	BSEL	.TM	text		80			BSE	LTM			
BQ	BS.BSTPTREF	BSTP	TREF	text		80			BST	PTREF			
BQ	BS.BSRFTDTC	BSRF	TDTC	datetime					BSR	FTDTC			
BQ	CP.CPSEQ	CPSE	Q	integer		8			CPS	EQ			
D Q	CP.CPGRPID	CPGF	RPID	text		80			CPG	RPID			
BQ	CP.CPREFID	CPRE	EFID	text		80			CPR	EFID			
BQ	CP.CPSPID	CPSF	PID	text		80			CPS	PID			
- CA		C D I N	IVID	toxt		lon			CDI	NIZID			

After clicking OK, and navigating to the "Variable Definitions", one gets:

One sees that the table has been filled with information, and that for each variable, a datatype and a maximum length has been proposed. You will still have to adapt this information for your own study. For example, for RFSTDTC and similar variables (RFENDTC, RFXSTDTC, ...) you only have collected the date without a time part, you should change the datatype "datetime" into "date".

 \times

This can easily be achieved by clicking in the cell, and select "date" instead of "datetime". For example:

_	000010	10000			
	RFSTDTC	datetime			
	RFENDTC	datetime	•		
	RFXSTDTC	date			
_	RFXENDTC	datetime			
	RFICDTC	time	_		
ł	RFPENDTC	tovt			
	DTHDTC				
	DTHFL	string		2	
	SITEID	double		80	
	INVID	URI		80	
	INVNAM	boolean	▼	80	
	DDTUDTO	date the stress of		1	

You will also need to adapt the value for "Length" to the maximum length the data can have for that variable in your SAS-XPT files. So if in your study "USUBJID" never has more than 20 characters, you should change the value of "60" into "20", <u>and</u> take care that in your SAS-XPT file the field length is exactly 20⁹. If you start developing your define.xml before the study is being finalized (which is a best practice), and do not know all the maximal lengths, no worry, you can still leave the "Length" with the proposed value, and do an automated adaption from the SAS-XPT file contents when you have all your SAS-XPT files available (see further on).

Remark that the "Length" attribute is only there due to the regulatory requirement to submit using SAS Transport, and does not have a meaning when using a modern transport format.

One also sees that all rows in the table have the "+" icon colored yellow, meaning that additional information is available. For example, when clicking on the "+" icon, the following dialog is displayed:

	lueList Reference	Va	Origin	Alias	Role	eference	CodeList R	RangeCheck
Measu	nalQuestion	xterr	I		estion	Qu	otion	Descrip
Translated Tex					e	Languag		
ntifier	Unique Subject Iden							en

Extra information for: ItemDef, with OID = USUBJID

showing that the "label" has already been added, i.e. loaded from the template.

For values that are coded, e.g. for LBTESTCD, clicking the "+" icon and navigating to the "CodeList Reference" tab leads to:

⁹This has to do with the fixed field and record length in SAS-XPT. Once the regulatory will start accepting Dataset-XML or Dataset-JSON files instead of outdated SAS-XPT, the requirement to provide a "Length" will probably be dropped.

Extra information for: ItemDef, with OID = LB.LBTESTCD

NangeCheck CC	Duelist Referenc	Cuestion	Allas		valueLIST Reference	MeasurementUnitDef
Description		Question			XternalQuestion	Measurementomicker
odeListOID	CD				Codelist definitions:	
L.G03047.LB1E31	CD				CL.MEDDRA	
					CL.ISO3166	
					CL.C141657.TENMV	/1TC
					CL.C141656.TENMV	V1TN
					CL.C141663.A4STR	1TC
					CL.C141662.A4STR	11N
					CL.C141001.D4STR	110 17N
					CL.C115388.SIXMW	1TC
					CL.C115387.SIXMW	/1TN
					CL.C182464.AIMS01	101T07OR
					CL.C182465.AIMS01	108T09OR
					CL.C182466.AIMS01	110OR
					CL.C182467.AIMS01	111T12OR
					CL.C182502.AIMS01	I01T07STR
					CL.C182503.AIMS01	10810951R
					CL.C182505.AIMS01	111112STR
					CL.C101805.AIMS01	ITC
					•	
						Search
A	dd Row			Delete	Selected Row	Copy Selected Row
Move Se	lected Row Up			Move Sel	ected Row Down	Validate
Sug	gest OIDs			Sort by	Order Number	Reassign OrderNumbers
Save	to Library			Load	from Library	Show XML
				Show	v Search Panel	

The left side shows the pre-defined codelist (OID identifier) from the loaded template, the right side shows a list of all available codelists (including external ones like MedDRA), that can be used for drag-and-drop. Later, when having developed a subset codelist (for our study) with LBTESTCD codes, we will replace the reference to a new one for our subset codelist.

With the search button that appears under the list, it is very easy to find a specific codelist by a word in its name.

Each of the choices also has a tooltip showing the full name of the codelist, e.g.:

	Codelist
	definitions:
1000	CL.C101805.AIM \$01TC
	CL.C101806.AIM \$01TN
1	CL.C66767.ACN
	CL.C101865.ACSPCAT
1	CL.C18248 Action Taken with Study Treatment
	CL.C182485.APCH102OR
1	CL.C182486.APCH103OR
2	

We will later also learn how to add information for "Origin" (required either on variable or valuelevel for each variable in the context of a regulatory submission), and other important information.

In the main window, we can easily search for a specific variable by clicking the "Show Search Panel" at the bottom of the screen:

 \times

RE RP	QS RE RP	Yes Yes Yes	No No	QS RE RP	QS RE RP		Ta Ta Ta	abulation abulation abulation	One record One record One record	per Location.QS per f Location.RE per f Location RP	STD.SDTMIG-3.4 STD.SDTMIG-3.4 STD.SDTMIG-3.4	v
		Add Row					Delete Selecte	d Row			Сору	Selected Row
		Move Selected Ro	w Up				Move Selected R	ow Down				Validate
		Suggest OIDs					Sort by Order N	lumber			Reassi	gn OrderNumbers
		Save to Library	1				Load from Li	brary				Show XML
							> Show Search	h Panel				
						/	/					

leading to the appearance of a "Search" panel near the top of the screen:

St	andards	Annotated CRFs	Supplemental [Documents Va	alueList Definitions	WhereClause Definitions	Dataset Definitions	Variable Definitions	Codelists Met	hod Definitions	Comment Definitions	Document links	
					Search for:			Search	Find Next	Match case	Whole words only		
					Search within:	: 🗹 All Columns							
						V Name V D	ataType 🖌 Length	SignificantDigi	ts 🔽 SA SFieldName				
					. 0.0		atarijpo 🔄 congar	- orginitouritorigi					
					SDSVarNa	ime 🖌 Origin 🛛 🖌 C	omment 🖌 DisplayFor	mat 🖌 CommentOID					
	OID	Nam	ie	DataType	Length	SignificantDigits	SASFieldName	SDSVarName	Origin	Comment	DisplayFormat	CommentOID	
1	Aut run ICD		DINTET 1	ICAL	00		MULLINIX II I						A 3
120	NH.MHEN	NTPT MHE	NTPT	text	80		MHENTPT						
DY	BS.BSSE	Q BSS	EQ	integer	8		BSSEQ						
D'	BS.BSGR	PID BSG	RPID	text	80		BSGRPID						
D'	BS.BSRE	FID BSR	EFID	text	80		BSREFID						
DY	BS.BSSPI	ID BSS	PID	text	80		BSSPID						
DY	BS.BSTES	STCD BST	ESTCD	text	8		BSTESTCD						
DY	BS.BSTES	ST BST	EST	text	40		BSTEST						
D.	BS.BSCA	T BSC	AT	text	80		BSCAT						
EN C	De neec	AT DOO	CAT	tout	00		DECOAT						

allowing e.g. to find "LBTESTCD" in the list of variables:

Search for: LBT	ESTCD			Search Find	Next Match case	Whole words only
Search within:	All Columns					
✓ OID	✓ Name	Repeating	IsReferenceData	SASDatasetName	Domain	
Origin	Role	Purpose	Comment	Structure	ArchiveLocationID	
StandardOID	IsNon Standard	🗌 HasNoData	CommentOID			

and clicking the "Search" button will then select the entry for LBTESTCD:

<u>н</u> т т					
<mark>_</mark>]Q	IS.ISRFTDTC	ISRFTDTC	datetime		
C) Q	LB.LBSEQ	LBSEQ	integer	8	
C) Q	LB.LBGRPID	LBGRPID	text	80	
C) Q	LB.LBREFID	LBREFID	text	80	
<mark>∎</mark> Q	LB.LBSPID	LBSPID	text	80	
<mark>∎</mark> Q	LB.LBTESTCD	LBTESTCD	text	8	
<mark>∎</mark> Q	LB.LBTEST	LBTEST	text	40	
C) Q	LB.LBTSTCND	LBTSTCND	text	80	
C) Q	LB.LBBDAGNT	LBBDAGNT	text	80	
C) Q	LB.LBTSTOPO	LBTSTOPO	text	80	
RQ	I R I RCAT	I BCAT	tovt	80	

clicking the "magnifying glass" icon on LBTESTCD leads to:

Name	Value
OID	LB.LBTESTCD
Name	LBTESTCD
DataType	text
Length	80
SignificantDigits	
SASFieldName	LBTESTCD
SDSVarName	
Origin	
Comment	
DisplayFormat	
CommentOID	
ontent for Desc Tra Language: Englist Text: Lab Test or	ription nslatedText h r Examination Sho

Now going back to the main window, and selecting the "Dataset Definitions", we find:

5-	Define.xml	Designer 2022 by XIV	1L4Pharma											
File Edit Add Transform Validate View Extra Options Help														
8 F														
8														
										4 - 4 - 4 -				
								Global Study v	ariables Study Me	tadata				
Sta	andards	Annotated CRFs	Supplemental Docum	ents ValueList [Definitions V	WhereClause Definiti	ons Da	taset Definitions	Variable Definitions	Codelists	Method Definitions	Comment Definitio	ns Document	links
		Name	Repeating	IsReferenceData	SASDatasetN	la Domain	Origin	Role	Purnose	Comment	Structure	Archivel ocation	StandardOID	IsNo
DO	200	CO	Yes	No	CO	CO	Ongin	1000	Tabulation	Comment	One record per	Location CO	STD SDTMIG-3.4	1
R	NDM.	DM	Yes	No	DM	DM			Tabulation		One record per	Location DM	STD SDTMIG-3.4	i
B	SE	SE	Yes	No	SE	SE			Tabulation	-	One record per	Location SE	STD SDTMIG-3.4	4
B	SM	SM	Yes	No	SM	SM			Tabulation		One record per	Location SM	STD SDTMIG-3.4	<u>i</u>
D C	SV	SV	Yes	No	SV	SV			Tabulation		One record per .	Location.SV	STD.SDTMIG-3.4	4
D	AG	AG	Yes	No	AG	AG			Tabulation		One record per .	Location.AG	STD.SDTMIG-3.4	4
D C	CM	CM	Yes	No	CM	CM			Tabulation		One record per .	Location.CM	STD.SDTMIG-3.4	4
D	EC	EC	Yes	No	EC	EC			Tabulation		One record per .	Location.EC	STD.SDTMIG-3.4	4
D	EX	EX	Yes	No	EX	EX			Tabulation		One record per	Location.EX	STD.SDTMIG-3.4	1
E) C	ML	ML	Yes	No	ML	ML			Tabulation		One record per f	Location.ML	STD.SDTMIG-3.4	F I
	PR	PR	Yes	No	PR	PR			Tabulation		One record per .	Location.PR	STD.SDTMIG-3.4	1
D)	SU	SU	Yes	No	SU	SU			Tabulation		One record per .	Location.SU	STD.SDTMIG-3.4	1
D	AE	AE	Yes	No	AE	AE			Tabulation		One record per .	Location.AE	STD.SDTMIG-3.4	1
D) C	BE	BE	Yes	No	BE	BE			Tabulation		One record per i	Location.BE	STD.SDTMIG-3.4	1
	CE	CE	Yes	No	CE	CE			Tabulation		One record per .	Location.CE	STD.SDTMIG-3.4	1
D, C	DS	DS	Yes	No	DS	DS			Tabulation		One record per .	Location.DS	STD.SDTMIG-3.4	1
	DV	DV	Yes	No	DV	DV			Tabulation		One record per .	Location.DV	STD.SDTMIG-3.4	1
	δ HO	HO	Yes	No	HO	HO			Tabulation		One record per .	Location.HO	STD.SDTMIG-3.4	4
D, C	MH	MH	Yes	No	MH	MH			Tabulation		One record per .	Location.MH	STD.SDTMIG-3.4	1
D) (BS	BS	Yes	No	BS	BS			Tabulation		One record per .	Location.BS	STD.SDTMIG-3.4	4
D) C	CP	CP	Yes	No	CP	CP			Tabulation		One record per t	Location.CP	STD.SDTMIG-3.4	1
	CV	CV	Yes	No	CV	CV			Tabulation		One record per f	Location.CV	STD.SDTMIG-3.4	1
D	DA	DA	Yes	No	DA	DA			Tabulation		One record per .	Location.DA	STD.SDTMIG-3.4	1
D) <	DD	DD	Yes	No	DD	DD			Tabulation		One record per f	Location.DD	STD.SDTMIG-3.4	1
	EG	EG	Yes	No	EG	EG			Tabulation		One record per .	Location.EG	STD.SDTMIG-3.4	4
D, C	FT	FT	Yes	No	FT	FT			Tabulation		O One record p	er finding per subje	TD.SDTMIG-3.4	4
D, C	GF	GF	Yes	No	GF	GF			Tabulation		One record per 1	Location.GF	TSTD.SDTMIG-3.4	4
0	NIE 🛛	IE	Yes	No	IE	IE			Tabulation		One record per i	Location.IE	STD.SDTMIG-3.4	4
B	IS	IS	Yes	No	IS	IS			Tabulation		One record per t	Location.IS	STD.SDTMIG-3.4	1

p. 21

with one row per domain. Clicking on the "+" icon e.g. For "DM" opens a new window with: Х

疑 Extra information for: ItemGroupDef, with OID = DM

?	Description Variable References Alias Class Documen	t links
_	Language	Translated Text
	en	Demographics

I.e. the domain label is already present ("Demographics"). Selecting the "Variable References" leads to:

ItemOID KeySequ MethodO Role RoleC OrderNumber Mandat Collecti IsNon HasNo C STUDYID Identifier 1 Yes Identifier STUDYID DOMAIN C DOMAIN Identifier 2 Yes Identifier STUDYID DOMAIN C DDMAIN Identifier 3 Yes Identifier STUDYID DOMAIN C DM.SUBJID Topic 4 Yes Identifier Adentifier Adentifier<	Description Varia	ble Referen	ices Alias	Class Docu	ment link	s					
STUDYID Identifier 1 Yes TUDYID Quomain Identifier 2 Yes Domain Quomain Identifier 3 Yes Domain Quomain Identifier 3 Yes Domain Quomain Identifier 3 Yes Domain Quomain Topic 4 Yes Domain Quomain Record Qualifier 5 No AG.AGSEQ Quomain Record Qualifier 7 No AG.AGRPID Quomain Record Qualifier 9 No AG.AGLNKID Quomain Record Qualifier 10 No AG.AGLNKID Quomain Record Qualifier 11 No AG.AGLNKID Quomain Record Qualifier 12 No AG.AGCAT Quomain Record Qualifier 13 No AG.AGCAT Quomain Record Qualifier 15 Yes AG.AGSPES Quomain Record Qualifier 16 No AG.AGSCAT Quomain Record Qualifier	ItemOID	KeySequ	MethodO I.	Role	RoleC	OrderNumber	Mandat	Collecti	IsNon	HasNo	Item definitions:
© DOMAIN Identifier 2 Yes DOMAIN © USUBJID Identifier 3 Yes DOMAIN © DM.RFSTDTC Record Qualifier 5 No AGAGSEQ © DM.RFSTDTC Record Qualifier 6 No AGAGSEQ © DM.RFSTDTC Record Qualifier 7 No AGAGSEQ © DM.RFXSTDTC Record Qualifier 7 No AGAGNPID © DM.RFXSTDTC Record Qualifier 9 No AGAGNPID © DM.RFCSTDTC Record Qualifier 10 No AGAGINKGRP © DM.RFICDTC Record Qualifier 11 No AGAGCINKGRP © DM.RFICDTC Record Qualifier 12 No AGAGCAT © DM.DTHDTC Record Qualifier 13 No AGAGSCAT © DM.STEID Record Qualifier 15 Yes AGAGSCAT © DM.NRTHC Record Qualifier 16 No AGAGSCAT © DM.NNVID Record Qualifier 16 No AGAGSCAT © DM.NSTHDTC Record Qualifier 17 No AGA				Identifier		1	Yes				STUDYID
QUSUBJID Identifier 3 Yes Justice QDM.SUBJID Topic 4 Yes AG.AGSEQ QDM.RFSTDTC Record Qualifier 5 No AG.AGSEQ QDM.RFSNDTC Record Qualifier 6 No AG.AGSPID QDM.RFSNDTC Record Qualifier 7 No AG.AGSPID QDM.RFCSTDTC Record Qualifier 9 No AG.AGLNKID QDM.RFCSTDTC Record Qualifier 9 No AG.AGLNKGRP QDM.RFCSTDTC Record Qualifier 10 No AG.AGCNT QDM.RFICDTC Record Qualifier 11 No AG.AGCAT QDM.DTHDTC Record Qualifier 13 No AG.AGSCAT QDM.STEID Record Qualifier 16 No AG.AGSCAT QDM.INVID Record Qualifier 16 No AG.AGSTAT QDM.BRTHDTC Record Qualifier 18 No AG.AGCCUR QDM.BRTHDTC Record Qualifier 19 No AG.AGCCUR QDM.BRTHDTC Record Qualifier 19 No AG.AG				Identifier		2	Yes				DOMAIN
CM.DM.SUBJID Topic 4 Yes AG.AGSEQ CM.DM.RFSTDTC Record Qualifier 5 No AG.AGSEQ CM.DM.RFSTDTC Record Qualifier 6 No AG.AGSEQ CM.DM.RFXSTDTC Record Qualifier 7 No AG.AGSEQ CM.DM.RFXSTDTC Record Qualifier 7 No AG.AGSEQ CM.DM.RFXENDTC Record Qualifier 9 No AG.AGSEQ CM.DM.RFCENDTC Record Qualifier 10 No AG.AGNCHKID CM.DM.RFCENDTC Record Qualifier 11 No AG.AGNCHKID CM.DM.RFICDTC Record Qualifier 11 No AG.AGCAT CM.DD.THFL Record Qualifier 13 No AG.AGSCAT CM.DM.STEID Record Qualifier 16 No AG.AGSCAT CM.DM.BRTHDTC Record Qualifier 18 No AG.AGSCAT CM.DM.BRTHDTC Record Qualifier 19 No AG.AGSCAT CM.DM.BRTHDTC Record Qualifier 19 No AG.AGCLAS CM.DM.AGE Record Qualifier 20 </td <td></td> <td></td> <td></td> <td>Identifier</td> <td></td> <td>3</td> <td>Yes</td> <td></td> <td></td> <td></td> <td>USUBJID</td>				Identifier		3	Yes				USUBJID
Contraction Record Qualifier 5 No Contraction Record Qualifier 6 No AGAGGRPID Contraction Record Qualifier 6 No AGAGGRPID Contraction Record Qualifier 7 No AGAGGRPID Contraction Record Qualifier 7 No AGAGGRPID Contraction Record Qualifier 9 No AGAGGRPID Contraction Record Qualifier 10 No AGAGGRPID Contraction Record Qualifier 10 No AGAGGRPID Contraction Record Qualifier 11 No AGAGGRPID Contraction Record Qualifier 12 No AGAGGRPID Contraction Record Qualifier 13 No AGAGGRPID Contraction Record Qualifier 14 No AGAGGRPID Contraction Record Qualifier 16 No AGAGGRPID Contraction Record Qualifier 18 No AGAGGRPID Contraction Record Qualifier 19 No AGAGG	🖺 🔍 DM.SUBJID			Topic		4	Yes				AG AGSEO
Contraction Record Qualifier 6 No Contraction Record Qualifier 7 No AG.AGSPID Contraction Record Qualifier 7 No AG.AGSPID Contraction Record Qualifier 8 No AG.AGLNKID Contraction Record Qualifier 9 No AG.AGLNKID Contraction Record Qualifier 10 No AG.AGLNKID Contraction Record Qualifier 11 No AG.AGLNKID Contraction Record Qualifier 12 No AG.AGCAT Contraction Record Qualifier 13 No AG.AGCAT Contraction Record Qualifier 15 Yes AG.AGCAT AG.AGCCUR Record Qualifier 16 No AG.AGSCAT AG.AGE Record Qualifier 18 No AG.AGSTAT AG.AGE Record Qualifier 19 No AG.AGCLAS AG.AGE Record Qualifier 19 No AG.AGCLAS AG.AGE Record Qualifier 19 No AG.AGCLAS	DM.RFSTDTC			Record Qualifier		5	No				
Q.DM.RFXSTDTC Record Qualifier 7 No Q.DM.RFXSTDTC Record Qualifier 8 No AG.AGSPID Q.DM.RFXSTDTC Record Qualifier 9 No AG.AGLNKID Q.DM.RFCSTDTC Record Qualifier 10 No AG.AGLNKGRP Q.DM.RFCENDTC Record Qualifier 11 No AG.AGDNIY Q.DM.RFICDTC Record Qualifier 12 No AG.AGCAT Q.DM.DTHDTC Record Qualifier 13 No AG.AGSCAT Q.DM.SITEID Record Qualifier 16 No AG.AGSCAT Q.DM.INVID Record Qualifier 16 No AG.AGSCAT Q.DM.INVID Record Qualifier 16 No AG.AGSCAT Q.DM.INVINAM Synonym Qualif 17 No AG.AGSTAT Q.DM.AGE Record Qualifier 19 No AG.AGCLAS Q.DM.SX Record Qualifier 21 Yes AG.AGCLAS Q.DM.RACE Record Qualifier 23 No AG.AGCDSEX	DM.RFENDTC			Record Qualifier		6	No				
Contraction Record Qualifier 8 No Contraction Record Qualifier 9 No AG.AGLNKID Contraction Record Qualifier 9 No AG.AGLNKID Contraction Record Qualifier 10 No AG.AGLNKIGRP Contraction Record Qualifier 11 No AG.AGLNKIGRP Contraction Record Qualifier 11 No AG.AGLNKIGRP Contraction Record Qualifier 12 No AG.AGLNKID Contraction Record Qualifier 13 No AG.AGDNEY Contraction Record Qualifier 14 No AG.AGDNEY Contraction Record Qualifier 15 Yes AG.AGNESP Contraction Record Qualifier 16 No AG.AGNESP Contraction Record Qualifier 18 No AG.AGNESP Contraction Record Qualifier 19 No AG.AGCLAS Contraction Record Qualifier 21 Yes AG.AGCLAS AG.AGDOSE Record Qualifier 22 No				Record Qualifier		7	No				AG.AGSPID
AGAGLNKGRP AGAGCLNKGRP AGAGCLNKGRP <tr< td=""><td>DM.RFXENDTC</td><td></td><td></td><td>Record Qualifier</td><td></td><td>8</td><td>No</td><td></td><td></td><td></td><td>AG.AGLNKID</td></tr<>	DM.RFXENDTC			Record Qualifier		8	No				AG.AGLNKID
Momerce NDTC Record Qualifier 10 No AG.AGTRT Momorphic NDM.RFICDTC Record Qualifier 11 No AG.AGTRT Momorphic NDM.RFICDTC Record Qualifier 12 No AG.AGTRT Momorphic NDM.RFICDTC Record Qualifier 12 No AG.AGTRT Momorphic NDTHDTC Record Qualifier 13 No AG.AGCAT Momorphic NDM.DTHFL Record Qualifier 15 Yes AG.AGCAT Momorphic NDM.NVID Record Qualifier 16 No AG.AGCAT Momorphic NDM.INVID Record Qualifier 17 No AG.AGSCAT Momorphic NDM.RCE Record Qualifier 19 No AG.AGSCAT Momorphic NDM.RCE Record Qualifier 19 No AG.AGCCUR Momorphic NDM.SEX Record Qualifier 20 No AG.AGCLAS Momorphic NDM.SEX Record Qualifier 21 Yes AG.AGODSE Momorphic NDM.ETHNIC Record Qualifier 23 No AG.AGDOSE AG.AGDOSE				Record Qualifier		9	No				AG.AGLNKGRP
Contraction Record Qualifier 11 No No AG.AGMODIFY Contraction Record Qualifier 12 No AG.AGDECOD Contraction Record Qualifier 13 No AG.AGDECOD Contraction Record Qualifier 13 No AG.AGCAT Contraction Record Qualifier 14 No AG.AGCAT Contraction Record Qualifier 15 Yes AG.AGCAT Contraction Record Qualifier 16 No AG.AGCAT Contraction Record Qualifier 16 No AG.AGCAT AG.AGDECOUR Record Qualifier 16 No AG.AGCCUR AG.AGEE Record Qualifier 18 No AG.AGCLUR AG.AGEL Variable Qualifier 19 No AG.AGCLAS AG.AGCLAS Record Qualifier 21 Yes AG.AGOSEXT AG.AGDOSE Record Qualifier 23 No AG.AGOSEXT				Record Qualifier		10	No				AG.AGTRT
Q. RP. RFPENDTC Record Qualifier 12 No AG.AGDECOD Q. DM.DTHDTC Record Qualifier 13 No AG.AGDECOD Q. DM.DTHFL Record Qualifier 14 No AG.AGCAT Q. DM.DTHFL Record Qualifier 14 No AG.AGCCAT Q. DM.STEID Record Qualifier 15 Yes AG.AGPRESP Q. DM.INVID Record Qualifier 16 No AG.AGPRESP Q. DM.BRTHDTC Record Qualifier 18 No AG.AGSCAT Q. DM.BRTHDTC Record Qualifier 19 No AG.AGSTAT Q. DM.AGE Record Qualifier 19 No AG.AGCLAS Q. DM.SEX Record Qualifier 21 Yes AG.AGCLAS Q. DM.RCE Record Qualifier 22 No AG.AGDOSE Q. DM.RTHNIC Record Qualifier 23 No AG.AGDOSE	C OM.RFICDTC			Record Qualifier		11	No				AG.AGMODIFY
Image: Split Strength of Split Stre	RP.RFPENDTC			Record Qualifier		12	No				AG.AGDECOD
Contraction Record Qualifier 14 No AGAGSCAT Contraction Record Qualifier 15 Yes AGAGSCAT Contraction Record Qualifier 16 No AGAGSCAT Contraction Record Qualifier 16 No AGAGSCAT Contraction Record Qualifier 16 No AGAGSCAT Contraction Record Qualifier 17 No AGAGSCAT Contraction Record Qualifier 18 No AGAGSCAT Contraction Record Qualifier 19 No AGAGSCAT Contraction Record Qualifier 20 No AGAGCLAS Contraction Record Qualifier 21 Yes AGAGCLAS Contraction Record Qualifier 22 No AGAGODSETVE Contraction Record Qualifier 23 No AGAGODSETVE	C OM.DTHDTC			Record Qualifier		13	No				AG AGCAT
Image: Construction of the second Qualifier 15 Yes AG.AdSCAT Image: Construction of the second Qualifier 16 No AG.AdSCAT Image: Construction of the second Qualifier 16 No AG.AdSCAT Image: Construction of the second Qualifier 17 No AG.AdSCAT Image: Construction of the second Qualifier 17 No AG.AdSCAT Image: Construction of the second Qualifier 18 No AG.AdSCAT Image: Construction of the second Qualifier 19 No AG.AdSCAT Image: Construction of the second Qualifier 20 No AG.AdSCAT Image: Construction of the second Qualifier 21 Yes AG.AdSCLAS Image: Construction of the second Qualifier 22 No AG.AdSCLAS Image: Construction of the second Qualifier 23 No AG.AdSCLAS	🖺 🔍 DM.DTHFL			Record Qualifier		14	No				AC ACSCAT
Construction Record Qualifier 16 No AG.AGPRESP Construction Synonym Qualif 17 No AG.AGPRESP Construction Synonym Qualif 17 No AG.AGPRESP Construction Record Qualifier 18 No AG.AGPRESP Construction Record Qualifier 19 No AG.AGPRESP Construction Record Qualifier 19 No AG.AGREASND Construction Record Qualifier 20 No AG.AGREASND Construction Record Qualifier 21 Yes AG.AGCLAS Construction Record Qualifier 22 No AG.AGDOSE Construction Record Qualifier 23 No AG.AGDOSE	🖺 🔍 DM.SITEID			Record Qualifier		15	Yes				AGAGSCAT
Image: Construction of the synthetic of the synthet synthetic of the synthetic of the synthet	🖺 🔍 DM.INVID			Record Qualifier		16	No				AG.AGPRESP
Contraction Record Qualifier 18 No AG.AGSTAT Contraction Record Qualifier 19 No AG.AGSTAT Contraction Record Qualifier 19 No AG.AGSTAT Contraction Variable Qualifier 20 No AG.AGCLAS Contraction Record Qualifier 21 Yes AG.AGCLAS Contraction Record Qualifier 22 No AG.AGCLAS Contraction Record Qualifier 22 No AG.AGCLAS Contraction Record Qualifier 23 No AG.AGDOSETURE	DM.INVNAM			Synonym Qualif		17	No				AG.AGOCCUR
No AG.AGREASND No No				Record Qualifier		18	No				AG.AGSTAT
CADMAGEU Variable Qualifier 20 No AG.AGCLAS CODM.SEX Record Qualifier 21 Yes AG.AGCLASCD CODM.RACE Record Qualifier 22 No AG.AGDOSE CODM.ETHNIC Record Qualifier 23 No AG.AGDOSE	DM.AGE			Record Qualifier		19	No				AG.AGREASND
Construction Record Qualifier 21 Yes AG.AGCLASCD Construction Record Qualifier 22 No AG.AGCLASCD AG.AGCLASCD AG.AGCDSE AG.AGCD	DM.AGEU			Variable Qualifier		20	No				AG.AGCLAS
AG.AGDOSE Record Qualifier 22 No AG.AGDOSE AG.AGDOSE CONTRACT RECORD QUALIFIER 23 No AG.AGDOSE AG.AGDOSE	DM.SEX			Record Qualifier		21	Yes				AG.AGCLASCD
CALCONSTANT CALCONSTA	DM.RACE			Record Qualifier		22	No				AG AGDOSE
	DM.ETHNIC			Record Qualifier		23	No				AC ACDOSTVT
	DM.ARM			Synonym Qualif		25	No				

showing which variables belong to "Demographics" with a list of available variables on the right side. For other domains than DM, this will allow us later to e.g. add additional timing variables to any "Findings", "Event" or "Interventions" domain/dataset.

One also sees that the "Mandatory" field has been filled. Its value is "Yes" for the case that the SDTM variable is "required", and to "No" when the SDTM variable is "expected" or "permissible"¹⁰.

Remark that "KeySequence" is not populated, as the assignment of the table keys is a **user decision**. This is important to realize: it is up to you to decide how your SDTM/SEND/ADaM table will be organized and which combination of variables will guarantee uniqueness of the records. In the case of DM, the typical assignments for KeySequence is "1" for "STUDYID" and "2" for "USUBJID".

Going back to the main window and clicking the "magnifying glass" on e.g. "DM", the following window is displayed:

×

¹⁰Essentially, the designation "Expected" is an horror. A better designation would have been "conditionally required".

~

Contents of ItemGroupDef with OID DM and with Name DM

Attributes:

OID DM Name DM Repeating Yes
Name DM Repeating Yes
Repeating Yes
IsReferenceData No
SASDatasetName DM
Domain DM
Origin
Role
Purpose Tabulation
Comment
Structure One record per subject
ArchiveLocationID Location.DM
StandardOID STD.SDTMIG-3.4
IsNonStandard
HasNoData
CommentOID

OK Cancel

and when scrolling to the bottom:

×

DM.ACTARMCD					
	ACTARMCD				Record Qualifier
DM.ACTARM	ACTARM				Synonym Qualifier
DM.ARMNRS	ARMNRS				Record Qualifier
DM.ACTARMUD	ACTARMUD				Record Qualifier
DM.COUNTRY	COUNTRY				Record Qualifier
DM.DMDTC	DMDTC				Timing
DM.DMDY	DMDY				Timing
ontent for Alias					
In the second se	(ODM extension) SubClass SE DDM extension) ref title Lxpt DM.xpt				

Also let us have a look at the list of controlled terminology, by clicking the "CodeLists" tab:

Annotated CRFs	Supplemental Documents	ValueList Definitions	WhereClause Definitions	Dataset Definitions	Variable Definitions	Codelists	Method De
OID			Name			DataType	
CL.MEDDRA			MedDRA Adverse Events [Dictionary		text	
CL.ISO3166			Country Codes			text	
CL.C141657.	TENMW1TC		10-Meter Walk/Run Functi	onal Test Test Code		text	
CL.C141656.	TENMW1TN		10-Meter Walk/Run Functi	onal Test Test Name		text	
CL.C141663./	A4STR1TC		4-Stair Ascend Functional	Test Test Code		text	
CL.C141662./	A4STR1TN		4-Stair Ascend Functional	Test Test Name		text	
CL.C141661.I	D4STR1TC		4-Stair Descend Functiona	al Test Test Code		text	
CL.C141660.I	D4STR1TN		4-Stair Descend Functiona	al Test Test Name		text	
CL.C115388.	SIXMW1TC		6 Minute Walk Functional	Test Test Code		text	
CL.C115387.	SIXMW1TN		6 Minute Walk Functional	Test Test Name		text	
CL.C182464./	AIMS0101T07OR		Abnormal Involuntary Move	ement Scale Clinical Cl	assification ORRES f	text	
CL.C182465./	AIMS0108T09OR		Abnormal Involuntary Move	ement Scale Clinical Cl	assification ORRES f	text	
CL.C182466./	AIMS0110OR		Abnormal Involuntary Move	ement Scale Clinical Cl	assification ORRES f	text	
CL.C182467./	AIMS0111T12OR		Abnormal Involuntary Move	ement Scale Clinical Cl	assification ORRES f	text	
CL.C182502./	AIMS0101T07STR		Abnormal Involuntary Move	ement Scale Clinical Cl	assification STRESC f	text	
CL.C182503./	AIMS0108T09STR		Abnormal Involuntary Move	ement Scale Clinical Cl	assification STRESC f	text	
CL.C182504./	AIMS0110STR		Abnormal Involuntary Move	ement Scale Clinical Cl	assification STRESC f	text	
CL.C182505./	AIMS0111T12STR		Abnormal Involuntary Move	ement Scale Clinical Cl	assification STRESC f	text	
CL.C101805./	AIMS01TC		Abnormal Involuntary Move	ement Scale Clinical Cl	assification Test Code	text	
CL.C101806./	AIMS01TN		Abnormal Involuntary Move	ement Scale Clinical Cl	assification Test Name	text	
CL.C66767.A	CN		Action Taken with Study Tr	reatment		text	
CL.C101865./	ACSPCAT		Acute Coronary Syndrome	Presentation Category		text	
CL.C182484./	APCH101OR		Acute Physiology and Chro	onic Health Evaluation I	Clinical Classificatio	text	
CL.C182485./	APCH102OR		Acute Physiology and Chro	onic Health Evaluation I	Clinical Classificatio	text	
CL.C182486./	APCH103OR		Acute Physiology and Chro	onic Health Evaluation I	Clinical Classificatio	text	
CL.C182487.	APCH104OR		Acute Physiology and Chro	onic Health Evaluation I	Clinical Classificatio	text	
CL.C182488./	APCH105AOR		Acute Physiology and Chro	onic Health Evaluation I	Clinical Classificatio	text	
CL.C182489./	APCH105BOR		Acute Physiology and Chro	onic Health Evaluation I	Clinical Classificatio	text	
CL.C182490.	APCH106AOR		Acute Physiology and Chro	onic Health Evaluation I	Clinical Classificatio	text	
CL.C182491.	APCH106BOR		Acute Physiology and Chro	onic Health Evaluation I	Clinical Classificatio	text	
CL.C182492.	APCH107OR		Acute Physiology and Chro	onic Health Evaluation I	Clinical Classificatio	text	
CI C182493	APCH108OR		Acute Physiology and Chr	onic Health Evaluation I	Clinical Classificatio	Itext	

For each loaded codelist, it displays the identifier (the OID), the name of the codelist, and the datatype (usually "text)". Clicking the "+" icon e.g. for "CL.C74456.LOC" ("Anatomical Location") and then selecting the "CodeListItem" tab leads to:

Extra information for: CodeList, with OID = CL.C74456.LOC

2	
- K -	

?	Description Co	deListItem	ExternalCodeList	Enumerate	ditem Ali	as	
_	CodedValue				Rank		OrderNum
	STH LUMBAR	SPINOUS PR	OCESS				
	ABDOMINAL A	AORTA					
	ABDOMINAL (CAVITY					
	ABDOMINAL L	YMPH NODE					
	ABDOMINAL F	REGION					
	ABDOMINAL S	SKIN					
	ABDOMINAL V	NALL					
	ABDUCENS N	IERVE					
	ABDUCTOR E	DIGITI MINIMI N	MUSCLE OF THE HAI	ND DV			
	ABDUCTOR E	DIGITI QUINTI	MUSCLE				
	ABDUCTOR H	HALLUCIS MU	SCLE				
	ABDUCTOR F	POLLICIS BRE	VIS MUSCLE				
	ABDUCTOR F	POLLICIS LON	IGUS MUSCLE				
	ACCESSORY	RENAL ARTE	RY				

One also sees that there is still underlying information for each of the coded values, as the "+" icon is colored yellow. Clicking on the "+" icon for e.g. "ABDOMINAL AORTA" leads to:

Extra information for: EnumeratedItem

Name
C32038
-

also providing the NCI code.

Unfortunately, CDISC publishes most of its Controlled Terminology as "EnumeratedItem" codelists, so without "Decode" and "TranslatedText" possibilities, ignoring that there are other languages than English in the world.

One can however transform an "EnumeratedItem" CodeList into a "CodeListItem" codelist (the latter allowing other languages) by using the menu "Transform - EnumeratedItem CodeList to CodeListItem CodeList"):

⊱ Define.xml Designer 2022 by XML4Pharma

File Edit Add	Transform	Validate	View	Extra	Options	Help
i 🚘 📑 🤞	Enumerate	d Codelist t	to Code	ListItem	Codelist	
	CodeList to	ValueList				

The system will then ask to select and codelist which is then transformed. The new OID is then the OID of the selected CodeList plus the suffix ".NEW".

For our "Body Location" CodeList CL.C74456.LOC, we then get a new CodeList with OID

CL.C74556.LOC.NEW (one can of course later change the OID), which also has "Decodes". Editing it by clicking the "+" then allows us to e.g. add a Japanese term:

⊱ Extra	information fo	or: CodeList, with OID	= CL.C74456.LOC.NEW			
?	Description	CodeListItem	ExternalCodeList En	umerateditem Alias		
_	Coded	Value	Rank	OrderNumbe	ar -	ExtendedValue
	📮 ۶ Extra	information for: Cod	leListItem - Coded Value = /	ABDOMINAL AORTA		
	2	Decode Alias				
			Language			Translated Text
		en			ABDOMINAL AORTA	
		ja			腹部大動脈	

Before going into detail about other features, we will first look how to create a define.xml file from an existing set of SAS-XPT files.

Creating a define.xml starting from an existing set of SAS-XPT files

In some cases, sponsors need to generate the define.xml "post SDTM", "post SEND" or "post ADaM". Although this is not a best practice, it is an existing one, and the "Define.xml Designer" has a lot of features and wizards to bring this to a good end, this in contradiction to "black box" software that starts from an Excel worksheet and usually leads to disaster, as the users do not understand what they are doing.

After using the "File - New Define.xml", select the checkbox "I want to start from a set of SAS-XPT files":

⊱ New Study Metadata	×
Define-XML version: 2.1.0	
I want to start from a CDISC SDTM/SEND/ADaM template	
define_template_ADaMIG_1_3.xml	^
define_template_SDTMIG_3.1.2_SDTM_1.2.xml	
define_template_SDTMIG_3.1.2_SDTM_1.2_oncology_draft.xml	=
define_template_SDTMIG_3.1.2_SDTM_1.2_PGx_new.xml	
define_template_SDTMIG_3.1.3_Med_Devices.xml	
define_template_SDTMIG_3.1.3_SDTM_1.3.xml	
define_template_SDTMIG_3.1.3_SDTM_1.3_Non_Subject_Data.xml	
define_template_SDTMIG_3.2_AssociatedPersons.xml	-
✓ I want to start from a set of SAS-XPT files	
SDTM Browse SAS-XPT	
I want to load by CDISC published Controlled Terminology	
SDTM_Terminology_2021-12-17.xml	^
SDTM_Terminology_2022-03-25.xml	
SDTM_Terminology_2022-06-24.xml	
SEND Terminology 2013-04-12.xml	

Then select the model, which can either be "SDTM" or "SEND" or "ADaM" or "Other". We will here demonstrate the functionality for the case of an SDTM define.xml.

It is also very wise to then select a CDISC-CT version. This will allow the system to compare the distinct values of coded items in your SAS-XPT files with the CDISC codelists, and set up CDISC-compliant subset codelists.

When clicking the "Browse SAS-XPT" button, a dialog appears:



This allows you to select all SAS-XPT files from a single directory, or a selected set of SAS-XPT files. When using the first option, you will only be able to select a directory, when using the second option, you will need to select one or more SAS-XPT files from a directory. For example for the first option:

⊱ Open	×
Look In:	tabulations
📑 sdtm 📑 sdtm - Kop	bie
Folder <u>n</u> ame:	ubmission Package\900172\m5\datasets\cdiscpilot01\tabulations\sdtm
Files of <u>T</u> ype:	Alle Dateien
	Open Cancel

And after clicking "Open":

You have cho from the sma	osen to generate a template define.xml all amount of metadata delivered from the following SAS-XPT files:
Directory: D:	CDISC_Standards\LZZT_updated_pilot_submission_package_2013\Updated Pilot Submission Package\900172\m5\datasets\cdiscpilot01\tabulations\sdtm
dm.xpt	se.xpt
sv.xpt	cm.xpt
ex.xpt	ae.xpt
ds.xpt	mh.xpt
lb.xpt	qs.xpt
sc.xpt	vs.xpt
ta.xpt	te.xpt
tv.xpt	ti.xpt
ts.xpt	reirec.xpt
suppae.xpt	suppdm.xpt
suppds.xpt	supplb.xpt
Please be av For each vari In almost all You will how the protocol,	vare that SAS-XPT files do only contain a minimum of metadata which is not granular at all. able, the system will assign a data type, the maximal length, and when possible, the associated codelist. cases, these assignments will be correct, but it surely is a good idea to check them. ever still define the valuelevel metadata, based on your CRFs or data collection tools, and any other sources of metadata

Explaining that the challenge of such an approach is that the SAS-XPT datasets only contain a minimum amount of metadata, and that you will need to get metadata from several sources, especially from the CRFs (in the case of SDTM) and from the source operational database.

After clicking OK:

✓ I want to start from a set of SAS-XPT files	
SDTM Browse SAS-XPT	
✓ I want to load by CDISC published Controlled Terminology	
SDTM_Terminology_2024_42_47.yml	- ·
SDTM_Termi Analyzing SAS-XPT files — — X	
SDTM_Termi	
SEND_Termir	
SEND_Terminology_2013-06-28.xml	
SEND_Terminology_2013-10-04.xml	
SEND_Terminology_2013-12-20.xml	
SFND Terminology 2014-03-28.xml	-

The system starts analyzing the SAS-XPT files¹¹ to generate a first primitive prototype of the define.xml. After that, it is a very good idea to select a version of the CDISC controlled terminology, and to add the general information that goes into the define.xml:

p. 28

¹¹ At this stage, it only analyzes the "header" part of the SAS-XPT files, so not the complete contents. Therefore, it is fast, but the generated metadata is not very precise.

✓ I want to start from a set of SAS-XPT files	
SDTM Browse SAS-XPT	
✓ I want to load by CDISC published Controlled Terminology	
SDTM_Terminology_2021-12-17.xml	
SDTM_Terminology_2022-03-25.xml	
SDTM_Terminology_2022-06-24.xml	
SEND_Terminology_2013-04-12.xml	
SEND_Terminology_2013-06-28.xml	
SEND_Terminology_2013-10-04.xml	
SEND_Terminology_2013-12-20.xml	
SEND_Terminology_2014-03-28.xml	•

Below this part of the wizard, one will find some more checkboxes¹²:

	✓ I want to load by CDISC published Controlled Terminology	
	SDTM_Terminology_2021-03-26.xml	
	SDTM_Terminology_2021-06-25.xml	
	SDTM_Terminology_2021-09-24.xml	
	SDTM_Terminology_2021-12-17.xml	
	SDTM_Terminology_2022-03-25.xml	
	SDTM_Terminology_2022-06-24.xml	
	SEND_Terminology_2013-04-12.xml	
	SEND_Terminology_2013-06-28.xml	•
	Generate Define-XML Variable DataType, Length and SignificantDigits from XPT content	
/	Try to create subset CodeLists from XPT content and selected Controlled Terminology	
1	Try to create sponsor-defined CodeLists from definitions in a 'sponsorcodelistvariables.dat' f	file
	Try to create Valuelists for Supplemental Qualifier datasets from XPT content	
	Try to create Valuelists from definitions in a 'valuelistvariables.dat' file	
l	Study OID (required)	

We will explain them here in detail, although there is a tooltip for each of them:

Try to create subset CodeLists from XPT content and selected Controlled Terminology	
Try to create Valuelists for Supplemental Qualifier datasets from XPT content	
Try to cl When checked, the system will try to create ValueLists for QVAL-QNAM for Supplemental	Qualifier datasets
Study OID (required)	-

The first one "Generate Define-XML Variable Data Type", "Length and SignificantDigits from XPT content", will, when checked, tell the software to generate more granular data types for the variable definitions, and adapt the values for "Length" and "SignificanDigits" (in case the data type is "float").

When this checkbox is <u>not</u> checked, the software will only inspect the header of the XPT files, and assign the datatype as "text" when the information in the header states that the variable is of type "char", and assign the datatype as "integer" when the information in the header states that the variable is of type "num" (numeric). It will also assign the lengths as indicated in the header (as 8 for numeric variables). So if the header says that the variable length for VSBLFL (baseline flag) is

¹² This is a new feature of the Define-XML Designer version 2022.

200, then 200 will be assigned¹³, even when the only value in that column is "Y". However, only reading the header is very fast.

When the checkbox is checked, a thorough analysis of the contents is performed, and the longest string value for variable values determined, and set to the "Length" of the variable in the define.xml. So, even when the header states "200" for the length of e.g. VSBLFL, but the only value found is "Y", then the Length in the define.xml will be set to "1".

For numeric variables, it will be checked whether the value is a float or an integer, and the "Length" and "SignificantDigits" (the latter in the case of "float") as defined in the Define-XML specification, will be determined.

Analyzing the full contents of the SAS-XPT files is however computing intensive, especially for large XPT files. So this process can take a few minutes.

The second checkbox "**Try to create subset CodeLists from XPT content and selected Controlled Terminology**" does exactly what it says. For each variable to which there is controlled terminology according to the Implementation Guide (IG), the system will retrieve the distinct values from that column from the XPT file, and create a codelist for the variable. This codelist is then compared with the CDISC one, and additional information, like the NCI code is added. If a distinct value is not found in the CDISC codelist, it is marked as an extension to the codelist, using the def:ExtendedValue="Yes" attribute.

In case of Define-XML 2.1, also the attribute def:StandardOID is added. This will be explained further on.

The newly generated codelist is then assigned to the corresponding variable definition (ItemDef). On one hand, this feature is very useful, allowing to obtain relative small codelists that reflect what has been captured, but at the other hand, it can be pretty tricky when does not think about the implications of this approach:

Imagine for example, that on the CRF, you had "Mild", "Moderate" and "Severe" for the Adverse Event Severity, but there was never a severe AE, so you XPT dataset will not have a value "SEVERE" for "AESEV". In that case, the generated codelist will only contain "MILD" and "MODERATE". The submission codelist must however reflect what was **planned**, in this case reflect the CRF, so also "SEVERE" must then be added to the codelist. As the system cannot know what exactly was planned, checking the generated codelist for completeness (and correctness) is the task of the user.

Also new is the checkbox "Try to create sponsor-defined CodeLists from definitions in a 'sponsorcodelistvariables.dat' file".

This file needs to reside in the folder where also the software is located. Here is an example of the contents of the file:

```
 *sponsorcodelistvariables.dat - Editor

Datei Bearbeiten Format Ansicht Hilfe

# we will generate sponsor codelists for all --CAT variables
--CAT

# and for following specific variables
ARMCD ARM
ETCD ELEMENT
EXTRT
```

¹³ The fixed-length format for variables is one of the major problems of the XPT format. It also means that if the string length is less than what is provided in the header, the remaining of the field for that variable is filled with blanks. This is why SAS Transport is such an inefficient format. Reason is that it is one step beyond the punch card ...

It states that it wants the software to generate sponsor-defined codelists for all --CAT variables, independent of the domain, and furthermore sponsor-defined codelists for ARMCD, ARM, ETCD, ELEMENT, and EXTRT.

Remark that lines starting with a "#" are comment lines, which will be skipped by the software.

When this checkbox is checked, the result for e.g. ARMCD and ARM (in DM) will e.g. be:

length State State

Attributes:					
Name		Value			
OID	CL.DM.ARMCD				
Name	Sponsor-defined C	odeList for variable ARN	4CD		
DataType	text				
SASFormatNar	ne				
StandardOID					
IsNonStandard	Yes				
CommentOID					
No information Content for Co	leListItem				
No information Content for Co No information Content for Ext No information	leListItem ernalCodeList				
No information Content for Con No information Content for Ext No information Content for Em	leListItem ernalCodeList umeratedItem			1	
No information Content for Co No information Content for Ext No information Content for Em CodedValue	leListItem ernalCodeList umeratedItem Rank OrderNumber	ExtendedValue Alia	s Description		
No information Content for Content for Content for Ext No information Content for Ext No information Content for Ent CodedValue	leListItem ernalCodeList umeratedItem Rank OrderNumber	ExtendedValue Alia	s Description		
No information Content for Content for Content for Content for Ext No information Content for Ent CodedValue	leListItem ernalCodeList umeratedItem Rank OrderNumber	ExtendedValue Alia	s Description		

One sees that for Define-XML 2.1, the "def:IsNonStandard" is set to "Yes", as it is a sponsordefined codelist and not a CDISC codelist.

The fourth checkbox "**Try to create ValueLists for Supplemental Qualifier datasets from XPT content**", will inspect the SUPPxx datasets, and analyze the QNAM and QVAL values. For each distinct QNAM/QLABEL, it will retrieve the distinct values of QVAL, and assign a data type to the latter. For example for QNAM=FSYMPDAT with QLABEL="Date of first symptoms", it will only find dates for QVAL, and then assign the datatype "date" and use that for generating the

 \times

ValueList. When the assigned datatype is "text", it will generate a codelist with the distinct values of QVAL, and use that in the valuelist.

For example, for the file SUPPDM, this will lead to:

Contents of element ValueListDef

Attributes:							
Name Value							
OID VL.SUPPDM.QVAL							
Content for Description							
No information							
content for ItemKel						-	1
ItemOID	Item Name	KeySequence	MethodOID	Method Name	ImputationMethodOID	ImputationMethod Name	R
IT.SUPPDM.QVAL.COMPLT24	COMPLT24						
IT.SUPPDM.QVAL.COMPLT8	COMPLT8						
IT.SUPPDM.QVAL.EFFICACY	EFFICACY						
IT.SUPPDM.QVAL.SAFETY	SAFETY						
IT.SUPPDM.QVAL.COMPLT16	COMPLT16						
	1777						ſ

and for example for the properties of "ITT":

?	Contents of ItemDef with OID IT.SUPPDM.QVAL.ITT and with Name ITT
	Attributes:

Name	Value			
OID	IT.SUPPDM.QVAL.ITT			
Name	ITT			
DataType	text			
Length	1			
SignificantDigits				
SASFieldName				
SDSVarName				
Origin				
Comment				
DisplayFormat				
CommentOID				
Content for Description				
TranslatedText				
Language: not assigned Text: Intent to Treat Population Flag				

with its assigned codelist:

?

ļ	Contents of CodeList with OID CL.SUPPDM.QNAM.ITT.IDVARVAL and with Name CodeList for QVAL for QNAM = ITT in dataset SUPPDM								
	Attributes:								
	Name	Value							

OID CL.SUPPDM.QNAM.ITT.IDVARVAL							
Name CodeList for QVAL for QNAM = ITT in dataset SUPPDM							
DataType	tex	text					
SASFormatNa	ASFormatName						
StandardOID	StandardOID						
IsNonStandard Yes							
CommentOID							
Content for Co No information Content for Ex No information	Fontent for CodeListItem Formation Fontent for ExternalCodeList Fontent for ExternalCodeList Formation						
			[]				
CodedValue	Rank	OrderNumber	ExtendedValue	Alias	Description		
Y							
Content for Alias No information							
				ОК	Cancel		

The last checkbox "**Try to create ValueLists from definitions in a 'valuelistvariables.dat' file**" allows to have ValueLists created (at least, the system will try ...) from definitions, which need to be provided in the file "valuelistvariables.dat" in the folder where also the software resides:

Schematron
SDTM
src
standards
📊 temp
XSLT
🥘 .classpath
🧾 license.dat
subsetcodelistvariables.dat
valuelistvariables.dat

The software comes with such a file, which one can edit, or even make different versions for different use cases, but when this feature is used, the definitions will be taken from the "valuelistvariables.dat" file.

Let us look at some example contents:

📗 valuelistvariables.dat - Editor

Datei Bearbeiten Format Ansicht Hilfe VSORRESU WHERE VSTESTCD EQ WEIGHT VSORRESU WHERE VSTESTCD IN SYSBP,DIABP #VSPOS WHERE VSTESTCD NE SYSBP,DIABP VSPOS WHERE VSTESTCD IN SYSBP,DIABP VSORRES WHERE VSTESTCD EQ FRMSIZE VSORRES WHERE VSTESTCD NE FRMSIZE # VSORRES WHERE VSTESTCD NE FRMSIZE # VSORRES WHERE VSTESTCD NE HEIGHT LBORRES WHERE LBCAT NOTIN CHEMISTRY,HEMATOLOGY LBSPEC WHERE LBCAT EQ HEMATOLOGY

Lines that start with a "#" are regarded as comments and will be skipped. This can very well be used for switching off some definitions.

The format is very similar to what one sees in the browser when displaying a define.xml in the section of "Value Level Metadata". It starts with the variable for which a valuelist will needed to be generated, the keyword "WHERE" followed by the selection condition.

For example, the first line states that a valuelist should be generated for VSORRESU with the condition that the value of VSTESTCD is "WEIGHT".

The second line states that a valuelist should be generated for VSORRESU with the condition that the value for VSTESTCD is either "SYSBP" or "DIABP".

The third line has been commented out, so will be skipped. This was done as the statement does not make much sense: it would generate a valuelist for VSPOS when the value of VSTESTCD is not one of SYSBP or DIABP. As VSPOS is usually only used for blood pressures, this would generate an empty codelist in the valuelist, which is not very useful.

The fourth line shows a classic use of valuelists, generating a valuelist for VSPOS for the case that VSTESTCD is either SYSBP or DIABP.

In the expressions, the following comparators are supported:

- EQ meaning "equals"
- NE meaning "does not equal"
- IN meaning "is in the list of" this requires that it is followed by a list of values, with the values separated by a comma
- NOTIN meaning "is not in the list of "- this requires that it is followed by a list of values, with the values separated by a comma

Remark that the other Define-XML comparators "LT", "LE", "GT" and "GE" are not (yet) supported.

When applied, this will e.g. result in a valuelist with 2 "use cases" for VSORRESU:

Attributes:							
Content for Description No information Content for ItemRef							
ItemOID	Item	Name KeySeq	uence N	MethodOID	Method Name	ImputationMethodOID	I
ItemOID IT.VS.VSORRESU.VSTESTCD.WEIGHT.	Item	Name KeySeq RESU	uence N	MethodOID	Method Name	ImputationMethodOID	

with the properties e.g. for the case of VSORRESU for "weight":

left Contents of element ItemDef

?

Contents of ItemDef with OID IT.VS.VSORRESU.VSTESTCD.WEIGHT.INCLUDE and with Name VSORRESU

Attributes:

Name	Value
OID	IT.VS.VSORRESU.VSTESTCD.WEIGHT.INCLUDE
Name	VSORRESU
DataType	text
Length	2
SignificantDigits	
SASFieldName	
SDSVarName	
Origin	
Comment	
DisplayFormat	
CommentOID	

and with the associated codelist:
CodedValue	Rank	OrderNumber	ExtendedValue	A	lias	Description
				Attr.Name	Attr.Value	
LB				Context	nci:ExtCodeID	
				Attr.Name	Attr.Value	
				Name	C48531	
				Attr.Name	Attr.Value	
ka				Context	nci:ExtCodeID	
кg				Attr.Name	Attr.Value	
				Name	C28252	
Content for A	lias					
Context	Nam	ie				

and as depicted in the browser by the stylesheet:

Controlled Terms

nci:ExtCodeID C66770

Units for Vital Signs Results subset for ValueList [CL.VS.VSORRESU.VSTESTCD.WEIGHT.INCLUDE, C66770]

Permitted Value (Code)	
LB [C48531]	
kg [C28252]	
	_

Units for Vital Signs Results subset for ValueList [CL.VS.VSORRESU.VSTESTCD.SYSBP_DIABP.INCLUDE, C66770]

Permitted Value (Code)	
mmHg [<i>C49670</i>]	

Also here, it is up to the user to carefully control the generated metadata. For example, if the CRF also had the unit choice "stones" for "weight", but it was never used, and thus is not present in the XPT file, it still must be added.

Remark that generating valuelists from XPT file contents is computing intensive, as each applicable XPT file must be analyzed. So, when one of the above features is used, you may want to go for a cup of tea or coffee.

After clicking OK in the "New Study MetaData" wizard, a proposal is made for the "MetaDataVersion OID", "Name" and Description":

5- New MetaDataVersion	23
MetaDataVersion OID (required)	
MV.SDTM	
MetaDataVersion Name (required)	
MetaData for SDTM	
MetaDataVersion Description	
MetaData for SDTM	
ОК	

After clicking "OK", you may say some progress bars showing up, informing you about the progress made for each of the options selected. You may want to go for a cup of tea of coffee ...

When ready, this leads to a first proposal for the variable definitions:

Standards	Annotated CRFs	Supplemental Documents	ValueList Defin	itions Where	Clause Definitions	Dataset Definitions	Variable Definit	ions C
OID		Na	ame	DataType	Length	SignificantDigits	SASFieldName	SDSVan
T.DM.S	TUDYID	ST	UDYID	text	12		STUDYID	
T.DM.D	🔍 IT.DM.DOMAIN			text	2		DOMAIN	
T.DM.U	KIT.DM.USUBJID			text	11		USUBJID	
T.DM.S	UBJID	SL	JBJID	text	4		SUBJID	
T.DM.R	FSTDTC	RF	STDTC	datetime			RFSTDTC	
T.DM.R	FENDTC	RF	ENDTC	datetime			RFENDTC	
IT.DM.R	FXSTDTC	RF	XSTDTC	datetime			RFXSTDTC	
T.DM.R	IT.DM.RFXENDTC			datetime			RFXENDTC	
T.DM.R	FICDTC	RF	ICDTC	datetime			RFICDTC	
T.DM.R	FPENDTC	RF	PENDTC	datetime			RFPENDTC	
T.DM.D	THDTC	DT	THDTC	datetime			DTHDTC	
T.DM.D	THFL	DT	THFL	text	1		DTHFL	
📑 🔍 IT.DM.SI	TEID	SI	TEID	text	3		SITEID	
T.DM.A	GE	AG	E	integer	8		AGE	
T.DM.A	GEU	AG	EU	text	6		AGEU	
T.DM.S	EX	SE	X	text	1		SEX	
T.DM.R	ACE	RA	ACE	text	78		RACE	
T.DM.E	THNIC	ET	HNIC	text	25		ETHNIC	
T.DM.AI	RMCD	AR	RMCD	text	8		ARMCD	
	14	1	114	4 mult	00		ADM	

When the checkbox "Generate Define-XML Variable Type, Lengths and SignificantDigits from XPT contents" was <u>not</u> checked (the "fast way"), one sees that a first estimate of the datatype has been made for each variable, and that the maximum length has been taken from the field definitions within the header of the SAS-XPT files. Also the "label" has been taken from the SAS-XPT file, for example for "RFPENDTC":

⊱ Extra in	nformation for: Iter	mDef, with OII	D = IT.DM.F	RFPENDTC			_		x	
?	RangeCheck	CodeList Re	eference	Role Alia	s Origin	Va	lueList Reference			
-	Descrip	otion	Qu	estion	E	kteri	nalQuestion	MeasurementUnitRef		
			Language	е				Translated Text		
							Date/Time of End of Pa	rticipation		
									-11	
									-11	
									-11	

Remark that no "language" needs to be defined, as the English language is considered as the default language by the Define-XML standard.

For the dataset definitions, we find:

Standards Annotated CRFs Supp			Supplemental Docum	emental Documents ValueList De				reClause Definitio	ns Dataset De	finitions	Variable Definitions
	OID	Name	Repeating	IsRefe	erenceData	SASDatase	etNa	Domain	Origin	Role	Purpose
D Q	IG.DM	DM	No	No		DM		DM			Tabulation
D Q	IG.SE	SE	Yes	No		SE		SE			Tabulation
BQ	IG.SV	SV	Yes	No		SV		SV			Tabulation
<mark>B</mark> Q	IG.CM	CM	Yes	No		CM		CM			Tabulation
D Q	IG.EX	EX	Yes	No		EX		EX			Tabulation
D Q	IG.AE	AE	Yes	No		AE		AE			Tabulation
<mark>B</mark> Q	IG.DS	DS	Yes	No		DS		DS			Tabulation
<mark>B</mark> Q	IG.MH	MH	Yes	No		MH		MH			Tabulation
D Q	IG.LB	LB	Yes	No		LB		LB			Tabulation
D Q	IG.QS	QS	Yes	No		QS		QS			Tabulation
D Q	IG.SC	SC	Yes	No		SC		SC			Tabulation
D Q	IG.VS	VS	Yes	No		VS		VS			Tabulation
D Q	IG.TA	TA	No	Yes		TA		TA			Tabulation
<mark>D</mark> Q	IG.TE	TE	No	Yes		TE		TE			Tabulation
<mark>B</mark> Q	IG.TV	TV	No	Yes		TV		TV			Tabulation
D Q	IG.TI	TI	No	Yes		TI		TI			Tabulation
D Q	IG.TS	TS	No	Yes		TS		TS			Tabulation
<mark>B</mark> Q	IG.RELR	REC RELREC	Yes	No		RELREC		RELREC			Tabulation
BQ	IG.SUPP	PAE SUPPAE	Yes	No		SUPPAE		SUPPAE			Tabulation
D Q	IG.SUPP	DM SUPPDM	Yes	No		SUPPDM		SUPPDM			Tabulation
B S	IG.SUPP	DS SUPPDS	Yes	No		SUPPDS		SUPPDS			Tabulation
BQ	IG.SUPP	LB SUPPLB	Yes	No		SUPPLB		SUPPLB			Tabulation

And when "drilling down" into the details using the "+" icon, e.g. for "VS":

Description	Variable References	Alias C	lass Docur	nent links							
ItemOID	KeySeque	e MethodOID	Imputation	Role	RoleCode	OrderNum	Mandatory	Collection	IsNonSt	HasNoD	Item definitions
C C IT.VS.STUD	YID						Yes				IT.DM.STUDYID
T.VS.DOMA	IN						No				IT.DM.DOMAIN
C SIT.VS.USUB	JID						Yes				IT.DM.USUBJID
T.VS.VSSE	Q						Yes				IT.DM.SUBJID
IT.VS.VSTE	STCD						Yes				IT.DM.RF STDTC
ILVS.VSTE	51						No				IT.DM.RFENDTO
IL SUSPO	5						NO				IT. DM REX STDT
	RES						No				IT DM REXENDT
	RESU						No				IT DM REICDTC
C IT VS VSST	RESN						No				IT DM DEDENDT
IT.VS.VSST	RESU						No				
T.VS.VSST	AT						No				
T.VS.VSLO	С						No				
C CIT.VS.VSBL	FL						No				IT.DW.SITED
T.VS.VISITN	NUM						No				II.DM.AGE
I VS VISIT							No				IT.DM.AGEU

One sees that a good amount of information already has been added, based on both the SAS-XPT files <u>and</u> the knowledge about the SDTM or SEND standard that the system has. Furthermore, everything is transparent, and it is completely clear what the system has generated and what not, this is contradiction to "black box" tools that usually start from Excel worksheets, and that do not allow to inspect what one has been generated.

In case the checkbox "Generate Define-XML Variable Type, Lengths and SignificantDigits from XPT contents" was checked, a full analysis of the contents of each XPT dataset was performed (the "slow" way), and the metadata information is more granular and more precise, for example for LB variables:

	- 1					
D,	Q	IT.LB.STUDYID	STUDYID	text	7	
D,	Q	IT.LB.DOMAIN	DOMAIN	text	2	
D,	Q	IT.LB.USUBJID	USUBJID	text	14	
D,	Q	IT.LB.LBSEQ	LBSEQ	integer	2	
D,	Q	IT.LB.LBREFID	LBREFID	text	7	
D,	Q	IT.LB.LBTESTCD	LBTESTCD	text	6	
D,	Q	IT.LB.LBTEST	LBTEST	text	19	
D,	Q	IT.LB.LBCAT	LBCAT	text	10	
D,	Q	IT.LB.LBORRES	LBORRES	text	8	
D,	Q	IT.LB.LBORRESU	LBORRESU	text	7	
D,	Q	IT.LB.LBORNRLO	LBORNRLO	text	4	
D,	Q	IT.LB.LBORNRHI	LBORNRHI	text	4	
D,	Q	IT.LB.LBSTRESC	LBSTRESC	text	8	
D,	Q	IT.LB.LBSTRESN	LBSTRESN	float	4	1 V
D,	Q	IT.LB.LBSTRESU	LBSTRESU	text	7	
D,	Q	IT.LB.LBSTNRLO	LBSTNRLO	float	3	2
D,	Q	IT.LB.LBSTNRHI	LBSTNRHI	float	3	2
D,	Q	IT.LB.LBSTNRC	LBSTNRC	text	19	· ·
D,	Q	IT.LB.LBNRIND	LBNRIND	text	6	
D,	Q	IT.LB.LBSPEC	LBSPEC	text	5	
D,	Q	IT.LB.LBMETHOD	LBMETHOD	text	8	
D,	Q	IT.LB.LBBLFL	LBBLFL	text	1	
_	- Ch 1					

One sees that for LBBFL (baseline flag) the "Length" has been set to "1", as the only value occurring in the dataset is "Y", even when the SAS-XPT header e.g. defines "8" for the length for that variable.

One also sees that for LBSTRESN, LBSTNRLO and LBSTNRHI, the system found out that the values are of type "float" (that information is not in the XPT header) with "Length" being "4" and "SignificantDigits". And indeed when looking into the XPT file, a typical value for LBSTRESN is "126.4".

Remark that one can also generate more exact and precise values for the "DataType", "Length" and "SignificantDigits" later, using the menu "Extra - Adapt Variable Length from SAS-XPT file contents". This is especially useful for "fine-graining" the define.xml once one has all XPT files as final.

Going back to the panel with the generated variables, one can use the "Show Search Panel" (near the bottom) to look for specific variables:

Move Selected Row Dowll	
Sort by Order Number	
Load from Library	
Show Search Panel	

Clicking the "Show Search Panel" leads to:

						Global Study	Variables Stu	dy Metadata	
Standards	Annotated CRFs	Supplemental Documents	ValueList Definitions	WhereClause	Definitions	Dataset Definitions	Variable Definition	ns Codelists	Method D
			Search fo	:			Searc	h Find Next	Mate
			Search wit	hin: 🗹 All Col	umns				
			✓ OID	🖌 Name	Dat	aType 🖌 Length	Significant)igits 🔽 SASField	IName
			SDSVa	Name 🔽 Origin	Cor	mment 🗾 DisplayFor	mat 🔽 CommentO	ID	
	חואטעד	Na ST	me DataT UDYID text	/pe Len	gth	SignificantDigits	SASFieldName	SDSVarName	Origin

allowing to search in the table with variables. Using the checkboxes one can limit the search to certain fields/columns.

As the amount of metadata in the XPT files is very limited, it is of utmost importance that one carefully inspects what has been generated in the define.xml, and extend, correct and adapt this to reflect as well the collected data as the study design.

For example, the system can not retrieve the exact structure (how it was organized) for each of the XPT datasets¹⁴.

Therefore, we must add additional information at the "Dataset Definition" level:

D	Dataset Definitions Variable Def		initions Cod	lelists	ts Imputation Methods		Presentati	ons	C	ondition Definition	s	Method Definitions	Comment Definitions	Document links	
	Annotated CRFs	Supple	emental Docun	nents		ValueList Defi	initions	١	Nhe	reClause Definition	ns	Includes	Protocol/Trial Design	Study Event De	fi
	OID	Name	Repeating	IsRefer	ence	SASDataset	Domain	O	R	Purpose	C	. Structure	Class	ArchiveLocationID Co	on
D,	G.DM	DM	No	No		DM	DM			Tabulation			SPECIAL PURPOSE	LF.DM	
L D	G.SE	SE	Yes	No		SE	SE			Tabulation			SPECIAL PURPOSE	LF.SE	
∎ <mark>D</mark> ≱	G.SV	SV	Yes	No		SV	SV			Tabulation			SPECIAL PURPOSE	LF.SV	
D,	G.CM	CM	Yes	No		CM	CM			Tabulation			INTERVENTIONS	LF.CM	
D,	🔍 IG.EX	EX	Yes	No		EX	EX			Tabulation			INTERVENTIONS	LF.EX	
□	G.AE	AE	Yes	No		AE	AE			Tabulation			EVENTS	LF.AE	
D,	G.DS	DS	Yes	No		DS DS				Tabulation			EVENTS	LF.DS	
D,	🔍 IG.MH	MH	Yes	No		MH	MH			Tabulation			EVENTS	LF.MH	
l Dj	🔍 IG.LB	LB	Yes	No		LB	LB			Tabulation			FINDINGS	LF.LB	
∎ <mark>D</mark> ≱	G.QS	QS	Yes	No		QS	QS			Tabulation			FINDINGS	LF.QS	
D,	G.SC	SC	Yes	No		SC	SC			Tabulation			FINDINGS	LF.SC	
l 🗅	🔍 IG.VS	VS	Yes	No		VS	VS		Tabulation				FINDINGS	LF.VS	
□	🔍 IG.TA	TA	No	Yes		TA	TA			Tabulation			TRIAL DESIGN	LF.TA	
D,	GIG.TE	TE	No	Yes		TE	TE			Tabulation			TRIAL DESIGN	LF.TE	
l 🗅	G.TV	TV	No	Yes		TV	TV			Tabulation			TRIAL DESIGN	LF.TV	
l Dj	🔍 IG.TI	TI	No	Yes		TI	TI			Tabulation			TRIAL DESIGN	LF.TI	
∎ <mark>D</mark> ≱	GIG.TS	TS	No	Yes		TS	TS			Tabulation			TRIAL DESIGN	LF.TS	
D,	GRELREC	RELREC	Yes	No		RELREC	RELREC			Tabulation			RELATIONSHIP	LF.RELREC	
D,	G.SUPPAE	AE SUPPAE Yes No			SUPPAE	SUPPAE			Tabulation			RELATIONSHIP	LF.SUPPAE		
	G.SUPPDM	JPPDM SUPPDM Yes No			SUPPDM	SUPPDM			Tabulation			RELATIONSHIP	LF.SUPPDM		
I C	G.SUPPDS	SUPPDS SUPPDS Yes No SUPPDS SUPPDS						Tabulation			RELATIONSHIP	LF.SUPPDS			

We see that for each SAS-XPT dataset, a row has been created, and a lot of information has been added, but some is still missing:

In order to find out what, click the "Validate" button in the buttons panel near the bottom:

Copy Selected Row	
Validate	
Reassign OrderNumbers	

The software does a quick "local" validation and soon reports:

¹⁴ This may maybe be possible in a future version of the software.

Validation Results

w = 1:	
Schema error: Structure is a required attribute.	
w = 2:	
Schema error: Structure is a required attribute.	
w = 3:	
Schema error: Structure is a required attribute.	
w = 4:	
Schema error: Structure is a required attribute.	
w = 5:	
Schema error: Structure is a required attribute.	

and in the table with the datasets itself, some cells are colored:

Purpose	С	Structure	Class	ArchiveL
Tabulation			SPECIAL PURPOSE	LF.DM
Tabulation			SPECIAL PURPOSE	LF.SE
Tabulation		Required attribute	PECIAL PURPOSE	LF.SV
Tabulation			INTERVENTIONS	LF.CM
Tabulation			INTERVENTIONS	LF.EX
Tabulation			EVENTS	LF.AE

stating that "Structure" is a required attribute. A quick look in the Define-XML specification learns us that this field contains free text like "one record per vital sign per vital signs measurement per subject". This is of course information that cannot be retrieved from the SAS-XPT file and that needs to be added manually. If the "structure" is not clear from the organization of the SAS-XPT file, you will need to ask the person who generated the file, or look into the documentation of or the code that generated the SAS-XPT dataset. In most cases however, the "structure" can be deduced from the organization of the dataset itself. Remark that this is important information for the reviewer (and that is also why it is a required attribute).

Also remark that the "structure" can deviate considerably from the structure that is proposed in the SDTMIG or SENDIG. So do NOT copy the "structure" from the SDTM-IG or SEND-IG just like that: think yourself!

After having added the "structure" for each dataset, let us "drill down" into the details by clicking the "+" icon. For example for the "Variable References":

information for: ItemGroupDef, v	vith OID = IG.V erences	'S lias Do	ocument lir	ıks					
ItemOID	KeySeq	Method	Imputati	Role	RoleCod	OrderNu	Mandatory	Collection	Item definitions:
T.VS.STUDYID							Yes		IT.DM.STUDYID
T.VS.DOMAIN							No		IT.DM.DOMAIN
IT.VS.USUBJID							Yes		IT.DM.USUBJID
T.VS.VSSEQ							Yes		IT.DM.SUBJID
IT.VS.VSTESTCD							Yes		IT DM RESTOTC
IT.VS.VSTEST							No		IT DM REENDTC
IT.VS.VSPOS							No		IT.DM.RFENDIC
IT.VS.VSORRES							No		II.DM.RFXSTDTC
IT.VS.VSORRESU							No		IT.DM.RFXENDTC
IT.VS.VSSTRESC							No		IT.DM.RFICDTC
IT.VS.VSSTRESN							No		IT.DM.RFPENDTC
IT.VS.VSSTRESU							No		IT.DM.DTHDTC
IT.VS.VSSTAT							No		IT.DM.DTHFL
IT.VS.VSLOC							No		IT.DM.SITEID
IT.VS.VSBLFL							No		IT DM AGE
T.VS.VISITNUM							No		
IT.VS.VISIT							No		IT.DW.AGEU

Clicking the "Validate" button near the bottom again leads to the message:

×



saying us that (in case the define.xml will be used in a regulatory submission), one must assign keys (and their sequence) to the variables. Examples about how this needs to be done are given in the Define-XML specification. An example for "Vital Signs" is:

Extra information for: ItemGroupDef, with OID = IG.VS									
?	Description Variable Reference	Alias Document links							
	ItemOID	KeySeq	Method	Imputati	Role	RoleCod	OrderN		
	IT.VS.STUDYID	1							
	T.VS.DOMAIN								
	T.VS.USUBJID	2							
	T.VS.VSSEQ								
	IT.VS.VSTESTCD	3							
	IT.VS.VSTEST								
	T.VS.VSPOS	6							
	IT.VS.VSORRES								
	IT.VS.VSORRESU								
	IT.VS.VSSTRESC								
	IT.VS.VSSTRESN								
	IT.VS.VSSTRESU								
	T.VS.VSSTAT								
	T.VS.VSLOC								
	T.VS.VSBLFL								
	T.VS.VISITNUM	5							
	T.VS.VISIT								
	T.VS.VISITDY								
	IT.VS.VSDTC	4							
	IT.VS.VSDY								

Do <u>not</u> copy this example without knowing what you are doing! This is an example only! You will need to look into the organization of your clinical database or into the code that generated the dataset in order to find out which were the key variables, and in which order the keys apply.

In case we accidentally assign the same key number twice, and validate again, the validation message becomes:

Description	Variable R	leferences A	lias Docume	nt links		
ItemOID		KeySequence	MethodOID	ImputationMet	Role	RoleCodeL
T.VS.STU	DYID	1				
T.VS.DOM	IAIN					
🖺 🔍 <mark>IT.VS.USU</mark>	IBJID	1				
R AIT VS VSS	EQ					
C SIT Validat	ion Results					

Ordering the dataset definitions

Especially when generating the define.xml starting from SAS-XPT files, the dataset names in the tab "Dataset Definitions" will probably not be in the order FDA reviewers want it to be¹⁵. For SDTM and SEND, the by the "<u>CDISC Metadata Submission Guide</u>" and FDA required order is per class (as in the "def:Class" attribute in the define.xml) and within the class, alphabetically. For ADaM, the suggested order is "Subject Level Analysis Datasets", followed by "Adverse Events Analysis Datasets", followed by "Basic Data Structure" and then "ADaM Other".

The dataset definitions can be ordered manually, using the buttons "Move Row Up" and "Move Row Down", but much more easy is to have it done by the program based on the above rules, using the menu "Extra – Order Dataset definitions alphabetically per class":

View	Extra	Options Help
	Insert C	CRF Page Numbers from annotated CRF
	Genera	te CodeList Subset
	Genera	te CodeList Subset starting from SAS-XPT
otocol	Adapt V	ariable Length from SAS-XPT file contents
	Order D	ataset definitions alphabetically per class

The ordering is done by the program. An example of "before" and "after" is given below:

¹⁵ Essentially, this is a bit ridiculous, as this is only for "ease of review" in the browser, which could easily be managed by a standardized FDA stylesheet. However, the FDA is not capable of developing stylesheets. For machine-readability, the order of the dataset definitions is completely irrelevant.

WhereClause Definiti	ons Dataset [efinitions Va	WhereClause Definition	ons Dataset D	efinitions Va		
A	nnotated CRFs		Annotated CRFs				
OID	Name	Repeating	OID	Name	Repeating		
D QIG.DM	DM	No	📑 🔍 IG.TA	TA	No		
G IG.SE	SE	Yes	📑 🔍 IG.TE	TE	No		
G G.SV	SV	Yes	📴 🔍 IG.TI	TI	No		
G.CM	CM	Yes	📑 🔍 IG.TS	TS	No		
G.EX	EX	Yes	📑 🔍 IG.TV	TV	No		
G.AE	AE	Yes	📑 🔍 IG.DM	DM	No		
G.DS	DS	Yes	📑 🔍 IG.SE	SE	Yes		
G G.MH	MH	Yes	📑 🔍 IG.SV	SV	Yes		
IG.LB	LB	Yes	📑 🔍 IG.CM	CM	Yes		
GQ IG.QS	QS	Yes	📑 🔍 IG.EX	EX	Yes		
G G.SC	SC	Yes	📑 🔍 IG.AE	AE	Yes		
G G.VS	VS	Yes	📑 🔍 IG.DS	DS	Yes		
G.TA	TA	No	📴 🔍 IG.MH	MH	Yes		
G G.TE	TE	No	📑 🔍 IG.LB	LB	Yes		
G G.TV	TV	No	📑 🔍 IG.QS	QS	Yes		
G G.TI	TI	No	G.SC	SC	Yes		
G G.TS	TS	No	📑 🔍 IG.VS	VS	Yes		
G.RELREC	RELREC	Yes	G.RELREC	RELREC	Yes		
G.SUPPAE	SUPPAE	Yes	📑 🔍 IG.SUPPAE	SUPPAE	Yes		
G.SUPPDM	SUPPDM	Yes	G.SUPPDM	SUPPDM	Yes		
G.SUPPDS	SUPPDS	Yes	G.SUPPDS	SUPPDS	Yes		
G.SUPPLB	SUPPLB	Yes	IG.SUPPLB	SUPPLB	Yes		

Adding and subsetting CDISC Controlled Terminology

For most "Findings" domains, you will want to use CDISC controlled terminology. When starting a new define.xml you will be able to load all the CDISC controlled terminology that is needed. When new CDISC controlled terminology is published, you can simply copy the new XML file¹⁶ to the directory, and it will be automatically available.

However, you will very often need to subset the codelists that were provided by CDISC and adapt them for your own study. If you generate the define.xml before you generate your datasets (currently in SAS-XPT format), which is the best practice, you can subset loaded codelists using the menu "Extra - Generate CodeList Subset". If you create your define.xml after you created the SAS-XPT files (not such good practice, but usually the only way in case of legacy datasets), you can compare your SAS datasets with the published controlled terminology and generate a subset of an existing codelist using information from both. The latter is explained in detail in the section "Adding and Subsetting CodeLists from SAS-XPT files".

In case you create your define.xml before generating the SDTM/SEND/ADaM datasets, you will probably subset your codelists with information from the protocol, or from the CRFs.

In the example below, we will subset the codelists for VSTESTCD (Vital Signs Test Code) and VSTEST (Vital Signs Test Name).

If you have not loaded any controlled terminology yet, you can now do so using the menu "Add - CDISC Controlled Terminology":



This menu can also be used to replace the controlled terminology by a newer version, or to add one or more (e.g. new) codelists.

All CDISC controlled terminology is then presented as a list:

¹⁶ A download page for new published CDISC Controlled Terminology is made available by XML4Pharma



We choose e.g. the SDTM controlled terminology version 2022-06-24.

If controlled terminology (e.g. an older version) was already loaded before, the system will ask:

		×
?	Add all CodeLists Add selected CodeLists	
	ОК	

And when "Add selected CodeLists" is selected, it will ask to select one or more, e.g.:

ect one or more CodeLists	×
CLC141657.TENNW1TC - 10-Meter Walk/Run Functional Test Test Code CLC141653.A4STR1TC - 4-Stair Ascend Functional Test Test Name CLC141663.A4STR1TC - 4-Stair Ascend Functional Test Test Name CLC141661.D4STR1TK - 4-Stair Ascend Functional Test Test Name CLC141660.D4STR1TK - 4-Stair Descend Functional Test Test Name CLC115388.SIXMW1TC - 6 Minute Walk Functional Test Test Name CLC115388.SIXMW1TC - 6 Minute Walk Functional Test Test Name CLC182464.AIMS0101707GR - Abnormal Involuntary Movement Scale Clinical Classification ORRES for AIMS0101 Through AIMS0107 TN/TC CLC182465.AIMS01081090GR - Abnormal Involuntary Movement Scale Clinical Classification ORRES for AIMS0108 Through AIMS0109 TN/TC CLC182467.AIMS0101107GR - Abnormal Involuntary Movement Scale Clinical Classification ORRES for AIMS0110 TN/TC CLC182467.AIMS010107GSTR - Abnormal Involuntary Movement Scale Clinical Classification STRES for AIMS0101 Through AIMS0107 TN/TC CLC182467.AIMS01011705TR - Abnormal Involuntary Movement Scale Clinical Classification STRES for AIMS0101 TN/TC CLC182502.AIMS01010705TR - Abnormal Involuntary Movement Scale Clinical Classification STRES C for AIMS0107 TN/TC CLC182505.AIMS01081095TR - Abnormal Involuntary Movement Scale Clinical Classification STRES C for AIMS0107 TN/TC CLC182505.AIMS01105TR - Abnormal Involuntary Movement Scale Clinical Classification STRES C for AIMS0107 TN/TC CLC182505.AIMS01105TR - Abnormal Involuntary Movement Scale Clinical Classification STRES C for AIMS0101 TN/TC CLC101805.AIMS01105TR - Abnormal Involuntary Movement Scale Clinical Classification STRES C for AIMS0110 TN/TC CLC101805.AIMS01105TR - Abnormal Involuntary Movement Scale Clinical Classification TSTES C for AIMS0111 Through AIMS0112 TN/TC CLC101805.AIMS011T - Abnormal Involuntary Movement Scale Clinical Classification STRES C for AIMS0111	
OK	

Once again, one can use the "Search" button for finding a specific codelist.

When one has then selected "all codelists" or "selected codelists", and one already has codelists loaded, e.g. an older version, the system will ask whether these earlier codelists may be overwritten when there is one with the same OID in the selected set:



If one clicks "No", there is of course the risk of duplicates ... So one will usually use "Yes", leading to:

Message		\times
i	693 duplicates were removed	
	ОК	

This may sometimes lead to surprises, as some codelists have been deleted or deprecated by CDISC in the course of time, but in the case of Define-XML, this will be clearly visible by the value in the "StandardOID" column. For example:

		OID	Name	DataType	SASFo	StandardOID	IsNonStand
E	٩,	CL.C100134.BPR01TC	Brief Psychiatric Rating Scale-A Questionnaire Test Code	text		STD.SDTM.CDISC-NCI_2018-03-30	
D	9	CL.C100133.BPR01TN	Brief Psychiatric Rating Scale-A Questionnaire Test Name	text		STD.SDTM.CDISC-NCI_2018-03-30	
D	9	CL.C135684.CGGUY1TC	Clinical Global Impression Questionnaire Test Code	text		STD.SDTM.CDISC-NCI_2018-03-30	
D	9	CL.C135683.CGGUY1TN	Clinical Global Impression Questionnaire Test Name	text		STD.SDTM.CDISC-NCI_2018-03-30	
E	0	CL.C66787.TDIGRP	Diagnosis Group Response	text		STD.SDTM.CDISC-NCI_2018-03-30	
E	9	CL.C111346.FAQ02TC	Functional Activities Questionnaire-NACC Version Questionnaire Test Code	text		STD.SDTM.CDISC-NCI_2018-03-30	
D	9	CL.C111345.FAQ02TN	Functional Activities Questionnaire-NACC Version Questionnaire Test Name	text		STD.SDTM.CDISC-NCI_2018-03-30	
E	0	CL.C103329.GNRLOBSC	General Observation Class	text		STD.SDTM.CDISC-NCI_2018-03-30	
E	9	CL.C132317.GENRTYP	Genetic Region of Interest Type	text		STD.SDTM.CDISC-NCI_2018-03-30	
E	0	CL.C100170.KPSSTC	Karnofsky Performance Status Scale Questionnaire Test Code	text		STD.SDTM.CDISC-NCI_2018-03-30	
E	9	CL.C100169.KPSSTN	Karnofsky Performance Status Scale Questionnaire Test Name	text		STD.SDTM.CDISC-NCI_2018-03-30	
E	9	CL.C114119.PBSTMT	Pharmacogenomics Biomarker Medical Statement	text		STD.SDTM.CDISC-NCI_2018-03-30	
D	9	CL.C116106.PFTESTCD	Pharmacogenomics Findings Test Code	text		STD.SDTM.CDISC-NCI_2018-03-30	
E	9	CL.C116105.PFTEST	Pharmacogenomics Findings Test Name	text		STD.SDTM CDISC NCI_2018-03-30	
E	9	CL.C135010.PGTSTCD	Pharmacogenomics/Genetics Methods and Supporting Information Test Code	text		STD.SDTM.CDISC-NCI_2018-03-30	
D	9	CL.C135011.PGTST	Pharmacogenomics/Genetics Methods and Supporting Information Test Name	text		STD.SDTM.CDISC-NCI_2018-03-30	
E	9	CL.C116111.SPCIES	SDTM Species	text	1	STD.SDTM.CDISC-NCI_2018-03-30	
E	9	CL.C141657.TENMW1TC	10-Meter Walk/Run Functional Test Test Code	text		STD.SDTM.CDISC-NCI_2022-06-24	
D	٩,	CL.C141656.TENMW1TN	10-Meter Walk/Run Functional Test Test Name	text		STD.SDTM.CDISC-NCI_2022-06-24	
D	9	CL.C141663.A4STR1TC	4-Stair Ascend Functional Test Test Code	text		STD.SDTM.CDISC-NCI_2022-06-24	
E	9	CL.C141662.A4STR1TN	4-Stair Ascend Functional Test Test Name	text		STD.SDTM.CDISC-NCI_2022-06-24	
E	9	CL.C141661.D4STR1TC	4-Stair Descend Functional Test Test Code	text		STD.SDTM.CDISC-NCI_2022-06-24	
D	9	CL.C141660.D4STR1TN	4-Stair Descend Functional Test Test Name	text		STD.SDTM.CDISC-NCI_2022-06-24	
E	9	CL.C115388.SIXMW1TC	6 Minute Walk Functional Test Test Code	text		STD.SDTM.CDISC-NCI_2022-06-24	
D	٩,	CL.C115387.SIXMW1TN	6 Minute Walk Functional Test Test Name	text		STD.SDTM.CDISC-NCI_2022-06-24	
D	9	CL.C182464.AIMS0101T07OR	Abnormal Involuntary Movement Scale Clinical Classification ORRES for AIMS	text		STD.SDTM.CDISC-NCI_2022-06-24	
E	9	CL.C182465.AIMS0108T09OR	Abnormal Involuntary Movement Scale Clinical Classification ORRES for AIMS	text		STD.SDTM.CDISC-NCI_2022-06-24	
C	9	CL.C182466.AIMS0110OR	Abnormal Involuntary Movement Scale Clinical Classification ORRES for AIMS	text		STD.SDTM.CDISC-NCI_2022-06-24	
C	9	CL.C182467.AIMS0111T12OR	Abnormal Involuntary Movement Scale Clinical Classification ORRES for AIMS	text		STD.SDTM.CDISC-NCI_2022-06-24	
R	9	CL.C182502.AIMS0101T07STR	Abnormal Involuntary Movement Scale Clinical Classification STRESC for AIM	text		STD.SDTM.CDISC-NCI 2022-06-24	

where one sees that for 17 codelists from the 2018-03-30 version, there is no newer 2022-06-24 version.

One can now of course already remove codelists that are not needed, but one can also do this later in an automated way using the menu "Edit - Clean" (see later).

Once loaded, each of the loaded codelists can be **subset**ted by using the menu "Extra - Generate CodeList Subset":

View	Extra Options Help
	Insert CRF Page Numbers from annotated CRF
	Generate ValueList from CodeList
	Generate CodeList Subset
ns	Generate CodeList Subset starting from SAS-XPT
ital Do	Generate CodeListItem-CodeList from EnumeratedItem-Codelist

A new dialog is displayed, allowing you to select for which codelist you want to generate a subset:

odeList Subset generation
Existing CodeList to subset from:
CL.MEDDRA - MedDRA Adverse Events Dictionary
CL.C115388.SIXMW1TC - 6 Minute Walk Functional Test Test Code
CL.C115387.SIXMW1TN - 6 Minute Walk Functional Test Test Name
CL.C101805.AIMS01TC - Abnormal Involuntary Movement Scale Questionn
CL.C101806.AIMS01TN - Abnormal Involuntary Movement Scale Questionn
CL.C66767.ACN - Action Taken with Study Treatment
CL.C101865.ACSPCAT - Acute Coronary Syndrome Presentation Category
CL.C120985.APCH1TC - Acute Physiology and Chronic Health Evaluation II
CL.C120984.APCH1TN - Acute Physiology and Chronic Health Evaluation II
CL.C101807.AVL01TC - ADNI Auditory Verbal Learning Functional Test Tes
CL.C101808.AVL01TN - ADNI Auditory Verbal Learning Functional Test Tes
CL.C66781.AGEU - Age Unit
Search
Search Find Next Find Previous
OK Cancel

Searching for the right codelist is made ease by the "Search" field and buttons. One can search on the codelist OID (CDISC identifier) or the codelist name. So for "vital signs test code" one easily finds:

	CL.C102590.VCNEVD - Vaccination Evidence Source						
	CL.C66741.VSTESTCD - Vital Signs Test Code						
	CL.C67153.VSTEST - Vital Signs Test Name						
	CL.C117746.HEPENC	CGR - Wes	t Haven Hepatic	Encephalopathy Grade	-		
	•						
	Search						
	vstestcd						
	Search Find Next Find Previous						
OK Cancel							

Clicking "OK" leads to a new dialog:

Generate	CodeList Subset	\times					
2	Select the items you want to appear in the subset						
•	ABSKNF - Abdominal Skinfold Thickness						
	BMI - Body Mass Index						
	BMR - Basal Metabolic Rate	=					
	BODLNGTH - Body Length						
	BODYFATM - Body Fat Measurement						
	BSA - Body Surface Area						
	DIABP - Diastolic Blood Pressure						
	EWEIGHT - Estimated Weight						
	FARMCIR - Forearm Circumference						
	FRMSIZE - Body Frame Size						
	HDCIRC - Head Circumference						
	HEIGHT - Height						
	Also automatically subset the corresponding 'decode' CodeList with OID 'CL.C67153.VSTEST' and and Name 'Vital Signs Test Name 'and merge it as 'decode' in the subset						
	OK Cancel						

from which the codes to appear in the subset can be selected.

Unfortunately, the CDISC Controlled Terminology Team still publishes codelist with test codes without any information about what the code means, i.e. without "decodes". The latter can only be found by taking the NCI code of the code, and then do a lookup in another codelist with the decodes). The software will always try to do this, so for the "VSTESTCD" codelist, although it may not contain the "decodes" like "Abdominal Skinfold Thickness", the software will pick them up from the "VSTEST" codelist, and display them.

Also, the SDTM and SEND standards requires us to both deliver the test code (--TESTCD) and the test name (--TEST) as two variables (two columns in the dataset), whereas this is completely unnecessary when using modern technologies, as there is a 1:1 relation between test code and test

name (for more information, see here). So when creating subsets, we will both need to create a subset for --TESTCD as well as for --TEST.

For ease of use, the checkbox "Also automatically subset ...", takes the "decodes" (the "test names") and merges them as "decode" in the subset VSTESTCD codelist. This then also automatically will create the corresponding subset codelist for VSTEST¹⁷.

Suppose that we want to create a vital signs test code subset consisting of "DIABP" (diastolic blood pressure"), "SYSBP" (systolic blood pressure", "HEIGHT" (body height), "WEIGHT" (body weight), "BMI" (body mass index) and "HIPCIR" (hip circumference). We tick the boxes for those codes that we want to appear in the subset:

Generate	CodeList Subset	\times					
?	Select the items you want to appear in the subset						
_	☑ BMI - Body Mass Index	F					
	BMR - Basal Metabolic Rate						
	BODLNGTH - Body Length	=					
	BODYFATM - Body Fat Measurement						
	BSA - Body Surface Area	Ц.					
	DIABP - Diastolic Blood Pressure						
	EWEIGHT - Estimated Weight						
	FARMCIR - Forearm Circumference						
	FRMSIZE - Body Frame Size						
	HDCIRC - Head Circumference						
	🕑 HEIGHT - Height						
	HIPCIR - Hip Circumference	-					
	Also automatically subset the corresponding 'decode' CodeL with OID 'CL.C67153.VSTEST' and and Name 'Vital Signs Test Name 'and merge it as 'decode' in the subset	ist					
	OK Cancel						

and then click the OK button. The software now proposes a new OID (identifier) and name for the subset codelist:

¹⁷ This also works for CDISC codelists that end with "Test Code" in the name, like the codelist with OID "CL.C141657.TENMW1TC" with Name "10-Meter Walk/Run Functional Test Test Code". If this codelist is taken, and the checkbox "Also automatically subset the corresponding 'decode' CodeList ..." is checked, also the corresponding subset codelist "10-Meter Walk/Run Functional Test Test Name" (subset codelist of codelist with OID "CL.C141656.TENMW1TN") will be generated.

Provide r	new OID and Name			
?	Please provide a new CodeList OID			
CL.C66741.VSTESTCD.SUBSET				
Please provide a new CodeList Name				
	Vital Signs Test Code subset			
	OK Cancel			

We can of course change these, for example:

Provide n	ew OID and Name
2	Please provide a new CodeList OID
•	CL.C66741.VSTESTCD.MyStudy
	Please provide a new CodeList Name
	Vital Signs Test Code for my study
	OK Cancel

If also the checkbox "Also automatically subset the corresponding 'decode' CodeList" is checked, also a subset codelist "CL.C67153.VSTEST.SUBSET" will be created.

After clicking "OK", a message is shown:

Message	×
i	A CodeList element with OID: CL.C66741.VSTESTCD.MyStudy has been created. In addition, also a corresponding subset 'decode' CodeList element with OID: CL.C67153.VSTEST.SUBSET has been created. You may need to check, edit and add additional information to the created CodeList element(s) now
	OK

and the new codelist(s) appears in the list of all codelists:

	wonu mealur organization bisabil	техт
CL.C130273.WD7TC	World Health Organization Disabil	text
CL.C130272.WD7TN	World Health Organization Disabil	text
CL.C66741.VSTESTCD.MyStudy	Vital Signs Test Code for my study	text
CL.C67153.VSTEST.SUBSET	Vital Signs Test Name subset	text

One can then of course still change the subset codelist, change the value for OID or Name, add new terms. For example, if one wants to extend this subset vital signs codelist with the term "Upper Tight Circumference", with a code "UPTICIR", just click the "add information" icon ("+"), leading to:

nformation for: C	odeList, with OID	= CL.C66741.VSTESTCI	D.MyStudy		
Description	CodeListItem	ExternalCodeList	EnumeratedItem	Alias	
CodedVa	lue	Rank	OrderNumber	ExtendedValu	e
S S BMI					
HEIGHT					
SYSBP					
WEIGHT					
E.Q.					

and add "UPTICIR" in a new row. As this is an extension to the CDISC codelist, you should also set the value of "ExtendedValue" to "Yes":

⊱ Extra ir	ra information for: CodeList, with OID = CL.C66741.VSTESTCD.MyStudy					
?	Description CodeListItem	ExternalCodeList	EnumeratedItem Alias]		
_	CodedValue	Rank	OrderNumber	ExtendedValue		
	🖸 🔍 BMI					
	SYSBP					
	R WEIGHT					
				Yes		
				Yes		

Do however NOT assign an NCI code to such a new entry that is not a CDISC term.

In case the codelist is an "CodeListItem" codelist, one should also add the "decode" by clicking the "+" icon and adding the information:

See Extra information for: CodeListItem - Coded Value = UPTICIR					
?	Decode Alias Description				
	Language	Translated Text			
		Upper Tight Circumference			

After clicking "OK", the subset codelist is extended and updated.

You might also now do similar for the subset codelist for VSTEST, adding "Upper Tight Circumference" as an extension.

Let us now have a quick look to the generated XML in the define.xml. This is never a bad idea if one wants to understand what one has done.

In order to do so, use the menu "View - Define.xml as XML":



and select the last "CodeList" in the list of tree nodes:



The XML is then shown on the right side:

```
<CodeList xmlns="http://www.cdisc.org/ns/odm/v1.3"
         xmlns:def="http://www.cdisc.org/ns/def/v2.1"
         xmlns:xs="http://www.w3.org/2001/XMLSchema"
         xmlns:xlink="http://www.w3.org/1999/xlink"
         DataTvpe="text"
         Name="Vital Signs Test Code subset"
         0ID="CL.C66741.VSTESTCD.MyStudy">
  <CodeListItem CodedValue="BMI">
     <Decode>
                       <TranslatedText>Body Mass Index</TranslatedText>
                   </Decode>
     <Alias Context="nci:ExtCodeID" Name="C16358"/>
  </CodeListItem>
   <CodeListItem CodedValue="DIABP">
     <Decode>
                        <TranslatedText>Diastolic Blood Pressure</TranslatedText>
                   </Decode>
     <Alias Context="nci:ExtCodeID" Name="C25299"/>
  </CodeListItem>
   <CodeListItem CodedValue="HEIGHT">
     <Decode>
                       <TranslatedText>Height</TranslatedText>
                   </Decode>
      <Alias Context="nci:ExtCodeID" Name="C25347"/>
  </CodeListItem>
   <CodeListItem CodedValue="HIPCIR">
     <Decode>
                       <TranslatedText>Hip Circumference</TranslatedText>
                   </Decode>
      <Alias Context="nci:ExtCodeID" Name="C100947"/>
   </CodeListItem>
   <CodeListItem CodedValue="SYSBP">
      <Decode>
                       <TranslatedText>Systolic Blood Pressure</TranslatedText>
                    </Decode>
      <Alias Context="nci:ExtCodeID" Name="C25298"/>
   </CodeListItem>
   <CodeListItem CodedValue="WEIGHT">
      <Decode>
                        <TranslatedText>Weight</TranslatedText>
                    </Decode>
      <Alias Context="nci:ExtCodeID" Name="C25208"/>
  </CodeListItem>
   todeListItem CodedValue="UPTICIR" def:ExtendedValue="Yes">
      <Decode>
        <TranslatedText>Upper Tight Circumference</TranslatedText>
      </Decode>
   CodeListItem>
   <Alias Context="nci:ExtCodeID" Name="C66741"/>
</CodeList>
```

Where we see that there is an NCI code (using the "Alias" element) for each "standard" vital signs test code, whereas the extension code "UPTICIR" does not have an NCI code, but has the def:ExtendedValue="Yes" attribute. We also see that the codelist as a whole has an NCI code (C66741) from the last "Alias" element.

This is also a requirement from the CDISC standards, i.e. that when one subsets a codelist from an existing CDISC codelist, one <u>must</u> keep the NCI code of the codelist itself.

Or when inspected in the browser using the stylesheet (menu "View - define.xml in browser):

View	Extra	Options	Help		
Deep ODM Tree					
define.xml as XML					
define.xml in browser (as HTML)					

which will result that the "view" on the define.xml will be opened in your standard (favorite) brower:

Vital Signs Test Code subset [C66741]

Permitted Value (Code)	Display Value (Decode)	
BMI [C16358]	Body Mass Index	
DIABP [C25299]	Diastolic Blood Pressure	
HEIGHT [C25347]	Height	
HIPCIR [<i>C100947</i>]	Hip Circumference	
SYSBP [C25298]	Systolic Blood Pressure	
WEIGHT [C25208]	Weight	
UPTICIR [*]	Upper Tight Circumference	

* Extended Value

The only thing we still need to do is to assign our new subset codelist to the VSTESTCD variable. This can easily be done by selecting the "Variable Definitions" tab, selecting the VSTESTCD variable:

	Dataset Definition	s Variable Defir	nitions Codelis	sts Imputa	ation Methods	Presentatio	ons Co	ondition Definition	ns Me	thod Definitions	Comment Definition	s Document	links
	Annotated CRF	s Supple	mental Documen	ts	ValueList Defin	tions	Whe	eClause Definiti	ons	Includes	Protocol/Trial Desi	gn St	tudy Ev
	Search for: VSTESTCD Search Find Next Match case Whole words only												
I			Search	within:	All Columns								
I			V OID	l.	✓ Name	DataType	📃 Leng	ıth 📃 Si	gnificantD)igits 🗹 SASFiel	dName		
I			SDS	VarName	Origin	Comment	🗌 Disp	layFormat 📃 Co	ommentO	D			
I													
I	OID	Name	DataType	Length	Significant	igits SASFie	ldName	SDSVarName	Origin	Commen	DisplayFormat	CommentOID	1
I	SVS.VSSPID	VSSPID	text	80									▲
I	C VS.VSTESTC	VSTESTCD	text	8									
I	SVS.VSTEST	VSTEST	text	30									
I	SVS.VSCAT	VSCAT	text	80									
I	SVS.VSSCAT	VSSCAT	text	80									
I	SVS.VSPOS	VSPOS	text	23									
I	SVS.VSORRES	VSORRES	text	80									
I	SVS.VSORRES	U VSORRESU	text	11									
I	SVS.VSSTRES	C VSSTRESC	text	80									
I	SVS.VSSTRES	VSSTRESN	integer	8									
			text	11									

and then click the "Add Information" icon ("+") to add additional information. When the selecting the "CodeList Reference" tab:

Extra information for: ItemDef, with OID = IT.VS.VSTESTCD

Description	RangeCheck	CodeList Reference	Role	Alias	Origin	ValueList Reference	
CodeListOID					Codelist definition	s:	
					CL.C1246	76.WHIVS2TC	
					CL.C1246	75.WHIVS2TN	
					CL.C1001	176.WPAI01TC	
					CL.C1001	75.WPAI01TN	
					CL.C1302	281.WD4TC	
					CL.C1302	280.WD4TN	
					CL.C130275.WD1TC		
					CL.C130274.WD1TN		
					CL.C130277.WD2TC CL.C130276.WD2TN		
					CL.C1302	279.WD3TC	
					CL.C1302	278.WD3TN	
					CL.C1302	269.WD5TC	
					CL.C1302	268.WD5TN	
					CL.C1302	271.WD6TC	
					CL.C1302	270.WD61N	
					CL.C1302	2/3.WD/1C	
					CL.C1302	2/2.WD/IN	
					CL.C65/4		
					CL.C0/15	03.VSTEST.SUBSET	
						2	
						Search	

Remark that one can easily find a specific codelist by using the "Search" button. Now just drag-and-drop "CL.C66741.VSTESTCD.MyStudy" to the empty cell on the right:

Extra information for: ItemDef, with OID = IT.VS.VSTESTCD

?	Description RangeCheck CodeList Reference Role Alias	Origin
_	CodeListOID CL.C66741.VSTESTCD.MyStudy	Codelis definitio
		CL.C12
		CL.C12
		CL.C10
		CL.C10
		CL.C13
		CI C13

After clicking "OK", the subset codelist "CL.C66741.VSTESTCD.MyStudy" is assigned to VSTESTCD.

We can inspect this using the "browser view" using the menu "View - define.xml in browser",

View	Extra	Options	Help		
Deep ODM Tree					
define.xml as XML					
define.xml in browser (as HTML)					

p. 57

With a "stylesheet selection" dialog showing up:

Stylesheet	selection	×
?	 Use CDISC stylesheet Use own stylesheet 	
		Browse
	OK Cancel	

You can choose between the "standard" CDISC stylesheet (the one that comes with the CDISC Define-XML standard package), and your own or any other one. In the latter case, the "Browse" button will become available, so that you can select your own preferred stylesheet, which will then be applied.

When the "CDISC stylesheet" is selected, this is leading to:

and when clicking on "Vital Signs Test Code for my study":

ngs (MI) 🔨	Vital Signs (VS) [Location:]							
ceptibilit	Variable	Label	Key	Туре	Length	Controlled Terms or Format		
Concen	STUDYID	Study Identifier		text	40			
tion (PE)	DOMAIN	Domain Abbreviation		text	2			
Parame	USUBJID	Unique Subject Identifier		text	60			
tem Finc	VSSEQ	Sequence Number		integer	8			
e (RS)	VSGRPID	Group ID		text	80			
ristics (S	VSSPID	Sponsor-Defined Identifier		text	80			
SS)	VSTESTCD	Vital Signs Test Short Name		text	8	Vital Signs Test Code for my study		

Microscopic Findi

- Morphology (MO)
- Microbiology Sus
- Pharmacokinetics
- Physical Examina
- Pharmacokinetics
- Questionnaires (0
- Reproductive Sys
- Disease Response
- Subject Characte
- Skin Response (S
- Subject Status (S

Vital Signs Test Code for my study [CL.C66741.VSTESTCD.MyStudy, C6674	1]
Permitted Value (Code)	

BMI [C16358]
DIABP [C25299]
HEIGHT [C25347]
HIPCIR [<i>C100947</i>]
SYSBP [<i>C25298</i>]
WEIGHT [C25208]
UPTICIR [*]

Extended Value

with for each "Standard" code, the NCI code displayed and for the "UPTICIR" code, it being marked as a "extended" value.

In case we checked the checkbox "Also automatically subset the corresponding 'decode' CodeList ..." also the corresponding subset codelist with VSTEST terms has been created, and the only thing we then still need to do is to assign it to the VSTEST variable.

If we did not check the checkbox, no subset codelist with VSTEST terms has been created, but we can then do this separately, again using the menu "Extra - Generate CodeList Subset".

Adding and subsetting codelists from SAS-XPT files

When generating a "prototype" from a set of SAS-XPT files, one has the option to have the system generate subset codelists from the information in the XPT files (see section "Creating a define.xml starting from an existing set of SAS-XPT files"). One should however not need to wait until all XPT files to start generating a define.xml file!

Also, when generating the define.xml starting from a template for a specific SDTM, SEND or ADaM version (see section "Creating a define.xml starting from an SDTM/SEND/ADaM template"), which is the better practice, it will not always be possible to generate all the subset codelists until the XPT files are final. Of course, one can often already use the information from the CRFs to generate such subset codelists, but also this is not always possible.

In case one wants to create a subset codelists from the contents of one or more XPT files in a later stage, one can always do so using the menu "Extra - Generate CodeList Subset starting from SAS-XPT":



The following dialog is presented:

CodeList	mapping		×					
2	Existing CodeList to mat	tch:						
	CL.C115388.SIXMW1TC - 6 Minute Walk Functional Test Test Code							
	CL.C115387.SIXMW	1TN - 6 Minute Walk Funct	ional Test Test Name					
	CL.C101805.AIMS01	TC - Abnormal Involuntary	Movement Scale Questionn					
	CL.C101806.AIMS01	TN - Abnormal Involuntary	Movement Scale Questionn					
	CL.C66767.ACN - Ac	tion Taken with Study Trea	atment					
	CL.C101865.AC SPC/	AT - Acute Coronary Syndr	ome Presentation Category					
	CL.C120985.APCH1	FC - Acute Physiology and	Chronic Health Evaluation II					
	CL.C120984.APCH1	FN - Acute Physiology and	Chronic Health Evaluation II					
	CL.C132311.ADTNQ	RS - Administration Techn	ique Response					
	CL.C101807.AVL01T	C - ADNI Auditory Verbal L	earning Functional Test Tes					
	CL.C101808.AVL01T	N - ADNI Auditory Verbal L	earning Functional Test Tes					
	CL.C66781.AGEU - A	ge Unit	_					
	Search		[*]					
	Search	Find Next	Find Previous					
	SAS-XPT file to load:							
Browse								
No SAS-XPT file selected yet								
			•					
		OK Cancel						

Now select a codelist you wish to subset. If one does not know the exact name, one can use the "Search" panel, for example for "vital signs test code":

CodeList mapping



In our example, we will generate a subset for the VSTESTCD codelist based on the contents of our vs.xpt dataset:

We select the item: CL.C66741.VSTESTCD - Vital Signs Test Code



Then select the SAS-XPT file you want to get the values from, using the "Browse" button. Obviously this is the vs.xpt file in this case.

CL.CT30241.VLEKSTTC - VIGNOS LOWER EXTREMITLY RAUNG SCALE C	linical classification rest code		
CL.C130240.VLERS1TN - Vignos Lower Extremity Rating Scale Clinical Classification Test Name			
CL.C66741.VSTESTCD - Vital Signs Test Code			
	Search		
SAS-XPT file to load:			
	Browse		
No SAS-XPT file selected yet			
	OK Cancel		

When an XPT file is selected, a dropdown to select the variable of interest will be displayed from which one can choose:

p. 61

Once the file selected, the system analyzes its metadata (this takes a few seconds), and makes a proposal for the variable for which the unique values from the SAS-XPT file will be retrieved:

Search		
vital signs test		
Search	Find Next	Find Previous
	Browse	
e vet	Division	
s.xpt		
STESTCD		

This choice can then still be changed using the dropbox.

When clicking "OK", the system takes all the VSTESTCD values from the SAS-XPT "vs.xpt" file and takes all the unique ones. It then tries to match them to the values from the codelist CL.C66741.VSTESTCD "Vital Signs Test Code". This leads to a new dialog:

Rows for which the checkbox is of When the coded value is present When the coded value is only pre- and all 'EnumeratedItem' element You will need to add the decoded as it cannot always be obtained u In the case of a CDISC codelist, for	checked will be added as an item to the r in the original CodeList, it will be copied sent in the SAS-XPT file, a 'CodeListItem ts will be converted into 'CodeListItem/De value yourself, nambigously from the information in the or these items, the item will be marked as	new (subset) codelist. from there. /Decode' element will be added, ecode' elements. SAS-XPT file. s an 'extended value' in the define.xml.	
Rows for which the checkbox is I Add to Subset	NOT checked, will NOT be added to the n	ew CodeList Coded Term from CodeList	
	DIABP	DIABP - Diastolic Blood Pressure	
v	FRMSIZE	FRMSIZE - Body Frame Size	
×	HEIGHT	HEIGHT - Height	
×	PULSE	PULSE - Pulse Rate	
	SYSBP	SYSBP - Systolic Blood Pressure	
	WEIGHT	WEIGHT - Weight	
		ABSKNF - Abdominal Skinfold Thickness	
		ARMSPAN - Arm Span	
		BMI - Body Mass Index	
		BMIAPCTL - BMI-for-Age Percentile	
	i	DMD - Decel Metchelia Dete	

The first column contains a checkbox, allowing <u>the user</u> to decide whether the coded term should be included in the subset.

The second column contains the term as found in the SAS-XPT file and the third column the coded term (and the decode when present) in the codelist. So if the term is present in as well the second as third column, there is a match between the SAS-XPT value and the already available codelist. In such a case, it is recommended to add the coded term to the subset codelist (so the checkbox is already checked).

When scrolling down:

Rows for which the checkbox is NOT checked, will NOT be added to the new CodeList

Add to Subset	Coded Term from SAS-XPT	Coded Term from CodeList	
×	PULSE	PULSE - Pulse Rate	•
×	SYSBP	SYSBP - Systolic Blood Pressure	
×	WEIGHT	WEIGHT - Weight	
		ABSKNF - Abdominal Skinfold Thickness	
		ARMSPAN - Arm Span	
		BMI - Body Mass Index	
		BMIAPCTL - BMI-for-Age Percentile	=
		BMR - Basal Metabolic Rate	
		BODLNGTH - Body Length	
		BODYFATM - Body Fat Measurement	
		BRTHWT - Birth Weight	

If one would find a value in the second column without a counterpart in the third column, this would mean that the value is NOT in the codelist, BUT it is in the SAS-XPT file. If the checkbox is then checked to indicate that it should be added to the subset codelist, it will then appear there as an "extended value" with an attribute 'def:ExtendedValue="Yes".

Obviously, our dataset did not contain a value in the SAS-XPT file that was not in the codelist yet.

Suppose however that there was a field in the CRF for "BMI", but that there are no data for it (yet) in the SAS-XPT dataset. In such a case "BMI" <u>must</u> be added to the codelist, as it was on the CRF. So we do check the checkbox for "BMI":

Rows for which the checkbox is NOT checked, will NOT be added to the new CodeList					
Add to Subset	Coded Term from SAS-XPT	Coded Term from CodeList			
×	PULSE	PULSE - Pulse Rate	-		
×	SYSBP	SYSBP - Systolic Blood Pressure			
×	WEIGHT	WEIGHT - Weight			
		ABSKNF - Abdominal Skinfold Thickness			
		ARMSPAN - Arm Span			
✓		BMI - Body Mass Index			
		BMIAPCTL - BMI-for-Age Percentile	=		
		BMR - Basal Metabolic Rate			
		BODLNGTH - Body Length			
		BODYFATM - Body Fat Measurement			

Clicking "OK" will then create a subset of the codelist. It automatically assigns the OID "CL.C66741.VSTESTCD.SUBSET". You will of course later be able to change the OID. The original codelist name is however retained.

The system then asks whether it should assign the newly generated subsetted codelist to the variable "VSTESTCD":



I some cases, a list if variables can appear here, and you can choose to which ones you want to have the newly generated codelist assigned.

This results in:

Card	<u>ما مساحا لا م م</u> ز	- De euroitetie e Oeuroite	44	
Care	Message		x	ľ.
Card Card Card	i	A CodeList element with OID: CL.C66741. You must check, edit and add additional i	VSTESTCD.SUBSET has been created. information to the created CodeList element now	
Card Card Cate		0	К	
ate	gory for mo	auston/Exclusion	Iexi	
				_

Next the Territoria of the Orderic Brown Advance Transfer MACO

and when clicking "OK" again, the table with all the codelists will be displayed, containing a new row for the subset codelist:

CL.C66741.VSTESTCD	Vital Signs Tes	t Code	text
CL.C67153.VSTEST	Vital Signs Test Name te		text
CL.C117746.HEPENCGR	West Haven Hepatic Encephalopathy Grade		text
CL.C66741.VSTESTCD.SUBSET	Vital Signs Test Code t		text
Add Row		Delete Sel	ected Row
Move Selected Row Up		Move Selecte	d Row Down

One can now still edit this codelist, add terms, remove terms, change the name, etc..

Clicking the "magnifying glass" icon gives an overview:

🔹 Conten	Image: Contents of CodeList with OID CL.C66741.VSTESTCD.SUBSET and with Name Vital Signs Test Code Attributes:							
	Name		Value	e				=
	OID	CI		TCD.SUBSET				
	Name	Vi	tal Signs Test Co	de				
	DataType	tex	st					
	SASFormatNa	me						
	Content for De No information Content for Co	scripti deList	on Item					
	CodedValue	Rank	OrderNumber	ExtendedValue	Decode	А	lias	
	DIABP				TranslatedText Language: English Text: Diastolic Blood Pressure	Attr.Name Context Attr.Name Name	Attr.Value nci:ExtCodeID Attr.Value C25299	

also containing "BMI":

	WEIGHT	I I		I ranslated I ext	_ 		
				Language: English	Attr.Name	Attr.Value	
				Text: Weight	Name	C25208	
					Attr.Name	Attr.Value	
				TranslatedText	Context	nci:ExtCodeID	
	ВМІ			Language: English	Attr.Name	Attr.Value	
				Text: Body Mass Index	Name	C16358	
Na Ca Na Wa	o information ontent for Enumera o information arning: Information is	ntedItem required !					
Co	ontent for Alias						
ſ	Context Nam	ne 741					
	1		_				•
				OK Cancel			

Remark that also the "Alias" child elements, containing the NCI code, were copied from the original codelist.

Suppose now that we also subsetted the "LBTESTCD" codelist, and that the SAS-XPT file contains an LBTESTCD value "ALBB". It might be that this is a mapping error: in that case one would need to re-generate the lb.xpt file.

If, however, it was not an error, but "ALBB" corresponds to "Aluminumbibutyrate Measurement", one would find the following list of coded items in the codelist:

5- E	tra information for: CodeList, with OID = CL.CO	55047.LBTESTCD			_ X
?	Description CodeListItem Exte	rnalCodeList EnumeratedItem	Alias		
	CodedValue	Rank	OrderNumber	ExtendedValue	
					▲ 3
				Yes	
	ALP				
	ALT C				
	ANISO				

Remark that it is automatically marked as "extended".

One would still need to add the "decoded value". This can be done using the "+" icon, and adding the text to the "Decode" element:

Se Extra in	nformation for: CodeListItem - Coded Value = ALBB
?	Decode Alias
_	Language
	en
	TranslatedText text content

Clicking the magnifying glass icon in the main table then displays:

🛓 Conten	nts of element Code	eListItem
?	Contents of	CodeListItem
	Attributes:	
	Name	Value
	CodedValue	ALBB
	Rank	
	OrderNumber	
	ExtendedValue	Yes
	Content for Dec	code
	Tra	nslatedText
	Language: Engli Text: Aluminum	ish nbibutyrate Measurement
	Content for Alia	as
	No information	

Remark the "ExtendedValue" with value "Yes" in the table of attributes, and that there is no "Alias" element with "nci:ExtCodeID", as this term is not in the CDISC controlled terminology. You can of course always do a "new term request" to the CDISC "controlled terminology team", but this may take a few months until it is approved.

When it is, you can then add the NCI code, and remove the "extended" marker.

Important here is that **the user is always in control**, and not the system, as in some other "black box" tools starting from Excel worksheets.

Looks good isn't it?

Transforming an "EnumeratedItem" CodeList into a "CodeListItem" CodeList

Many of the CDISC controlled terminology is published as an "Enumerated" CodeList, i.e. all values are published without description and without any translations into other languages. For example:

```
<CodeList OID="CL.Cl02578.DSSOUT" Name="Disease Outcome" DataType="text">
   <EnumeratedItem CodedValue="BACTERIOLOGICAL CURE">
       <Alias Context="nci:ExtCodeID" Name="C102600"/>
    </EnumeratedItem>
    <EnumeratedItem CodedValue="BACTERIOLOGICAL FAILURE">
       <Alias Context="nci:ExtCodeID" Name="C102601"/>
    </EnumeratedItem>
    <EnumeratedItem CodedValue="CLINICAL CURE">
       <Alias Context="nci:ExtCodeID" Name="C102607"/>
   </EnumeratedItem>
    <EnumeratedItem CodedValue="CLINICAL FAILURE">
        <Alias Context="nci:ExtCodeID" Name="C102608"/>
    </EnumeratedItem>
    <EnumeratedItem CodedValue="RECURRENT DISEASE">
        <Alias Context="nci:ExtCodeID" Name="C38155"/>
   </EnumeratedItem>
    <Alias Context="nci:ExtCodeID" Name="C102578"/>
```

Usually, the "coded value" is written uppercase, which is not very useful for populating a CRF. Also, in the case of electronic submissions to regulatory authorities in non-English speaking countries (Japan, China), sponsors will usually submit the English term as well as the term in the local language.

So we will want to transform such an "Enumerated" CodeList into a classic CodeList with a "Decode" and "TranslatedText" elements, like in:

```
<CodeList OID="CL.Cl02578.DSSOUT" Name="Disease Outcome" DataType="text">

<CodeListItem CodedValue="BACTERIOLOGICAL CURE">

<Decode>

<TranslatedText xml:lang="en">Bacteriological Cure</TranslatedText>

<TranslatedText xml:lang="ja">細菌学的治癒</TranslatedText>

</Decode>

<Alias Context="nci:ExtCodeID" Name="Cl02600"/>

</CodeListItem>
```

Our system has a feature to do such a transformation automatically. The result will initially always have the value for Decode/TranslatedText being the same value as for "CodedValue", but the user will then be able to change this and add translations for additional languages.

In order to do so, use the menu "Extra - Generate CodeListItem-CodeList from EnumeratedItem-CodeList". This will then present the use with a list of all available codelists that have at least one "EnumeratedItem" child element:



Then select the codelist for which you want to create Decode/TranslatedText elements, allowing you to change the way this appears on a CRF and/or you want to add translations. For example:

CodeList transformation



Checking "Set English as default language" will make it clearer later which language is meant when adding translations for other languages.

When checking "Try to generate CRF-friendly decoded texts" the system will try to generate camelcase "decoded" texts, which e.g. can be used in CRFs¹⁸.

After clicking the "OK" button, the following message is displayed:



and the codelist is added at the bottom to the table with the codelists.

If you do not want the old codelist anymore, but only want to work with the new one, you can delete the old one and rename the new one.

The table is displayed automatically:

p. 69

¹⁸ One of the use cases of define.xml is to develop CRFs that are obeying the SDTM requirements of the sponsor.

CL.C66741.VSTESTCD Vital Signs Test			text
CL.C67153.VSTEST	CL.C67153.VSTEST Vital Signs Test Name		text
CL.C117746.HEPENCGR	CL.C117746.HEPENCGR West Haven Hepatic Encephalopathy Grade		text
CL.C102578.DSSOUT.NEW Disease Outco			text
Add Row		Delete Sele	cted Row
Move Selected Row Up		Move Selected	I Row Down
Suggest OIDs		Continue Contr	

When clicking the "+" icon, one sees that the child elements are "CodeListItem" elements now instead of "EnumeratedItem" items:

?	Description CodeListItem Exter	nalCodeList Enu
	CodedValue	Rank
	BACTERIOLOGICAL CURE	
	BACTERIOLOGICAL FAILURE	
	CLINICAL CURE	
	CLINICAL FAILURE	
	RECURRENT DISEASE	

Extra information for: CodeList, with OID = CL.C102578.DSSOUT.NEV

and one can start adding translations into other languages. For example, for "BACTERIOLOGICAL CURE", we find for the "decode":

Se Extra information for: CodeListItem - Coded Value = BACTERIOLOGICAL CURE

?	Decode Alias Description	
_	Language	Translated Text
	en	Bacteriological Cure

which is "camel case".

and e.g. add the Japanese translation, leading to:

⊱ Extra	Extra information for: CodeListItem - Coded Value = BACTERIOLOGICAL CURE					
?	Decode Alia	is				
_	Language		Translated Text			
	en		Bacteriological Cure			
	ja		細菌学的治癒			

The "CodedValue" (asto be used in a database) however remains "BACTERIOLOGICAL CURE".

Displaying your define.xml as HTML in a browser using a stylesheet

Many people (unfortunately) know define.xml only from what they see in the browser. What is seen in the browser is however only a particular view on the information in the define.xml provided by a stylesheet¹⁹. When submitting the define.xml to the FDA or the PMDA, it is the sponsors duty²⁰ to also provide a stylesheet so that the reviewers can see the information in a user-friendly way.

Most sponsors use the stylesheet that is provided by CDISC. Sponsors are however also free to develop their own stylesheet (or alter the one from CDISC). This however requires a good amount of XSLT knowledge.

Also from within our software, one can already obtain a view of the developed define.xml in the browser by applying the stylesheet. To do so, use the menu "View - define.xml in browser":

View	Extra	Options	Help		
Deep	ODM Tre	ee			
define.xml as XML					
define	e.xml in l	browser (a	as HTML)		

You will then be asked whether you want to use the default <u>CDISC stylesheet</u>, or your own or other favorite one:

Stylesheet	selection	×
?	 Use CDISC stylesheet Use own stylesheet 	
		Browse
	OK Cancel	

If you select "Use own stylesheet", you will need the "Browse" button to select a stylesheet file from your local file system.

Then click OK, and the define.xml will be displayed (as HTML of course) in your default²¹ (favorite) browser. For example:

¹⁹ Technically, stylesheets can even be used to manipulate the contents of the XML!

²⁰The FDA would better forbid that the stylesheet is provided by the sponsor, and use their own one. This would ensure that all studies are displayed in the same, sponsor-independent way.

²¹The system recognizes what your default browser is.

My own study

Expand all VLM Collapse all VLM

Standards Datasets Controlled Terminology

Date/Time of Define-XML document generation: 2022-09-12T11:47:20+01:00 Define-XML Context: Submission Stylesheet version: 2019-02-11

Study Name	My own study
Study Description	Description of my own study
Protocol Name	Protocol name of my own study
Metadata Name	MetaData for SDTMIG 3.4 for My own study
Metadata Description	MetaData for SDTMIG 3.4 for My own study Version 1.0

Standards for Study My own study				
Standard	Туре	Status	Documentation	
SDTMIG 3.4	IG	Final		
CDISC/NCI SDTM 2022-06-24	ст	Final		

		Datasets					
Dataset	Description	Class	Structure	Purpose	Keys	Documentation	Location
DM [SDTMIG 3.4]	Demographics	SPECIAL PURPOSE		Tabulation			<u>dm.xpt</u> &
SE [SDTMIG 3.4]	Subject Elements	SPECIAL PURPOSE		Tabulation			<u>se.xot</u> अ
SV [SDTMIG 3.4]	Subject Visits	SPECIAL PURPOSE		Tabulation			sv.xpt @
CM [SDTMIG 3.4]	Concomitant/Prior Medications	INTERVENTIONS		Tabulation			cm.xpt @
EX [SDTMIG 3.4]	Exposure	INTERVENTIONS		Tabulation			ex.xpt @
AE [SDTMIG 3.4]	Adverse Events	EVENTS		Tabulation			<u>ae.xpt</u> 생
DS [SDTMIG 3.4]	Disposition	EVENTS		Tabulation			ds.xpt @
MH [SDTMIG 3.4]	Medical History	EVENTS		Tabulation			<u>mh.xpt</u> ទ
EG [SDTMIG 3.4]	ECG Test Results	FINDINGS		Tabulation			<u>eq.xpt</u> क्ष
IE [SDTMIG 3.4]	Inclusion/Exclusion Criteria Not Met	FINDINGS		Tabulation			<u>ie.xpt</u> dP
LB [SDTMIG 3.4]	Laboratory Test Results	FINDINGS		Tabulation			<u>lb.xpt</u> අ
PE [SDTMIG 3.4]	Physical Examination	FINDINGS		Tabulation			<u>pe.xpt</u> ल
QSCG [SDTMIG 3.4]	Questionnaires	FINDINGS		Tabulation			<u>qscq.xpt</u> क्ष
OSCS [SDTMIG 3.4]	Questionnaires	FINDINGS		Tabulation			<u>qscs.xpt</u> ही
QSMM [SDTMIG 3.4]	Questionnaires	FINDINGS		Tabulation			<u>qsmm.xpt</u> क्ष

Creating and populating Sponsor-defined Codelists

You probably also want to generate some sponsor-defined codelists.

Sponsor-defined codelists in Define-XML are not different from CDISC codelists except that they do not have a CDISC-NCI code (as provided in an "Alias" element).

In order to generate a new codelist, either navigate to the "CodeLists" panel and click the button "Add Row", or use the menu "Add - CodeList Definition". This will add a new, empty row at the bottom of the CodeList panel:

		Add Row		
,				
	CL.C130272.WD7TN	World Health Organization Disabil	text	
	CL.C130273.WD7TC	World Health Organization Disabil	text	
	CL.C130270.WD6TN	World Health Organization Disabil	text	
	CL.C130271.WD6TC	World Health Organization Disabil	text	
	CL.C130268.WD5TN	World Health Organization Disabil	text	

Now assign an OID (identifier), and Name for the codelist, e.g.:

- OCLO 13027 1.WD010	wond riealur Organization Disability AJext			
CL.C130270.WD6TN	World Health Organization Disability A text			
CL.C130273.WD7TC	World Health Organization Disability A text			
CL.C130272.WD7TN	World Health Organization Disability A text			
	Sponsor-defined codelist for ARMCD			
Add Row				

and select the value for the data type (usually "text") from the dropdown:
CL.C130270.WD6TN	World Health Organization Disability A	text	
CL.C130273.WD7TC	World Health Organization Disability A	text	
CL.C130272.WD7TN	World Health Organization Disability A	text	
	Sponsor-defined codelist for ARMCD	text	
	Add Row		

You can now adding some terms to your codelist by clicking on the "+" icon. This opens a new dialog:

疑 Extra information for: CodeList, with OID = CL.ARMCD

?	Description CodeListItem ExternalCodeList EnumeratedItem	Alias
_	Language	Translated Text

On the tab "Description", you can now add a description of the codelists, even in different languages. You will now also need to decide whether you just want to add a list of the codes without explanation ("enumerated items") or a list of terms with explanations ("codelist items" with "decodes"). In most cases you will want to add explanations to the coded values. In that case, select the tab "CodeListItem". This leads to:

⊱ Extra	information for: CodeList, with OID) = CL.ARMCD		
?	Description CodeListItem	ExternalCodeList Enur	nerateditem Alias	
_	CodedValue	Rank	OrderNumber	ExtendedValue
	R.S.			
	E.S.			
	R.S.			
	E.S.			
	E.S.			
	I B O			

Remark that the selected tab is always displayed in yellow.

You can now add codes, in our case for "ARMCD" (trial arm code). Let us already add a few ones:

Description CodeListItem Exter	rnalCodeList EnumeratedItem	Alias	
CodedValue	Rank	OrderNumber	ExtendedValue
WONDER10			
WONDER20			
R.Q.			
RQ			

But what do "WONDER10" and "WONDER20" mean? Best practice is to provide an explanation by using the "decode" of Define-XML.

For doing so, click on the "+" icon, e.g. for the row with "WONDER10". This opens a new dialog:

See Extra information for: CodeListItem - Coded Value = WONDER10

?	Decode Alias Description	
_	Language	Translated Text

Now add the explanation in the field "TranslatedText", e.g.:

See Extra information for: CodeListItem - Coded Value = WONDER10

You can do so in different languages, but then also need to provide the language. For example:

Extra information for: CodeListItem - Coded Value = WONDER10

 Decode
 Alias
 Description

 Language
 Translated Text

 en
 10mg active ingredient

 ja
 有劾成分 10mg

Then do also for "PLACEBO" and "WONDER20".

You will usually not need to populate "Rank", unless you want to add an order for importance, like in "mild - moderate - severe". Also remind that "OrderNumber" is only about the order of display, which is something completely. If you just want to have the display order the same as how you added the items, there is no need to populate "OrderNumber". In the case of a sponsor-defined codelist, also leave "ExtendedValue" untouched, as you do not extend a CDISC codelist here.

When inspecting the complete codelist (by clicling the "magnifying glass" icon), this may then look like:

 \times

Х

 \times

p. 74

Name		Val	ue				
OID	CI	L.ARMCD					
Name	Sp	oonsor-defined co	delist for ARMCE				
DataType	tex	xt					
SASFormatNan	ne						
StandardOID							
IsNonStandard							
CommentOID							
Content for Cod	eList Rank	Item OrderNumber	ExtendedValue	Decode	Alias	Description	1
				TranslatedText			
				Language: English			
PLACEBO				Text: Placebo			
				TranslatedText			
				Language: Japanese Text: プラセボ			
				TranslatedText			
				TranslatedText Language: English			
WONDER10				TranslatedText Language: English Text: 10mg active ingredient			
WONDER10				TranslatedText Language: English Text: 10mg active ingredient TranslatedText			
WONDER10				TranslatedText Language: English Text: 10mg active ingredient TranslatedText Language: Japanese Text: 有劾成分 10mg			
WONDER10				TranslatedText Language: English Text: 10mg active ingredient TranslatedText Language: Japanese Text: 有劾成分 10mg TranslatedText			

In the case of Define-XML 2.1, you will still want to state that this is a "non-standard" (i.e. non-CDISC codelist), which you can do by using the dropdown for "IsNonStandard":

	OID	Name	DataType	SASFormatName	StandardOID	IsNonStandard	CommentOID
D) C	CL.ARMCD	Sponsor-defined codelist for ARMCD	text			Yes 🔻	
D, C	CL.C154449.UHDR1TC	Unified Huntington's Disease Rating Scale '9	text		STD.SDTM.CDISC-NCI_2022-06-24	Yes	
B	CL C154448 UHDR1TN	Unified Huntington's Disease Rating Scale '9	text		STD SDTM CDISC-NCI 2022-06-24		-

Remark that in Define-XML 2.1 "StandardOID" and "IsNonStandard" are "mutually exclusive", i.e. it is not allowed to have both set.

Suppose now that you have already created your SAS-XPT files, and want to have the sponsordefined codelist populated with the values from a SAS-XPT file.

In order to do so, create the sponsor-defined codelist, but do now add any terms to it. For example:

p. 75

CL.C130270.WD6TN	World Health Organization Disability A te	ext	
CL.C130273.WD7TC	World Health Organization Disability A te	ext	
CL.C130272.WD7TN	World Health Organization Disability A te	ext	
	Sponsor-defined codelist for ARMCD te	ext	
	Add Row		

One sees that the "+" icon is not colored, meaning that there is no underlying information. If you e.g. want to populate the codelist from the values for ARMCD in your SAS-XPT file "dm.xpt", use the menu "Extra - Generate CodeList Subset starting from SAS-XPT:

⊱ Define.xml Designer 2022 by XML4Pharma		
File Edit Add Transform Validate View	Extra Options Help	
	Insert CRF Page Numbers from annotated CRF	
	Generate CodeList Subset	
	Generate CodeList Subset starting from SAS-XPT	
Standards Annotated CRFs Supplement	Adapt Variable Length from SAS-XPT file contents	е
OID Name	Order Dataset definitions alphabetically per class	20
	Data 1990 On Or O	<i>n</i> 1

and in the dialog that is displayed, select your newly generated (but still empty) codelist:

CodeList mapping

2	Existing CodeList to match:
•	CL.C130278.WD3TN - World Health Organization Disability Assessment Schedule 2.0 - 12-Item Self Questionnaire Test Name
	CL.C130269.WD5TC - World Health Organization Disability Assessment Schedule 2.0 - 36-Item Interviewer Questionnaire Test Code
	CL.C130268.WD5TN - World Health Organization Disability Assessment Schedule 2.0 - 36-Item Interviewer Questionnaire Test Name
	CL.C130271.WD6TC - World Health Organization Disability Assessment Schedule 2.0 - 36-Item Proxy Questionnaire Test Code
	CL.C130270.WD6TN - World Health Organization Disability Assessment Schedule 2.0 - 36-Item Proxy Questionnaire Test Name
	CL.C130273.WD7TC - World Health Organization Disability Assessment Schedule 2.0 - 36-Item Self Questionnaire Test Code
	CL.C130272.WD7TN - World Health Organization Disability Assessment Schedule 2.0 - 36-Item Self Questionnaire Test Name
	CL.ARMCD - Sponsor-defined codelist for ARMCD
	Search
	SAS-XPT file to load:
	Browse
	No SAS-XPT file selected yet
	OK Cancel

Then select your dm.xpt file using the "Browse.." button. For example leading to:

CodeList	mapping
?	Existing CodeList to match:
-	CL.C130278.WD3TN - World Health Organization Disability Assessment Schedule 2.0 - 12-Item Self Questionnaire Test Name
	CL.C130269.WD5TC - World Health Organization Disability Assessment Schedule 2.0 - 36-Item Interviewer Questionnaire Test Code
	CL.C130268.WD5TN - World Health Organization Disability Assessment Schedule 2.0 - 36-Item Interviewer Questionnaire Test Name
	CL.C130271.WD6TC - World Health Organization Disability Assessment Schedule 2.0 - 36-Item Proxy Questionnaire Test Code
	CL.C130270.WD6TN - World Health Organization Disability Assessment Schedule 2.0 - 36-Item Proxy Questionnaire Test Name
	CL.C130273.WD7TC - World Health Organization Disability Assessment Schedule 2.0 - 36-Item Self Questionnaire Test Code
	CL.C130272.WD7TN - World Health Organization Disability Assessment Schedule 2.0 - 36-Item Self Questionnaire Test Name
	CL.ARMCD - Sponsor-defined codelist for ARMCD
	Search
	SAS-XPT file to load:
	Browse
	dm.xpt
	ARMCD
	OK Cancel

One sees that "ARMCD" is automatically selected as the SAS-XPT variable to retrieve from, but you can of course still change that by using the dropbox.

After clicking "OK", the SAS file is being analyzed, and a message is displayed:



It states that a new codelist will be generated with the OID "CL.ARMCD.NEW" and which will be populated with the distinct values from the ARMCD variable in the dm.xpt file.

We decided to make this a new codelist, as when one is not satisfied with the results of the retrieval. one can still use the original codelist, and add values there.

Of course you can later always change the value for the OID (identifier) of the codelist. It is just an identifier and has no meaning by itself.

After clicking "OK", one is presented with:

?	Rows for which the checkbox is che When the coded value is present in When the coded value is only prese and all 'EnumeratedItem' elements You will need to add the decoded va as it cannot always be obtained una In the case of a CDISC codelist, for t	ecked will be added as an item to the new (subs the original CodeList, it will be copied from ther ent in the SAS-XPT file, a 'CodeListItem/Decode' of will be converted into 'CodeListItem/Decode' ele alue yourself, ambigously from the information in the SAS-XPT these items, the item will be marked as an 'exte	eet) codelist. e. element will be added, ements. file. nded value' in the define.xml.
	Rows for which the checkbox is NO)T checked, will NOT be added to the new CodeL	ist
	Rows for which the checkbox is NO Add to Subset	OT checked, will NOT be added to the new CodeL Coded Term from SAS-XPT	ist Coded Term from CodeList
	Rows for which the checkbox is NO	OT checked, will NOT be added to the new CodeL Coded Term from SAS-XPT WONDER10	ist Coded Term from CodeList
	Rows for which the checkbox is NO Add to Subset	DT checked, will NOT be added to the new CodeL Coded Term from SAS-XPT WONDER10 WONDER20	ist Coded Term from CodeList
	Rows for which the checkbox is NO Add to Subset	DT checked, will NOT be added to the new CodeL Coded Term from SAS-XPT WONDER10 WONDER20 PLACEBO	ist Coded Term from CodeList

One can then still exclude some values by unchecking one or more checkboxes, e.g. due to regulatory requirements for the codelist (think about "OTHER", "MULTIPLE", "--ALL"). In this case, we accept everything by clicking "OK", leading to:

Message		×
i	Select the variables to which you would like to ass the codelist with OID CL.ARMCD.NEW.	ign
	IT.DM.ARMCD - ARMCD	
	IT.TA.ARMCD - ARMCD	
	IT.TV.ARMCD - ARMCD	•
	ОК	

The system has now detected that the "ARMCD" variable is also used in the TA and TV dataset definition, for which the variable identifiers are "IT.DM.ARMCD" (from the dm.xpt file), "IT.TA.ARMCD" (for the TA dataset definition) and "IT.TV.ARMCD" (for the TV dataset definition) and asks whether the system should assign this new codelist to these 3 variables. You will usually want to have that, and thus check all three checkboxes.

After clicking "OK", one gets some additional advice ...



This is good! For example, when having generated the items in the codelist by retrieval from the SAS-XPT file, there is no information about the meaning of the generated codes like "WONDER10" and "WONDER20" as they were obtained:

CodedValue	Rank	OrderNumber	ExtendedValue	Decode	Alias	Description
WONDER10				TranslatedText Language: not assigned Text: WONDER10		
WONDER20				TranslatedText Language: not assigned Text: WONDER20		
PLACEBO				TranslatedText Language: not assigned Text: PLACEBO		
SCRNFAIL				TranslatedText Language: not assigned Text: SCRNFAIL		

As the system only has retrieved "WONDER10", "WONDER20", "PLACEBO" and "SCRNFAIL", it has assigned the same values for the "decode". As we did in the prior case, we can then still change that by clicking on the "+" icon in the table with the codelists for "CL.ARMCD.NEW", and then "drill down" to the individual items.

Cleaning up your define.xml

After working some time on your define.xml, you will probably want to clean up some things. You might have some variables, valuelist definitions, comment and method definitions, and even "where clauses" that you once defined, but then finally decided not to use.

Technically spoken, this means that you have "definitions" (e.g. "def:CommentDef") that are never reference. This might even happen to variable definitions ("ItemDef"), for example "permissible" variables that you loaded from the SDTM template, but that you do not use in any of your datasets.

Having unreferenced definitions is in fact not a problem at all - they will even not show up in the HTML view that you get when displaying the define.xml using the stylesheet. However, the FDA does not like it.

In order to clean up your define.xml, use the menu "Edit - Clean". An inventory of all unreferenced "definitions" is then made and the result displayed:

Clean ODM			X					
i	Cleaning up the define.xml means that you can remove all definitions that are not used (i.e. not referenced by other define.xml elements). This ensures you that your define.xml is as compact as possible, and does not contain definitions that are not used anyway.							
In case your define.xml is not complete yet, cleaning up the define.xml is POTENTIALLY DANGEROUS, as you may remove elements definitions that you may want to use in future								
	It is always a good ide	ea to save your work before cle	aning up the define.xml					
	Select the definitions for which you want a cleanup							
	ValueListDef	Total: 20 - Unreferenced: 1	Show unreferenced					
	WhereClauseDef	Total: 123 - Unreferenced: 2	Show unreferenced					
	ltemDef	Total: 423 - Unreferenced: 0	Show unreferenced					
	CodeList	Total: 84 - Unreferenced: 0	Show unreferenced					
	MethodDef	Total: 57 - Unreferenced: 1	Show unreferenced					
	CommentDef	Total: 28 - Unreferenced: 1	Show unreferenced					
		OK Cancel						

informing us that there is 1 unreferenced ValueList (def:ValueList element), 2 "where clauses" (def:WhereClauseDef", and 1 unreferenced method definition ("MethodDef") and 1 unreferenced comment definition ("def:CommentDef").

By clicking a "Show unreferenced" button, one can immediately see which definitions are unreferenced. For example for "MethodDef":

Unreferen	ced MethodDef elements	×
i		Name Unreferenced Method

The user can now decide which type of unreferenced elements can be cleaned up, i.e. be removed from the define.xml. For example, if we do want to remove all unreferenced comments ("def:CommentDef") and "where clauses" ("def:WhereClauseDef"), but keep unreferenced "method definitions" (e.g. because you still want to use them later) and unreferenced "value list definitions", check the corresponding checkboxes:

Clean ODM	1		×						
()	Cleaning up the define.xml means that you can remove all definitions that are not used (i.e. not referenced by other define.xml elements). This ensures you that your define.xml is as compact as possible, and does not contain definitions that are not used anyway.								
	In case your define.xml is not complete yet, cleaning up the define.xml is POTENTIALLY DANGEROUS, as you may remove elements definitions that you may want to use in future.								
	It is always a good ide	ea to save your work before cle	aning up the define.xml						
	Select the definitions for which you want a cleanup								
	ValueListDef Total: 20 - Unreferenced: 1 Show unreference								
	✓ WhereClauseDef	Total: 123 - Unreferenced: 2	Show unreferenced						
	ItemDef	Total: 423 - Unreferenced: 0	Show unreferenced						
	CodeList	Total: 84 - Unreferenced: 0	Show unreferenced						
	MethodDef	Total: 57 - Unreferenced: 1	Show unreferenced						
	CommentDef	Total: 28 - Unreferenced: 1	Show unreferenced						
		OK Cancel							

and just click "OK" to start the cleanup action.

If you then use the menu "Edit - Clean" again, the result is:

Select the definitions for which you want a cleanup						
ValueListDef	Total: 20 - Unreferenced: 1	Show unreferenced				
WhereClauseDef	Total: 121 - Unreferenced: 0	Show unreferenced				
🗌 ItemDef	Total: 423 - Unreferenced: 0	Show unreferenced				
CodeList	Total: 84 - Unreferenced: 0	Show unreferenced				
MethodDef	Total: 57 - Unreferenced: 1	Show unreferenced				
CommentDef	Total: 27 - Unreferenced: 0	Show unreferenced				

The effect can of course also immediately be seen by inspecting the tables for the "WhereClause" and for the "Comment" or by inspecting the define.xml XML itself using "View - define.xml as XML".

Saving your work to file

Independent on whether you want to clean your define.xml or not, you will want to save your work from time to time.

By default, the software already saves the define.xml file every 5 minutes to a file in the folder "autosave", which is a subfolder to the folder where you installed the software.

You can change the "autosave" intervals by using the menu "Options - Settings" and look for the field "Number of minutes between define.xml autosave":

(only necessary in case of ODM-extensions to	TranslatedText)
 Use text window for TranslatedText content 	
Number of minutes between define.xml autosave	5
Use Schematron for local validation	

You will only be allowed to enter integer values. If you set "0" or a negative number, no autosaving will be done at all.

The files in the "autosave" directory have a name allowing you to find out when exact they were created. For example:

Name	Änderungsdatum	Тур
NDM_2019_10_19_16-52-35.xml	19.10.2019 16:52	XML-Date
nticologia (%) 000 001 001 001 001 001 001 001 001 00	19.10.2019 16:57	XML-Date
NDM_2019_10_19_17-2-35.xml	19.10.2019 17:02	XML-Date
NDM_2019_10_19_17-23-28.xml	19.10.2019 17:23	XML-Date

The file with name "ODM_2019_10_19_17-29-28.xml was created on October 19nd at 17:23:28 local time (24 hour notation is used).

So, even if the software or your computer would crash, you can always recover your earlier work.

Remark that it is not a bad idea to regularly clean up the contents of the "autosave" directory.

To actively save your define.xml to file, use the menu "File - Save define.xml". The following dialog is displayed:

Save define.xml to file

ODM File Description	
Study OID (required)	
MyStudy	
Metadata Version OID (required)	
MV.SDTM	
Metadata Version Name (required)	
MetaData for SDTMIG 3.4 for My own study	
Metadata Version Description	
MetaData for SDTMIG 3.4 for My own study Version 1.0	
dof:DofinoVorsion	
uer Dennie version	
2.1.0	
2.1.0 Define-XML Context	
2.1.0 Define-XML Context Submission	
2.1.0 Define-XML Context Submission Do not add a stylesheet reference (yet)	
2.1.0 Define-XML Context Submission One of add a stylesheet reference (yet) Add CDISC stylesheet reference	
2.1.0 Define-XML Context Submission Do not add a stylesheet reference (yet) Add CDISC stylesheet reference Add own stylesheet reference	

At this point, you can still change some of the metadata.

You can also select whether your output define.xml needs to have a reference to the stylesheet file. You can choose between not adding a reference at, adding a reference to the default stylesheet file delivered by CDISC, or to add a reference to a stylesheet that you developed yourself.

In the case of the latter choice, use the "Browse" button to browse for the name and location of that stylesheet file.

Remark that it remains your own responsibility to add (or copy) the stylesheet file to the directory where you save the file to. For the latest version of the CDISC define.xml stylesheet, please look in your define.xml CDISC distribution package or the CDISC website.

After clicking "OK", you will be asked where the file needs to be written to:

×

⊱ Save		×
Save In:	temp	• A A B
define.xml	🗋 test_study.xml	
DM.xml		
Svrl.xml		
temp.xml		
🗋 test.xml		
test_def.x	ml	
test_defin	e.xml	
File <u>N</u> ame:	test_define.xml	
Files of <u>Type</u> :	CDISC define.xml files	▼
		Save Cancel

Please also remark that is not always necessary to first save your work to file in order to display your define.xml in your favorite browser. As explained before, you can always use the menu "View - define.xml in browser".

Creating ValueLists

Creating ValueLists (value-level metadata) is often seen as one of the most challenging tasks when creating a define.xml. With our software however, it becomes very easy.

ValueLists are usually created when there are different "test codes", and the datatype, the maximal length, units and associated codelists for the results differ depending on the test codes. Typical examples are found in the "Findings" domains like VS (vital signs), LB (laboratory) and EC (electrocardiogram). For example, for vital signs, systolic and diastolic are integer number (with mmHg as unit), weight is a float, and "frame size" is text and governed by a codelist. For laboratory findings, the valuelists will usually be even more complicated.

Valuelists are mostly created based on --TESTCD and then assigned to -ORRES, and to --STRESC variables. They can also be assigned to --ORRESU and/or --STRESU variables. For supplemental qualifiers, a valuelist is very often assigned to QVAL, based on QNAM.

In this manual, we will show how to set up a valuelist for VS (vital signs) The best way to start generating a valuelist is to start from a codelist. Let us suppose that you have already loaded the CDISC-CT codelist for VSTESTCD:

Description CodeListItem	ExternalCodeList Enumerate	ditem Alias		
CodedValue	Rank	OrderNumber	ExtendedValue	
ABSKNF				
📴 🔍 BMI				
BODLNGTH				
BODYFATM				
🖪 🔍 BSA				
RMSIZE				
🖸 🔍 HEIGHT				
🖸 🔍 HR				
🖸 🔍 IDEALWT				
🖸 🔍 LBM				
🖸 🔍 MAP				
OXYSAT				
PULSE				
R PULSEPR				
RESP				
SAD SAD				
SSSKNF				

As one can see, there is a large number of tests for vital signs, and it might be that you have some more, or some less in your study, and have already extended or subsetted the codelist for VSTESTCD.

In order to start developing a new ValueList for vital signs tests, use the menu "Transform - CodeList to ValueList":

⊱ Define.xml Designer 2022 by XML4Pharma							
File Edit Add	Transform	Validate	View	Extra	Options	Help	
🏻 🚘 📑 🧕	Enumerated	d Codelist t	to Code	ListItem	Codelist		
CodeList to ValueList							

This results in:

Selected (CodeLists to convert to ValueLists								
?	CL.C96784.TUTESTCD - Tumor or Lesion Identification Test Code								
_	CL.C96783.TUTEST - Tumor or Lesion Identification Test Name								
	CL.C123650.TUIDRS - Tumor or Lesion Identification Test Results								
	CL.C96779.TRTESTCD - Tumor or Lesion Properties Test Code								
	CL.C96778.TRTEST - Tumor or Lesion Properties Test Name								
	CL.C124309.TRPROPRS - Tumor or Lesion Properties Test Result								
	CL.C100148.UPD1TC - Unified Parkinson's Disease Rating Scale Question								
	CL.C100147.UPD1TN - Unified Parkinson's Disease Rating Scale Question								
	CL.C71620.UNIT - Unit								
	CL.C66770.VSRESU - Units for Vital Signs Results								
	CL.C103489.UPS01TC - Urgency Perception Scale Questionnaire Test Cod								
	CL.C103488.UPS01TN - Urgency Perception Scale Questionnaire Test Nar								
	CL.C102590.VCNEVD - Vaccination Evidence Source								
	CL.C66741.VSTESTCD - Vital Signs Test Code								
	CL.C67153.VSTEST - Vital Signs Test Name								
	CL.C117746.HEPENCGR - West Haven Hepatic Encephalopathy Grade								
	Coarch								
	Search Find Next Find Previous								
	Create simple 'WhereClause' automatically								
	OK Cancel								

and select the codelist "CL.C66741.VSTESTCD - Vital Signs Test Code" (you can search for a specific codelist using the "Search" field and buttons)

If you started from a set of SAS-XPT files, and have subsetted this codelist yet, you will of course select the subsetted codelist²².

Remark the checkbox "Create simple 'WhereClause' automatically. When holding the mouse over it, more explanation is displayed:

Crea	te simple 'WhereClause' automatically
	When checked, a simple 'WhereClause' will be created automatically for each entry in the ValueList.
	For example for a ValueList for LBTESTCD, with entries 'ALB', 'ALP',,
	WhereClauses 'where LBTESTCD='ALB" and 'where LBTESTCD='ALP" will be created.
	This requires the codelist to have been assigned to a variable

²²In case you selected a codelist with many entries, the system will first provide a message that it might be useful to subset the codelist first. You can then still interrupt the process and do the subsetting first. For example, the by CDISC published codelists for LBTESTCD and LBTEST have several hundred entries, but you will not have used them all in your study, so you should then subset these codelists first.

We will explain this in detail in the next section "Automatically assigning WhereClauses to ValueList entries". We will however leave the checkbox unchecked for now.

CL.C66741.VS	CLC66741.VSTESTCD - Vital Signs Test Code										
New ValueList (OID:				[VL.CL.C66741.V	STESTCD				
OID	Name	Data Type	Length	Sign.Digits	Origin	Comment	Description	def:DisplayFor	Method	CodeList	WhereClau
IT.ABSKNF	ABSKNF						Abdominal Sk				WC.IT.ABSK
IT.BMI	BMI						Body Mass In				WCJT.BMI
IT.BODLNGTH	BODLNGTH						Total Body Le				WC.IT.BODI
IT.BODYFATM	BODYFATM						Body Fat Mea				WCJT.BODY
IT.BSA	BSA						Body Surface				WCJT.BSA
IT.DIABP	DIABP						Diastolic Bloo				WC.IT.DIAB
IT.FARMCIR	FARMCIR						Forearm Circ				WOJT.FARM
IT.FRMSIZE	FRMSIZE						Body Frame S				WCJT.FRMS
IT.HDCIRC	HDCIRC						Head Circumf				WC.IT.HDC
IT.HEIGHT	HEIGHT						Height				WC.IT.HEIG
IT.HIPCIR	HIPCIR						Hip Circumfer				WC.IT.HIPC
IT.HR	HR						Heart Rate				WCJT.HR
IT.IDEALWT	IDEALWT						Ideal Body We				WOJTJDEAL
IT.KNEEHEEL	KNEEHEEL						Knee to Heel				WOJT KNEE
IT.LBM	LBM						Lean Body Ma.				WOITLBM
IT.MAP	MAP						Mean Arterial				WC IT MAP
IT.OXYSAT	OXYSAT						Oxvoen Satur				WCJT.OXYS
IT.PULSE	PULSE						Pulse Rate				WOITPULS
IT.PULSEPR	PULSEPR						Pulse Pressure				WOJT PULS
IT.RESP	RESP						Respiratory R				WO IT RESP
IT.SAD	SAD						Sagittal Abdo				WC IT SAD
IT SSSKNF	SSSKNE						Subscapular				WC IT SSSE
IT.SYSBP	SYSBP						Systolic Blood				WOJT SYSE
IT.TBW	TBW						Total Body Wa				WC IT TBW
IT.TEMP	TEMP						Temperature				WC IT TEME
1											
		Insert	row					Remo	we row		
]					

After clicking "OK", this immediately leads to the following dialog and table:

In the upper right part of the dialog, a new identifier (OID) is proposed for the ValueList, consisting of the prefix "VL." and the OID of the original codelist:

	-
VL.CL.C66741.VSTESTCD	

You can change it and assign another OID, but in most cases you will probably want to keep the proposed one, as it immediately shows that the ValueList was derived from the "VSTESTCD" codelist.

The table contains a list of all items in the codelist:

🛓 Codel	List to ValueList	-		_		_	
2							
•	CL.C66741.VST	ESTCD - Vital Si	gns Test Code				
	New ValueList (DID:				[VL.CL
	OID	Name	Data Type	Length	Sign.Digits	Origin	С
	IT.ABSKNF	ABSKNF					
	IT.BMI	BMI					
	IT.BODLNGTH	BODLNGTH					
	IT.BODYFATM	BODYFATM					
	IT.BSA	BSA					
	IT.DIABP	DIABP					
	IT.FARMCIR	FARMCIR					
	IT.FRMSIZE	FRMSIZE					
	IT HDCIRC	HDCIRC					

The red fields indicate that these are mandatory fields, and the you must provide a value for it.

We can now starting editing the table. In first instance, we want to delete those rows (test codes) that are not applicable to our study. Usually this will be easier if we already subsetted the vital signs codelist, and use that as a starting point for our valuelist. Deleting the unnecessary rows can e.g. lead to:

CL.C66741.V	STESTCD - Vital Sig	ns Test Code						
New ValueList	t OID:						VL.CL.C66741.	/5
OID	Name	Data Type	Length	Sign	Origin	Com	Description	_
IT.BMI	BMI						Body Mass Index	
IT.DIABP	DIABP						Diastolic Blood Pressure	
IT.FRMSIZE	FRMSIZE						Body Frame Size	_
IT.HEIGHT	HEIGHT						Height	_
IT.HR	HR						Heart Rate	
IT.SYSBP	SYSBP						Systolic Blood Pressure	
IT.TEMP	TEMP						Temperature	
IT.WEIGHT	WEIGHT						Weight	
IT.WSTCIR	WSTCIR						Waist Circumference	-

Remark that one can remove more than one row at the time by using the "Ctrl" key when selecting rows. Then use the "Remove Row" button to remove the selected rows. Usually the field "Description" is very useful to understand the codes from the codelist. Use it!

For each remaining test, we now need to assign the appropriate data type. Usually, this is immediately clear by a look on the CRF. For example for BMI we expect a float, whereas for "Hearth Rate" we probably inspect an integer (depending on the unit that was used, mostly "beats per minute"). For "Body Frame Size" however, there will usually be an associated codelist (checkboxes on the CRF) with choices like "small", "large" etc..

For example:

IT.HR HR	in				
IT.SYSBP SYS	BP in				
IT.TEMP TEM	P 🗖				
IT.WEIGHT WEI	GHT 🗖				
IT.WSTCIR WST	CIR fl	loat	•		
	in	nteger			
	fl	loat			
	te	ext			
	d	ate	=		
	p	artialdate			
	ti	ime			
	p	artialtime			
	d	latetime	•		

It is a good idea now to click the button "Validate" (the cyan colored button near the bottom). This will lead to:

OID	Name	Data Type	Length	Sign	Origir
IT.BMI	BMI	float			
IT.DIABP	DIABP	integer			
IT.FRMSIZE	FRMSIZE	text			
IT.HEIGHT	HEIGHT	float			
IT.HR	HR	integer			
IT.SYSBP	SYSBP	integer			
IT.TEMP	TEMP	float			
IT.WEIGHT	WEIGHT	float			
IT.WSTCIR	WSTCIR	integer			

I.e. we also need to provide the maximal length for each of the test $results^{23}$.

We fill the "Length" field, and then validate again using the "Validate" button:

CL.C66741.	/STESTCD - Vit	al Signs Test	Code								
New ValueLi	st OID:					N	/L.CL.C66741.VST	ESTCD			
OID	Name	Data Type	Length	Sign	Or	Comment	Description	def:DisplayFor	Method	CodeList	WhereClaus
IT.BMI	BMI	float	5				Body Mass Index				WC.IT.BMI
IT.DIABP	DIABP	integer	3				Diastolic Blood				WC.IT.DIABP
IT.FRMSIZE	FRMSIZE	text	10				Body Frame Size				WO.IT.FRMSIZ
IT.HEIGHT	HEIGHT	float	5				Height				WO.IT.HEIGHT
IT.HR	HR	integer	3				Heart Rate				WC.IT.HR
IT.SYSBP	SYSBP	integer	3				Systolic Blood				WC.IT.SYSBP
IT.TEMP	TEMP	float	4				Temperature				WO.IT.TEMP
IT.WEIGHT	WEIGHT	float	5				Weight				WOJT.WEIGHT
IT.WSTCIR	WSTCIR	integer	3				Waist Circumfe				

Everything looks all right, except that the "WhereClause" field is still colored red.

We also still need to assign a codelist to the item "Body Frame Size" (FRMSIZE) as the CRF shows checkboxes (choices) for this. It might be that we have already generated a separated codelist for this, but it might also be that we can use a CDISC codelist for "body frame size". For example,

²³Remark that for datatype "Date", "Time" and similar data types, no maximal length needs to provided.

clicking the "CodeList" field for "SIZE" leads to:

_				
	Codel	list		Where
				WC.IT.B
				WO.IT.D
	CL.C66733.SI7F		•	WC.IT.F
	CL.C66733.SIZE			WC.IT.H
	CL.C76351.SKINCLA	c		WC.IT.H
_	CL.C112024.SRTEST	CL.C66733.S	IZE	WC.IT.S
_	CL.C112023.SRTES1	Size		WC.IT.T
_	CL.C74561.SKINTYP	Possible valu	es:	WC.IT.W
	CL C102587 RISKSO			WC.II.V
	CL C70733 SDECCO	SWALL		
	CL.C70734 CDECTV	SWALL		
	CL.C/8/34.SPECTYF	Έ	•	

When making decisions about which codelist is applicable, holding the mouse over a selected item always shows the allowed values, which is very helpfull.

Also remark that "SIZE" is an extensible codelist, so if you have more checkboxes on your CRF for "body frame size" than "Small", "Medium" or "Large", you may want to extend that codelist (see further on).

So we now have:

🍰 Codel	list to ValueList	_	_									×
?												
_	CLC66741.VSTESTCD - Vital Signs Test Code											
	New ValueList OID: VL.CL.C66741.VSTESTCD											
	OID	Name	Data Type	Length	Sign.Digits	Origin	Comment	Description	def:DisplayFormat	Met	CodeList	WhereClause
	IT.BMI	BMI	float	5				Body Mass Index				WC.IT.BMI
	IT.DIABP	DIABP	integer	3				Diastolic Blood				WC.IT.DIABP
	IT.FRMSIZE	FRMSIZE	text	10				Body Frame Size			CL.C66733.SIZE	WC.IT.FRMSIZE
	IT.HEIGHT	HEIGHT	float	5				Height				WC.IT.HEIGHT
	IT.HR	HR	integer	3				Heart Rate				WC.IT.HR
	IT.SYSBP	SYSBP	integer	3				Systolic Blood				WC.IT.SYSBP
	IT.TEMP	TEMP	float	4				Temperature				WOJT.TEMP
	IT.WEIGHT	WEIGHT	float	5				Weight				WC.IT.WEIGHT
	IT.WSTCIR	WSTCIR	integer	3				Waist Circumfe				WC.IT.WSTOIR

Working with the "Where Clause" is often seen as the most difficult part of define.xml 2.0. However with the wizards provided in our software, this becomes very easy. As of define.xml 2.0, a "Where Clause" <u>must</u> be provided for each item in a valuelist. It is a "machine-readable" as well as "human-readable" description about when the valuelist item applies. For example, "FRMSIZE" information (data type, length, associated codelist) will be used when ... the value of VSTESTCD is "FRMSIZE"... Looks logical isn't it?

The define.xml 2.0 specification (and SDTM sample file) however shows a more complicated example for VSORRESU (vital signs original result unit) where the value is "inches" when COUNTRY (in DM) is "USA" and "centimeter" when COUNTRY is either "MEX" (Mexico) or "CAN" (Canada).

Now click the most-right cell "WC.IT.BMI". This opens a wizard for setting up the "where clause":

Designin	g/Updating WhereClause for Item: IT.BMI	×
?	OID: Comment:	WC.IT.BMI
		Number of RangeChecks: $1\frac{1}{2}$
		Show 'Where' clause
	Comparator: EQ 💌 Item OID:	STUDYID DOMAIN USUBJID SUBJID DM.RF STDTC DM.RF STDTC

The identifier (the OID) has already been created, you can however change it (mostly you will want to keep the one that is suggested).

The "Comment" field only needs to be populated when your "where" contains variables from other domains. This is well explained in the define.xml 2.0 specification in example 4.2.2.2 about the metadata for VSORRESU (vital signs original result units) depending on the country (USA, Mexico or Canada).

If you fill something into the "Comment" field, a "def:CommentDef" element will be generated and a reference to it is added in the "ItemRef" element within the ValueList.

Then you can set the number of condition "RangeChecks" that will be combined. For example, if you want to have a condition like "where the test code is BMI <u>and</u> the age is less than 70), then you will need two "RangeChecks" and need to set the "Number of RangeCheck" spinner to 2. In our case however, we only need to state "where the vital signs test code is BMI", so we can leave the spinner to "1". Also the value for the "Comparator" can be left to "EQ" as it means "equals to" as is also shown by the tooltip:

			STUDYID
			DOMAIN
			USUBJID
			SUBJID
		1	DM.RF STDTC
Comparator:	EQ 💌	Item OID:	DM.RFENDTC
	EQ	1	DM.RFX STDTC
	NE		DM.RFXENDTC
	LT Equal		DM.RFICDTC
	LE		DM.RFPENDTC
	GT		DM.DTHDTC
	GE		
	IN		
	NOTIN		

We then still need to assign the SDTM/SEND/ADaM variable in the condition definition which in our case is "VSTESTCD". It can quickly be found by typing the first characters "VS" and then selecting "VS.VSTESTCD":



The "CheckValue" then still has to be typed in. It is "BMI".

You can always check what you have developed as a condition by clicking the button "Show 'Where' Clause". The human readable condition is then displayed. In our case:

	RS.VISIT Where-Clause Expression	
Comparator: EQ 💌	i where VSTESTCD EQ 'BMI'	ieckValue: BMI
	VS.VSTESTCD VS.VSTEST VS.VSCAT	

Now click OK until back in the table with all ValueList items.

Setting the "WhereClause" now needs to be repeated for all the items in the "ValueList" table. Regularly use the "Validate" button to ensure that (at least structurally) everything is correct. Finally, one would obtain a table without any red cells anymore:

New ValueL	ist OID:						VL.CL.C66741.VSTESTC	D			
OID	Name	Data Type	Length	Sign.Digits	Origin	Comment	Description	def.DisplayFor	Method	CodeList	WhereClau
IT.BMI	BMI	float	5		-		Body Mass Index				WC.IT.BMI
IT.DIABP	DIABP	integer	3				Diastolic Blood Pressure				WC.IT.DIABP
IT.FRMSIZE	FRMSIZE	text	10				Body Frame Size			CL.C66733.SIZE	WC.IT.FRMSIZ
IT.HEIGHT	HEIGHT	float	5				Height				WC.IT.HEIGH
IT.HR	HR	integer	3				Heart Rate				WC.IT.HR
IT.SYSBP	SYSBP	integer	3				Systolic Blood Pressure				WC.IT.SYSBP
IT.TEMP	TEMP	float	5				Temperature				WC.IT.TEMP
IT.WEIGHT	WEIGHT	float	5				Weight				WC.IT.WEIGH
IT.WSTCIR	WSTCIR	integer	3				Waist Circumference				WC.IT.WSTCI

In most cases, you will also need to add an "Origin" to each of the variables (you can still do it later too, but with the danger that it is forgotten). In order to add an "Origin", click in the "Origin" cell. For example, for "BMI":

Designing	g/Updating Origin for Item: IT.BMI
?	Origin type: Assigned
	Electronic Data Transfer
	 CRF Document (leaf) ID: No def:leaf elements have been defined yet No page details
	 Page list (physical reference) Named destinations Page list / List of named destinations
	O Page range: first page - last page First page: Last page:
	OK Cancel

In this case, the "Origin" is derived, as the BMI was not captured on the CRF, but was calculated from "height" and "weight". So, in this case, one needs to set it to "Derived":

Designing	g/Updating Origin for Item: IT.BMI
?	Origin type: Origin type:
	Protocol
	Derived
	 Electronic Data Transfer
	○ CRF
	Document (leaf) ID: No def:leaf elements have been defined yet
	○ No page details
	O Page list (physical reference)

and after clicking "OK":

OID	Name	Data Type	Length	Sign.Digits	Origin	Comment	Description	def:	Method	CodeList
IT.BMI	BMI	float	5		Derived		Body Mass Index			
IT.DIABP	DIABP	integer	3				Diastolic Blood Pressure			
IT.FRMSIZE	FRMSIZE	text	10				Body Frame Size			CL.C66733.SIZE
IT.HEIGHT	HEIGHT	float	5				Height			
IT.HR	HR	integer	3				Heart Rate			
IT.SYSBP	SYSBP	integer	3				Systolic Blood Pressure			
IT.TEMP	TEMP	float	5				Temperature			
IT.WEIGHT	WEIGHT	float	5				Weight			
IT.WSTCIR	WSTCIR	integer	3				Waist Circumference			

Now we should also provide the "method" <u>how</u> the BMI was calculated. For this click in the "Method" cell. This results in:



and one can e.g. add:



And after clicking "OK":

ent	Description	def:	Method	CodeList
	Body Mass Index		BMI was calculate	
	Diastolic Blood Pressure			
	Body Frame Size			CL.C66733.SIZE
	Height			

Adding an "Origin" and a "Method" will automatically create a "def:Origin" and a "MethodDef" element in the define.xml (with references), so you will later still be able to change the information using the regular methods.

In a similar way, one can always add a "Comment" by clicking in the "Comment" cell. This will generate a "def:CommentDef" with a reference to it, so that you can still later change the information.

Essentially, we should add an "Origin" to each of the newly created value-level variables, for example for "DIABP":



Now, the diastolic blood pressure was captured using the CRF, but we haven't added the information yet about where the annotated CRF resides. So, we are not allowed yet to add the page information yet. When "CRF" is clicked, the following message appears:



However, as we will to have to add an "Origin" to each of the variables, domain variables as well as valuelist variables, we can do this later when assigning origins to all of them.

When done editing the table, then clicking "OK" leads to the following message:

Message	
(1)	Number of new ValueLists created: 1 Number of new value-level Variables (ItemDef elements) created: 9 Number of new method definitions (MethodDef elements) created: 1 Number of new comments (def:CommentDef elements) created: 2 Number of new 'Where Clauses' (def:WhereClauseDef elements) created: 9 You will still need to edit/extend the information each of the generated ValueLists. For each of the newly generated value-level Variables, you will need to add the Origin (when not done yet), with references to the pages or sections on the CRF (when Origin-Type='CRF').
	OK

showing a nice summary of what has been created.

You can now always edit the created valuelists, variable definitions, comments, method descriptions and "where clauses" by selecting the appropriate tab. For example, for "ValueLists":

Dataset Definitions	Variable Definitions Codelists
Annotated CRFs	Supplemental Documents
	TESTCD

or for "Where Clauses":

Dataset Definitions Variable Definitions Codelist	Imputation Methods Presenta	ations Condition Definitions Me
Annotated CRFs Supplemental Documents	ValueList Definitions	WhereClause Definitions
OID	CommentOID	
CIT.DAYS		
🖸 🔍 WC.IT.BMI		
CIT.DIABP	COM.WC.IT.DIABP	
C WC.IT.HEIGHT		
C WC.IT.HR		
C WC.IT.SYSBP		
K WC.IT.WEIGHT		
KWC.IT.WSTCIR		
		8

Also remark that new variables (at the "value" level) have been created. These appear in the "Variable Definitions" table:

Dataset Definitions	Variable Definition	ons Codelists	Imputation Meth	iods Presenta	tions Condition	Definitions
Annotated CRFs	Suppleme	ntal Documents	ValueLis	t Definitions	WhereClaus	e Definitions
OID	Name	DataType	Length	SignificantDigits	SASFieldName	SDSVarNam
TS.TSPARMCD	TSPARMCD	text	80			
STS.TSPARM	TSPARM	text	80			
STS.TSVAL	TSVAL	text	80			
STS.TSVALNF	TSVALNF	text	80			
STS.TSVALCD	TSVALCD	text	80			
STS.TSVCDREF	TSVCDREF	text	80			
STS.TSVCDVER	TSVCDVER	text	80			
📴 🔍 IDVAR	IDVAR	text	80			
📴 🔍 IDVARVAL	IDVARVAL	text	80			
RELTYPE	RELTYPE	text	80			
RELID	RELID	text	80			
🖸 🔍 QNAM	QNAM	text	80			
🖸 🔍 QLABEL	QLABEL	text	80			
🖸 🔍 QVAL	QVAL	text	80			
📮 🔍 QORIG	QORIG	text	80			
📴 🔍 QEVAL	QEVAL	text	80			
🖺 🔍 IT.BMI	BMI	float	5			
T.DIABP	DIABP	integer	3			
T.FRMSIZE	FRMSIZE	text	8			
K KIT.HEIGHT	HEIGHT	integer	3			
🖺 🔍 IT.HR	HR	integer	3			
T.SYSBP	SYSBP	integer	3			
IT.TEMP	TEMP	float	5			
T.WEIGHT	WEIGHT	float	5			
C AIT.WSTCIR	WSTCIR	integer	3			

After you have defined where the annotated CRF resides, and how that file is named, you will be able to add the "Origin" to each of the variables, be it domain variables or valuelist variables.

Automatically assigning WhereClauses to ValueList entries

When creating a valuelist starting from a codelist, you will have noticed the checkbox "Create simple 'WhereClause' automatically":

CL.C102590.VCNEVD) - Vaccination Evidence	Source
CL.C66741.VSTESTC	CD - Vital Signs Test Cod	e
CL.C67153.VSTEST -	Vital Signs Test Name	
CL.C117746.HEPENC	GR - West Haven Hepat	c Encephalopathy Grade
		•
Search		
Search		
Search Search	Find Next	Find Previous
Search Search Create simple 'Where	Find Next eClause' automatically	Find Previous

When holding the mouse over it, more explanation is displayed:

Creat	te simple 'WhereClause' automatically
	When checked, a simple 'WhereClause' will be created automatically for each entry in the ValueList.
	For example for a ValueList for LBTESTCD, with entries 'ALB', 'ALP',,
	WhereClauses 'where LBTESTCD='ALB" and 'where LBTESTCD='ALP" will be created.
_	This requires the codelist to have been assigned to a variable

In order to generate such "WhereClauses" you will already need to have assigned the codelist to a variable. In the case of "Vital Signs Code" (CL.C66741.VSTESTCD), you will almost surely have assigned this codelist to the variable "VSTESTCD":

xtra information for: Iter	nDef, with OID = VS.VST	ESTCD				
RangeCheck	CodeList Reference	Role Alias	Origin ValueList I	Reference		
Descrip	tion Qu	estion	ExternalQues	tion	MeasurementUnitRef	
CodeListOID				Codelis	t	
CL.C66741.VST	ESTCD			definitio	ns:	
				CL.C12	4309.TRPROPRS	-
				CL.C10	0148.UPD1TC	
				CL.C10	0147.UPD1TN	
				CL.C71	620.UNIT	
				CL.C66	770.VSRESU	
				CL.C10	3489.UPS01TC	
				CL.C10	3488.UPS01TN	
				CL.C10	2590.VCNEVD	
				CL.C66	741.VSTESTCD	
				CL.C67	153.VSTEST	
				CL C11	7746 HEDENCOR VITAL SIGNS TEST CODE	

If the codelist was not assigned to any SDTM/SEND/ADaM variable, and the checkbox "Create

simple 'WhereClause' automatically" is checked, a warning message will be displayed:



If however, the codelist was already assigned (for example the VSTESTCD codelist to the VSTESTCD variable), all the "WhereClauses" will be generated automatically. These are simple "WhereClauses" with the structure:

"where *testcode* EQ 'valuelistvalue"

For example, for the valuelist variable "DIABP" from the codelist VSTESTCD, the "WhereClause" that is generated corresponds to:

"where VSTESTCD EQ 'DIABP""

So, let us try this out for the codelist VSTESTCD, and generate a ValueList from it. When the checkbox "Create simple 'WhereClause' automatically" is checked, this results (after a few seconds) in:

New ValueList OID: VL.CLC.66741.VSTESTCD OID Name Data Type Length Sign.Digits Origin Comment Description defDisplayFor Method CodeList WhereCL TABSKNF ABSKNF BMI WC.IT.ABS WC.IT.ABS WC.IT.BMI TI.BMI BMI BMI WC.IT.BMI WC.IT.BMI WC.IT.BMI T.BODLINGTH BODINGTH BMIR WC.IT.BMI WC.IT.BMI WC.IT.BMI T.BODLINGTH BODINGTH BODINGTH WC.IT.BOI WC.IT.BOI T.BODINGTH BODYFATM BODYFATM WC.IT.BOI WC.IT.BOI T.BAR BODYFATM BODYFATM WC.IT.BOI WC.IT.BOI T.BAR BODYFATM BODYFATM WC.IT.BOI WC.IT.BOI T.BAR BODYFATM WC.IT.BOI WC.IT.BOI WC.IT.BOI T.BAR BODYFATM WC.IT.BOI WC.IT.BOI WC.IT.BOI T.BAR BODYFATM BODYFATM WC.IT.BOI WC.IT.BOI T.BAR BSA	CL.C66741.VST	FESTCD - Vital Sig	gns Test Code									
OID Name Data Type Length Sign Digits Origin Comment Description defDisplayFor Method CodeList WhereCl IT ABSKNF ABSKNF ABSKNF WC.IT ABS WC.IT ABS WC.IT ABS IT.BMI BMI BMI WC.IT BMI WC.IT BMI WC.IT BMI IT.BODLINGTH BODLINGTH BODLINGTH WC.IT BMI WC.IT BMI IT.BODDVFATM BODVFATM BODVFATM WC.IT BMI WC.IT BMI IT.BASA BSA WC.IT BMI WC.IT BMI WC.IT BMI WC.IT BMI IT.BASA BSA WC.IT BMI BODVFATM WC.IT BMI WC.IT BMI IT.BAR BSA WC.IT BMI BODVFATM WC.IT BMI WC.IT BMI IT.BAR BSA BODVFATM WC.IT BMI WC.IT BMI WC.IT BMI IT.BAR BSA BODVFATM WC.IT BMI WC.IT BMI WC.IT BMI IT.BAR BSA BODVFATM WC.IT BMI WC.IT BMI WC.IT BMI IT.FAMSIZE </th <th>New ValueList (</th> <th>DID:</th> <th></th> <th></th> <th></th> <th> </th> <th>VL.CL.C66741.V</th> <th>STESTCD</th> <th></th> <th></th> <th></th> <th></th>	New ValueList (DID:					VL.CL.C66741.V	STESTCD				
IT ABSKNF ABSKNF WCIT ABS IT.BM BM WCIT ABS IT.BM BM WCIT BM IT.BMR BMR WCIT BMR IT.BODLNGTH BODLNGTH WCIT BM IT.BODLYFATM BODLYGTH WCIT BOD IT.BAS BODLYGTH WCIT BOD IT.BAS BSA WCIT BOD IT.FARMIZE FRMCIR WCIT BOD IT.FARMIZE FRAMCIR WCIT FRI IT.FARMIZE FRMSIZE WCIT FRI IT.HECIR HDCIRC WCIT HD IT.HECIR HICIR WCIT HE	OID	Name	Data Type	Length	Sign.Digits	Origin	Comment	Description	def:DisplayFor	Method	CodeList	WhereClau
IT EMI BMI W0.IT EMI IT.BMI BMR W0.IT EMI IT.BOULNGTH BMR W0.IT EMI IT.BOOLNGTH BODVFATM W0.IT EMI IT.BOOLNGTH BODVFATM W0.IT EMI IT.BOOLNGTH BODVFATM W0.IT EMI IT.BOOLNGTH BODVFATM W0.IT EMI IT.BAB BSA W0.IT EMI IT.BAB BSA W0.IT EMI IT.BAB BSA W0.IT EMI IT.BAB BSA W0.IT EMI IT.FARMCIR FRAMCIR W0.IT.FAMI IT.FARMICIR FRAMICIR W0.IT.FAMI IT.FARMICIR FRAMICIR W0.IT.FAMI IT.HECIR FRMSIZE W0.IT.FAMI IT.HECIR HDCIRC W0.IT.HID IT.HEIGHT HEIGHT W0.IT.HID IT.HR HIR W0.IT.HID	IT.ABSKNF	ABSKNF						ABSKNF				WC.IT.ABSK
IT.BMR BMR WC.IT.BM IT.BODLNGTH BODLNGTH BODLNGTH WC.IT.BM IT.BODLYFATM BODVFATM BODVFATM WC.IT.BOJ IT.BAR BSA BODVFATM WC.IT.BM IT.BAR BSA WC.IT.BOJ WC.IT.BOJ IT.BAR BSA WC.IT.BM WC.IT.BM IT.DIABP DIABP WC.IT.BAR WC.IT.BAR IT.FARMCIR FARMCIR WC.IT.FRM WC.IT.FRM IT.FARMCIR FARMCIR WC.IT.FRM WC.IT.FRM IT.HEIGHT HEIGHT WC.IT.HD WC.IT.HD IT.HEIGHT HIPCIR WC.IT.HIP WC.IT.HIP IT.HIPCIR HIPCIR WC.IT.HIP WC.IT.HIP	IT.BMI	BMI						BMI				WC.IT.BMI
IT BODLNGTH BODLNGTH WC.IT.BOJ IT.BODYFATM BODYFATM WC.IT.BOJ IT.BA BSA WC.IT.BOJ IT.BA BSA WC.IT.BOJ IT.DIABP DIABP WC.IT.DIA IT.F.ARMCIR FARMCIR WC.IT.DIA IT.F.ARMCIR FARMCIR WC.IT.DIA IT.F.ARMCIR FARMCIR WC.IT.FRM IT.F.RMSIZE FRMSIZE WC.IT.FRM IT.HDCIRC HDCIRC WC.IT.HD IT.HECIR HIEGHT WC.IT.HIP IT.HIPCIR HIPCIR WC.IT.HIP IT.HR HR WC.IT.HIP	IT.BMR	BMR						BMR				WC.IT.BMR
IT BODYFATM BODYFATM BODYFATM WCIT BOJ IT.BSA BSA BSA WCIT BOJ IT.BABP DIABP DIABP WCIT DIA IT.FARMCIR FARMCIR WCIT FAR WCIT FAR IT.FARMCIR FARMCIR WCIT FAR WCIT FAR IT.FARMCIR FARMCIR WCIT FAR WCIT FAR IT.HOLIRC HOLIRC WCIT FAR WCIT FAR IT.HOLIRC HDCIRC WCIT FAR WCIT FAR IT.HEIGHT HEIGHT WCIT HIP WCIT HIP IT.HIPCIR HIPCIR WCIT HIP WCIT HIP IT.HR HIR WCIT HIP WCIT HIP	IT.BODLNGTH	BODLNGTH						BODLNGTH				WC.IT.BODL
IT BSA BSA WCIT BS/ IT.DIABP DIABP WCIT BS/ IT.DIABP DIABP WCIT DA IT.FARMCIR FARMICIR WCIT FAR IT.FRMSIZE FRMSIZE WCIT.FAR IT.HORCR HOCIRC WCIT.FAR IT.HEIGHT HOCIRC WCIT.HD IT.HEIGHT HICIR WCIT.HIP IT.HIPCIR HIPCIR WCIT.HIP IT.HIPCIR HIPCIR WCIT.HIP IT.HIPCIR HIPCIR WCIT.HIP	IT.BODYFATM	BODYFATM						BODYFATM				WC.IT.BODY
IT DIABP DIABP WCIT DIA IT,FARMCIR FARMCIR WCIT DIA IT,FARMCIR FARMCIR WCIT DIA IT,FARMCIR FARMCIR WCIT DIA IT,FARMSIZE FRMSIZE WCIT FAR IT HDCIRC HDCIRC WCIT HD IT,HEIGHT HEIGHT WCIT.HIP IT,HIPCIR HIPCIR WCIT.HIP IT,HR HR WCIT.HIP	IT.BSA	BSA						BSA				WC.IT.BSA
IT FARMCIR FARMCIR W0.IT FAR IT FRMSIZE FRMSIZE W0.IT FAR IT FRMSIZE FRMSIZE W0.IT FAR IT HDCIRC HDCIRC W0.IT FAR IT HEIGHT HBCHT W0.IT HIP IT HIPCIR HIPCIR W0.IT HIP IT HIPCIR HIPCIR W0.IT HIP IT HR HR W0.IT HIP	IT.DIABP	DIABP						DIABP				WC.IT.DIABF
IT.FRMSIZE FRMSIZE WCIT.FRI IT.HDCRC HDCRC WCIT.FRI IT.HDCRC HDCRC WCIT.HD IT.HEIGHT HEIGHT WCIT.HD IT.HIPCIR HIPCIR WCIT.HIP IT.HIPCIR HIPCIR WCIT.HIP IT.HR HR WCIT.HIP	IT.FARMCIR	FARMCIR						FARMCIR				WC.IT.FARM
ITHDCIRC HDCIRC W0.ITHD IT.HEIGHT HEIGHT W0.IT.HE IT.HIPCIR HIPCIR W0.IT.HE IT.HR HR W0.IT.HE IT.R HR W0.IT.HE	IT.FRMSIZE	FRMSIZE						FRMSIZE				WC.IT.FRMS
IT.HEIGHT HEIGHT WC.IT.HEI IT.HIPCIR HIPCIR WC.IT.HEI IT.HIPCIR HIPCIR WC.IT.HEI IT.HR HR WC.IT.HEI IT.HR HR WC.IT.HEI	IT.HDCIRC	HDCIRC						HDCIRC				WC.IT.HDCI
IT.HIPCIR HIPCIR WC.IT.HIP IT.HIR HR WC.IT.HIR HR HR WC.IT.HIR	IT.HEIGHT	HEIGHT						HEIGHT				WC.IT.HEIGI
IT HR HR WOITH HR WOITH HR WOITH HR	IT.HIPCIR	HIPCIR						HIPCIR				WC.IT.HIPCI
	IT.HR	HR						HR				WC.IT.HR
IT.IDEALWT IDEALWT WC.IT.IDE	IT.IDEALWT	IDEALWT						IDEALWT				WC.IT.IDEAL
	IT.LBM	LBM						LBM				WC.IT.LBM
IT.LBM LBM WC.IT.LBM WC.IT.LBM	IT.MAP	MAP						MAP				WC.IT.MAP
IT.LBM LBM WC.IT.LBM WC.IT.LBM WC.IT.LBM WC.IT.LBM	TT THE PROPERTY OF				1						1	

One now notices that the "WhereClause" cell is not colored, i.e. the software generated a technically correct "WhereClause" for each valuelist entry. One can see the generated "WhereClause" by holding the mouse over the cell. For example:

CodeList	WhereClause					
	WC.IT.ABSKNF	-	i			
	WC.IT.BMI					
	WC.IT.BMR					
	where VSTESTCD EQ 'BMI'					

for the cell "WC.IT.BMI"

You can of course still refine the "WhereClause" by clicking on the cell. For example, for the "WhereClause" for the valuelevel variable DIABP:

Designing	g/Updating WhereClause for Item: IT.DIABP		×					
?	OID: WC.IT.DIABP							
	Number of RangeChecks: 1							
	Show 'Where' clause							
		VS.VSTESTCD						
		VS.VSTEST						
		VS.VSCAT						
		VSVSSCAT						
	Comparator: EQ Vitem OID:	VSVSORRES	CheckValue: DIABP					
		VS.VSORRESU	Silvertuner Birth					
		VS.VSSTRESC						
		VS.VSSTRESN						
		VS.VSSTRESU						
		VS.VSSTAT						

For example, if you would have a "WhereClause" stating "where the VSTESTCD is 'DIABP' and the country of measurement is 'USA' or 'Canada'", you can easily do so using the wizard by adding an extra condition, like:



and when clicking the "Show 'WhereClause'" button, the "human-readable" expression is displayed:



You will probably be able to generate over 90% of the "WhereClauses" automatically in this way, but it is important to understand that checking them is necessary.

After clicking "OK" until one gets in the main window, and then selecting the tab "WhereClause Definition", one sees the newly generated "WhereClauses" in addition to any already generated:

<u>Important remark</u>: as generating valuelists from codelists is computing intensive, especially when also automatically generating the "WhereClauses" automatically, it is recommended to subset the codelist first to what appears in the CRFs when starting from the by CDISC published codelists. For example, the LBTESTCD codelist has several hundred entries, and it can take 5-10 minutes to convert it to a valuelist (containing the several hundred entries) when also generating the "WhereClauses" automatically. Also, as you will need to assign the datatype, lengths, codelists etc.

for each valuelist variable, it makes sense to subset the codelist LBTESTCD first to what is exactly needed, and then to create a valuelist from that subset.

Dataset Definitions	Variable Definitions Co	odelists	putation Methods	Present	ations Conditio	n Definitions Method
Annotated CRFs	Supplemental Docu	iments	ValueList Defi	nitions	WhereClau	ise Definitions
OID				(CommentOID	
C WC.DA.DATESTCD	.DISPAMT					
🖸 🔍 WC.DA.DATESTCD	RETAMT					
WC.EG.EGTESTCE).INTP					
WC.EG.EGTESTCE	D.PRMEAN					
WC.EG.EGTESTCE).QRSDUR					
WC.EG.EGTESTCE	D.QTMEAN					
WC.EG.EGTESTCE).VRMEAN					
WC.EG.EGTESTCE).QTCB					
WC.EG.EGTESTCE).QTCF					
WC.IE.IETESTCD.E	EXCL01					
WC.IE.IETESTCD.E	EXCL02					

Of course, you can then still edit any "WhereClause" here. For example for EGTESTCD "PRMEAN":

	Number of Descent the start		
	NUMBER OF RangeChecks:		
	Show 'Where' claus	se	
			Ê
	IT.EG.EGTESTCD		_
	IT.EG.EGTESTCD IT.EG.VISIT		
	IT.EG.EGTESTCD IT.EG.VISIT IT.EG.VISITDY		
	IT.EG.EGTE STCD IT.EG.VISIT IT.EG.VISITDY IT.EG.VISITNUM IT EX DOMAIN		
Comparator: EQ V Item OID:	IT.EG.EGTE STCD IT.EG.VISIT IT.EG.VISITDY IT.EG.VISITNUM IT.EX.DOMAIN IT.EX.EXDOSE	CheckValue: PRMEAN	
Comparator: EQ Vitem OID:	IT.EG.EGTE STCD IT.EG.VISIT IT.EG.VISITDY IT.EG.VISITNUM IT.EX.DOMAIN IT.EX.EXDO SE IT.EX.EXDO SFRM	CheckValue: PRMEAN	

When clicking the "Edit" ("+") icon for WC.EG.EGTESTCD.PRMEAN, exactly the same wizard will show up as the one that was used during the generation of the "WhereClause". Of course one can also add new "WhereClauses" here, they do not need to be created from the "CodeList to ValueList" wizard at all.

Also **do not forget** to assign valuelists to SDTM/SEND/ADaM variables themselves. For example, you will usually assign the valuelist with the different vital sign test codes to VSORRES, as this then describes the metadata of VSORRES (datatype, length, codelist) depending on the value of VESTESTCD.

If you want to assign a valuelist to VSORRESU (and other --ORRESU and --STRESU) variables, you will first need to subset the "VSRESU" (Units for Vital Signs Results) codelist or "UNIT" codelist, and then generate a ValueList from it.

For example, for assigning the previously created ValueList to VSORRES, go to the "Variable Definitions" tab, select VSORRES,

Dataset Definitions Variable	Definitions Code	elists Im	putatio	n Methods	Presentat	ions Coi	ndition Definitions	Method Definiti
Annotated CRFs S	upplemental Docum	ents	Va	lueList Definit	tions	Where	Clause Definitions	Includes
	Sea	rch for: VS	ORRES	}				Search Find
	Search within: 🗌 All Columns							
	🖌 OID 🛛 🖌 Name 🔄 DataType 🔄 Length 🔄 SignificantDigits 🔙 SAS							
	SDSVarName Origin Origin Comment DisplayFormat CommentOID							
OID	Name	DataType		Length	Signit	ficantDigits	SASFieldName	SDSVarName
C C IT.VS.VSDY	VSDY	integer		3			VSDY	
C C IT.VS.VSORRES	VSORRES	text		30			VSORRES	
IT.VS.VSORRES.DIABP	VS.DIABPOR.O	integer		2			DIABPOR	
IT.VS.VSORRES.FRMSIZE	VS.FRMSIZE.O	text		6			FRMSZOR	
IT.VS.VSORRES.HEIGHT	VS.HEIGHT.OR	float		5	1		HEIGHTOR	
				-				

and then click the "Add Information" ("+") icon on the left of "VSORRES", and choose the tab "ValueList Reference":

⊱ Extra i	nformation for: ItemDef, with O	ID = IT.VS.VSORRES		-
?	RangeCheck CodeList F	Reference Role Alia: Question	s Origin ValueList Reference ExternalQuestion	
	ValueListOID VL.VS.VSORRES		Value List definitions:	
			VL.EG.EGORRES VL.EG.EGSTRESC	
			VL.IE.IEORRE S VL.LB.LBORRE S	
			VL.PE.PEORRES	

where you can simply drag-and-drop from the list of existing ValueList definitions on the right to the cell on the left, or click the "ValueListOID" cell, after which a list to choose from is displayed. This is explained in detail in the section: "Assigning valuelists to SDTM/SEND/ADaM variables".

P.S. As ValueLists only have an OID (but no "Name" attribute), it may be useful to assign a meaningful OID to each entry in the "ValueList" table.

Annotated CRF and Supplemental Documents

When doing a regulatory submission, you are still required to provide an "annotated CRF" in PDF format²⁴. Also, you will often want to supply supplemental documents, like a "reviewers guide", also in PDF format.

²⁴It would be much better if the requirements were that an ODM file with the study design must be delivered, where the questions are annotated with SDTM information. ODM is machine-readable, PDF isn't.

In define.xml, you can provide links to such documents, so that when the reviewer inspects your define.xml in the browser, a single click suffices to open the annotated CRF or supplemental document, ideally on the page of interest.

This is taken care of in define.xml documents by so-called "def:leaf" elements, allowing you to specify a reference ("href") to the document of interest.

 Study Metadata

 Method Definitions
 Comment Definitions
 Document links

 Is
 Includes
 Protocol/Trial Design
 Study Event Definitions

In order to provide such a leaf, click the tab "Document Links":

The following table is displayed:

D	href	1000
		1
		100
		100
		100
		100
		1000

you can now provide an ID and a reference (usually just the file name) for each document. For example, to define a link to an annotated CRF and to a "reviewers guide":

ID	href	
LF.aCRF	aCRF.pdf	
	Reviewers_Guide.pdf	
R.Q.		

When you then click the "Validate" button, you will notice that some information is still missing:

ID	href
LF.aCRF	aCRF.pdf
	Reviewers_Guide.pdf
Invalid content	
RIÓ	

So you will need to provide additional content. This can be provided by clicking on the "+" icon, leading to:

⊱ Extra in	formation for: leaf
?	

prompting you to provide a title by clicking the "+" icon again:

⊱ Extra ir	nformation for: leaf
?	
	Extra information for: title
	Annotated CRF for the CES study.
	OK Cancel

The same should then be done for the "reviewers guide". Revalidating then leads to:

ID	href	
C C C C C C C C C C C C C C C C C C C	aCRF.pdf	
	Reviewers_Guide.pdf	

Both "+" icons turned orange, meaning that additional information has been added and there are no validation errors.

We now only defined the "leafs", i.e. the hyperlinks and a title for each of the documents, but the define.xml still requires us to state explicitly which of these is the "annotated CRF" (define.xml is machine-readable and machines should not rely on naming conventions only), and which documents belong to the "Supplemental Documents".

In order to do so, click the tab "Annotated CRFs":

ľ	Annotated CRFs	Supplemental Documents	ľ

Currently, it is only allowed to provide a single PDF with all annotated CRFs in it (separate documents, e.g. one for each CRF, is not allowed).

Clicking the "+" icon leads to:

⊱ Extra i	nformation for: AnnotatedCRF	X
?	DocumentRef	
_	leafID	Define Leaf definitions:
		LF.aCRF
		LF.REFGUIDE

and one can now "drag-and-drop" a "leaf" from the right to the first cell on the left, with the result:

See Extra i	nformation for: AnnotatedCRF		×
	LeafID LF.aCRF	Define Leaf definitions: LF.aCRF LF.REFGUIDE	

Click OK to finalize the process of assigning an "annotated CRF" to the study submission.

The same can then be repeated for any other documents like the "reviewers guide", by using the tab "Supplemental Documents":

p. 106

Annotated CRFs	Supplemental Documents	ValueList Definitions	WhereClau
Extra i	nformation for: SupplementalDoc	Define Leaf definitions: LF.aCRF LF.REFGUIDE	×

Remark that you can add more than one supplemental document.

Assigning valuelists to SDTM/SEND/ADaM variables

ValueList are used for informing the reviewers about the metadata for individual groups of data. For example, for the Laboratory domain (LB), there can be hundreds of tests, and the information for each test can differ depending on the test code. For example, albumin measurements will have other units as glucose units.

Essentially, most findings domains have "hypervertical" structure, with a data model based on the "Entity - Attribute - Value" model (EAV model) This means that we have a single "entity" column (defining what the row "is about"). In the "Findings" domains, this is usually the --TESTCD column²⁵. There then are a number of attributes (like additional names and synonyms, but most importantly the --ORRESU (original result unit) and the --STRESU (standardized result unit). The "value" in EAV is delivered by the --ORRES column and by the --STRESC column and or the --STRESN column²⁶.

We already generated a valuelist "VL.CL.C66741.VSTESTCD" containing additional information (metadata) for 9 value-level vital signs variables. Currently, this valuelist is still "on its own", and we need to provide the information to which SDTM/SEND/ADaM variable it applies.

The define.xml 2.0 specification is very clear about this (section 4.4):

Value Level Metadata should be provided when there is a need to describe differing metadata attributes for subsets of cells within a column. It is most often used on SDTM Findings domains to provide definitions for Variables such as --ORRES, --ORRESU, --STRES, --STRESU that are specific to each test code (value of --TESTCD). It is not required for Findings domains where the results have the same characteristics in all records, such as IE domains. In ADaM, value level metadata often describes AVAL or AVALC in BDS data structures based on values of PARAMCD.

In our case, we provided information about what length of result can be expected, what datatype, and whether the result is coded, depending on the value of the VSTESTCD. So, we must assign the valuelist "VL.CL.C66741.VSTESTCD" to VSORRES.

If the value of the standardized result (VSSTRESC/VSSTRESN) is just copied from VSORRES, we can additionally also assign this valuelist to VSSTRESC and/or VSSTRESN.

²⁵A better choice would have been the --LOINC column, at least for LB and VS.

²⁶The reason that two columns are defined for the standardized result is due to the extremely old SAS-XPT format, in which columns can only either be "numeric" or "character".

In order to assign the valuelist "VL.CL.C66741.VSTESTCD" to VSORRES, select the tab "Variable Definitions" and look for "VSORRES" (one can use the "Show Search Panel"). One easily finds the row:

	5	Search for: VS	ORRES				Search
	S	earch within:	All Colur	nns			
		OID	✓ Name	🗌 DataTy	pe 📃 Length	Sig	gnificantDig
		SDSVarNan	ne 🗌 Origin	Comm	ent 📃 Display	/Format 📃 Co	mmentOID
OID	Name	DataType	Length	SignificantDi	SASFieldNa	SDSVarName	Origin
SVS.VSORRES	VSORRES	text	80				
VS.VSORRE	VSORRESU	text	11				
VS.VSSTRE	VSSTRESC	text	80				

Depending on how the variable definitions were created, you will still want to assign another "Length". The above picture shows a maximal length of "80" coming from the template SDTM. If you generated the variable definitions from a SAS-XPT file, the maximal length will probably already be correct, unless of course that you generated your SAS-XPT files with fields that are "too broad", something that the FDA doesn't like at all²⁷.

Suppose that the longest result for VSORRES was 20 characters. In that case we need to set the "Length" to 20, and also take care that this field is not longer than 20 bytes in the SAS-XPT file:

Γ		OID	Name	DataType	Length	SignificantDi
	<u></u>	VS.VSORRES	VSORRES	text	20	
	<u> </u>	VS.VSORRE	VSORRESU	text	11	
	<u> </u>	VS.VSSTRE	VSSTRESC	text	80	

Remark that max. Lengths have to be assigned to all variable definitions, except for when the data type is a date, time or datetime or an incomplete version of one of these three.

In order to assign the valuelist to VSORRES, click the "+" icon. This leads to:

information for: Iter	mDef, with OID =	VS.VSORRES		_	
RangeCheck	CodeList Refer	rence Role Alia:	s Origin Va	lueList Reference	
Descrip	Description Question Extern			nalQuestion	MeasurementUnitRef
	Language				Translated Text
en				Result or Finding in O	riginal Units

In this case, the "Label" was already added (in define.xml 2.0, it is added to the "Description" element). Be sure that the label for any domain variable is exactly identical (case sensitive!) to what is defined in the SDTM-IG, SEND-IG, or ADaM-IG.

Unfortuanetely, there is no room for any deviation at all here, even if a slightly different label would better explain what the variable is about. See the article

²⁷This is another relict of the SAS-XPT era. When using XML instead of SAS-XPT, there would be no problem at all.
"http://cdiscguru.blogspot.com/2015/12/sdtm-labels-freedom-or-slavery.html" for further details.

Getting the label 100% correct can especially be challenging in case of additional variables that come from the model rather than from the implementation guide (like additional timing variables) and when coming from the SAS-XPT files themselves. In future, we will use the "CDISC Library API" to check labels against the published ones from CDISC.

Another tab in this dialog is the "ValueList Reference" tab. When clicked, the following appears:



Until now, we only defined one valuelist, so there is only one entry at the right. In order to state that the valuelist "VL.CL.C66741.VSTESTCD" applies to the variable "VSORRES", just drag-and-drop it from the right to the left, leading to:

⊱ Extra i	nformation for: ItemDef, with OI	ID = VS.VSORRES	-	×
?	RangeCheck CodeList R	eference Role Alias	Origin ValueList Reference	MassuramontUnitDof
	ValueListOID	Quesuon	Value List	weasurementomicker
	VL.CL.C66741.VSTESTCD		definitions: VL.CL.C66741.VSTES	STCD

and you are done ...

Later, when inspecting the variable VSORRES in the browser, this will then appear as:

Value Level Metadata - VS [VSORRES]

Variable	Where	Туре	Length / Display Format	Controlled Terms or Format	Origin	Derivation/Comment
VSORRES	<u>VSTESTCD</u> = "BMI" (Body Mass Index) and =	float	5			Weight (kg) divided by height (m) squared
VSORRES	VSTESTCD = "DIABP" (Diastolic Blood Pressure) and =	integer	3			DIABP comment
VSORRES	<u>VSTESTCD</u> = "FRMSIZE" (Body Frame Size)	text	8	["LARGE", "MEDIUM", "SMALL"] < <u>Size</u> >		
VSORRES	<u>VSTESTCD</u> = "HEIGHT" (Height)	integer	3			
VSORRES	<u>VSTESTCD</u> = "HR" (Heart Rate)	integer	3			
VSORRES	VSTESTCD = "SYSBP" (Systolic Blood Pressure)	integer	3			
VSORRES	<u>VSTESTCD</u> = "TEMP" (Temperature)	float	5			
VSORRES	<u>VSTESTCD</u> = "WEIGHT" (Weight)	float	5			
VSORRES	<u>VSTESTCD</u> = "WSTCIR" (Waist Circumference)	integer	3			

Assigning an Origin to variables

Every data point in a submission has an origin. It is of utmost importance for traceability that this information is provided to the reviewer. It can be that the data point comes from the CRF, it can come from the protocol, or it can be assigned or derived (e.g. all --DY variables). It can also come from an electronic data transfer (e.g. some laboratory data that were not captured on a CRF), or (usually in case of ADaM), it can come from a "predecessor", e.g. when the data point was copied from SDTM into ADaM.

The origin can either be assigned on the "domain variable level" or on the "valuelist level". For example, "STUDYID" will always be identical and will probably come from the protocol. So we can assign it at the domain variable level.

On the other hand, "VSORRES" will need to be assigned on the "valuelist level", as there usually is no field "vital sign original result" on the CRF, but there are fields "systolic blood pressure", "systolic blood pressure", "height", "weight" and so on.

So it does not make sense (it even would probably be an error) to assign an Origin to VSORRES. However, as we assigned a valuelist to "VSORRES", this is already an indication that origins will be assigned on the valuelist level, i.e. for "DIABP", "SYSBP", "HEIGHT" etc..

So in our variable list, we will leave the value for "Origin" empty for VSORRES, but will assign origins for the valuelist variables "DIABP", "SYSBP", "HEIGHT" etc..

In order to assign an origin to e.g. the vital sign "BMI", look for it in the list of "Variable Definitions":

📑 🔍 QNAM	QNAM	text	80	
🛛 🖸 🔍 QLABEL	QLABEL	text	80	
QVAL	QVAL	text	80	
	QORIG	text	80	
QEVAL	QEVAL	text	80	
🖺 🔍 IT.BMI	BMI	float	5	

Then click the "+" icon to add additional information, and select the "Origin" tab:

Se Extra in	Extra information for: ItemDef, with OID = IT.BMI					
?	RangeCheck CodeList R	Reference Role Alias	Origin ValueList Reference	MeasurementIInitRef		
	Туре					

Although the define.xml XML-Schema allows you to add more than one origin, this is essentially not allowed. If you need to assign more than one "origin", this usually means that you did not well enough develop your valuelist.

For example, if you have have different glucose values, and one set was delivered electronically by lab 1, and the other set was captured on the CRF, you may want to have two different variables "IT.GLUC.LAB1" and "IT.GLUC.LAB2" so that you can assign "electronic data transfer" to the former as being the origin, and "CRF" to the latter²⁸. In the "where clause" you will then need to differentiate e.g. by the "DM.SITE" or "LB.NAM" (laboratory name) variable.

But now back to our BMI valuelist variable. In this case the origin is "derived", as the value was calculated from the weight and height. So we assign "Derived" as the type, by clicking in the "Type" field which leads to the following dialog:

²⁸Unfortunately, the situation for lab values can be even much more complicated, as CDISC and the FDA still refuse to allow the LOINC code of the lab test as the unique identifier.

Extra i	RangeChe De Type C C C C C C C C C C C C C C C C C C C	r: ItemDef, with O	ID = IT.BMI Reference Role Question RF erived ssigned rotocol DT: electronic data redecessor wher r: OK	Alias		
---------	---	--------------------	--	-------	--	--

Remark that the option "Other" is essentially not allowed in the case of a regulatory submission. If one selects this option, the following warning is shown:

Origin Ty	pe		23]	
?	 CRF Derived Assigned Protocol eDT: electronic d Predecessor 	ata transfe	۶r		
	Other:		Remark that in only the values Protocol, eDT,	the CF and	e context of a regulatory submission, RF, Derived, Assigned, d Predecessor are allowed

In our case however, we set the "Origin Type" to derived, leading to:

S- Extra information for: ItemDef, with OID = IT.BMI							
?	RangeCheck	CodeList R	eference	Role	Alias	Origin	
	Descrip	tion	Qu	estion		E	
	Туре						
	Derived						

When the type is "Derived" we also need to add the derivation method. We already saw how that can be done when creating a "ValueList" from a "CodeList" and will revisit this topic later.

For the "diastolic blood pressure", the origin is the CRF. So, when drilling into the details for "DIABP" we get:

⊱ Extra	information for: ItemDef, with O	ID = IT.DIABP	
?	RangeCheck CodeList R Description	Reference Role Alias Question	Origin Valuel External
	Type		
	RQ.		

When then clicking the "Type" field, we select "CRF":

Extra in	formation Range	n for: ItemDef, with OID = IT.DIABP Check CodeList Reference Role Alias Origin 1
		Description Question Exterpe

leading to:

⊱ Extra in	Extra information for: ItemDef, with OID = IT.DIABP						
?	RangeCheck CodeList Reference Role Alias Ori Description Question	gin					

Clicking the "+" icon then allows to define where the field for "diastolic blood pressure" can be found in the annotated CRF:

⊱ Extra i	nformation for: Origin	
?	Description DocumentRef	
_	leafID	Define Leaf definitions:
	E.C.	LF.aCRF
		aCRF.pdf -
		Annotated CRF for the CES study.

We state that the information can be found in the document defined by the "leaf" "LF.aCRF" by doing a drag-and-drop from the right to the left:

⊱ Extra information for: Origin						
?	Description DocumentRef					
_	leafID C. C. LF. aCRF	Define Leaf definitions: LF.aCRF				

and the further details about "where" in the annotated CRF by clicking the "+" icon:

⊱ Extra ir	formation for: DocumentRef, with leafID = LF.aCRF
?	PDFPageRef
_	Page Details for External Document Reference with leafID LF.aCRF
	O No page details
	Page list (physical reference)
	O Named destinations
	Page list / List of named destinations
	Page ran List of PDF named destination or pages numbers separated by a space First page: 1 Do NOT use a comma as a separator Only use a single space.
	Last page: 22
	Validate
	OK Cancel

We can either provide no page details at all, or a page list (one or more page numbers separated by a space), or a page range:

🔾 Named	destinations	
Page list / L	ist of named destinations	
Page ra	inge: first page - last page	
First page:	11	
Last page:	15	

The "Validate" button helps us avoiding making syntactic errors (like using comma-separated page numbers).

Suppose that the diastolic blood pressure was collected in different forms, which appears as pages 11, 13 and 16 in the "annotated CRF PDF". The selection would then be:

⊱ Extra in	formation for: DocumentRef, with leafID = LF.aCRF
?	PDFPageRef
_	Page Details for External Document Reference with leafID LF.aCRF
	O No page details
	Page list (physical reference)
	O Named destinations
	Page list / List of named destinations
	Page range: first page - last page
	First page: 11
	Last page. 15
	Validate
	OK Cancel

After clicking the OK button, the information is stored in the define.xml.

When viewing the define.xml in the browser, the result then is:

Value Level	Metadata - VS	[VSORRES]

Variable	Where	Туре	Length / Display Format	Controlled Terms or Format	Origin	Derivation/Comment
VSORRES	VSTESTCD = "BMI" (Body Mass Index) and =	float	5		Derived	Weight (kg) divided by height (m) squared
VSORRES	VSTESTCD = "DIABP" (Diastolic Blood Pressure) and =	integer	3		CRF Pages <u>11</u> <u>13 16</u>	DIABP comment
VSORRES	<u>VSTESTCD</u> = "FRMSIZE" (Body Frame Size)	text	8	["LARGE", "MEDIUM", "SMALL"] < <u>Size</u> >		

When the user than e.g. clicks on "11" in "CRF Pages", then the browser should automatically open the "annotated CRF PDF" on page 11²⁹.

Like this, one should essentially assign an "Origin" to <u>every</u> variable in the define.xml. If the variable is a domain variable to which a valuelist has been assigned, one will usually assign the "Origin" to the value-level variables rather than on the parent domain level variable (as was the case for VSORRES).

For developers of define.xml, it is not always clear what "Origin" should be assigned to variables that do not represent data from a CRF (e.g. "STUDYID", "DOMAIN", "--CAT", "--DY"). Therefore, a list is provided in the appendix with typical "Origin" assignments for SDTM variables. Please remark that these assignments are suggestions only, and not binding at all.

²⁹Remark that this functionality is (or must be) provided by the stylesheet. It is not programmed in the define.xml itself.

Retrieving PDF Page Numbers from the annotated CRF

Getting the PDF page numbers from the annotated CRF into the define.xml can be a tedious task that is error prone. Therefore we developed a feature to extract annotations from the annotated CRF and compare them with the variable name definitions (ItemDefs) and the "WhereClause" definitions. This is not a "black box" feature as in some other software packages: before doing the PDF page assignments, the user can still inspect the suggestions, and select the ones to be implemented.

Of course, you will first need to add the location of the annotated CRF to the define.xml using the menus "Add - Document Link" and "Add - Annotated CRF".

Once this link is present, use the menu "Extra - Insert CRF Page Numbers from annotated CRF":

View	Extra	Options Help
	Insert	CRF Page Numbers from annotated CRF
	Gener	te ValueList from CodeList
	Gener	te CodeList Subset starting from SAS-XPT and propose them for assigning Variable Origin CRF page numbers
ne /	Gener	te CodeListItem-CodeList from EnumeratedItem-Codelist

The system will then look up the location of the annotated CRF you already provided and start retrieving all annotations:



It will then analyze them, comparing them with all define.xml variable names that have been defined, but also with all "WhereClause" definitions.

🋓 Retrievi	ng and analyzing 📼 💷 🕺
Analyzi	ng 210 PDF annotations
	26%

For each, a "similarity" is calculated: if it is 100, meaning that the annotation exactly matches the variable name from the define.xml or the "WhereClause" definition (as human-readable text). For the latter, similarities of over 60-70% usually mean a "hit".

When the analysis is ready (this can take 1-2 minutes in the case of a large study - but you probably only need to do this once), the following dialog is presented:



For each annotation that could be matched to a variable (or to a set of variables that have the same "WhereClause"), a checkbox will appear in the dialog.

When the Item is colored green, this means that the Origin was already assigned in the define.xml as of type "CRF" and the page number(s) or the page range³⁰ exactly match with the page numbers retrieved from the annotated CRF. In this case you will probably not want to reassign the page number(s) in the define.xml.

In case the Item is colored red, this means that the "Origin" in the define.xml was not assigned to be "CRF", but something else. In the above example, the origin was assigned to be "Protocol", but a "STUDYID" annotation was found in the annotated CRF on page 6.

In case the Item is colored blue, this means that no origin was assigned yet in the define.xml. All items will be colored blue e.g. in case that you have not assigned any origins yet, and first want to automatically assign the ones that are of type "CRF" and are retrieved from the annotated CRF.

In case the Item is colored orange, this means that page numbers were retrieved from the annotated CRF, but these do not coincide with the ones that were already assigned. For example:

³⁰The algorithm takes into account that the define.xml def:Origin has a "FirstPage" and "LastPage" assigned. For example if in the define.xml one already FirstPage="6" and LastPage="10", and the page numbers from the annotated CRF are 6, 7, 8, 9, and 10, this is found to be a perfect match.



For QSTEST, an 100% matching annotation was found in the annotated CRF on pages 13, 14, 15 and 17, but in the define.xml, only page number 17 was assigned.

You can now check the checkboxes for the items for which you want to auto-generate the "def:Origin" page numbers in the define.xml, based on <u>your</u> considerations of what is correct and meaningful. For example:

Item: VISIT [IT.PE.VISIT] pages = (4,5) Annotation = VISIT - similarity = 100%
Item: PEDTC [IT.PE.PEDTC] pages = 10 Annotation = PEDTC - similarity = 100%
Item: QSTEST [IT.QS.QSTEST] pages = (13,14,15,17) Annotation = QSTEST - similarity = 100%
Item: QSCAT [IT.QS.QSCAT] pages = (13,14,15,17) Annotation = QSCAT - similarity = 100%
Item: QSORRES [IT.QS.QSORRES] pages = (13,14,15,17) Annotation = QSORRES - similarity = 100%
Item: VISIT [IT.QS.VISIT] pages = (4,5) Annotation = VISIT - similarity = 100%

The "green" ones were already correctly assigned, so you don't want to change that anymore. For PE.VISIT and QS.VISIT you decide that the earlier assignment "derived" was correct, this although there as such annotations on pages 4 and 5. So you leave these unchecked too. Just for QSTEST, QSCAT and QSORRES, you decide to implement the automated generation of a "def:Origin" in the define.xml.

After clicking OK, the assignments are implemented. If one then does an "Origin" inspection for e.g. QSORRES, one will find:

🔾 No page	details
Page list	t (physical reference)
O Named o	destinations
Page list / Li	ist of named destinations
13 14 15 17	
Page rai	nge: first page - last page
First page:	
Last page:	

with these page numbers extracted from the annoted CRF.

The same also works for "ValueList" variables, for example:

Item: CSDD13 [IT.QS.QSORRES.CSDD13] pages = 15 Annotation = QSORRES when QSTESTCD = CSDD13 - similarity = 77%
Item: RANDOM [IT.TS.TSVAL.RANDOM] pages = 16 Annotation = DSTERM / DSDECOD = RANDOMIZED - similarity = 31%

A "match" has been found for the CDSS13 "ValueList" variable with a high "similarity", for which

in less than i monun)			1	
D. CYCLIC FUNCTIONS		OSORRES	when QSTESTCD = 0	SDD12
12. DIURNAL VARIATION OF MOOD Symptoms worse in the morning		0	1	2
13. DIFFICULTY FALLING ASLEEP Later than usual for this individual		QSORRES	when QSTESTCD = 1	CSDD13
14. MULTIPLE AWAKENINGS DURING SLEEP		QSORRES	when QSTESTCD = 0 1	2
15. EARLY MORNING AWAKENING Earlier than usual for this individual	S when CD =	0	1	2
CSDD15				

the PDF annotation is "QSORRES when QSTESTCD = CSDD13".

In this case, no CRF pages have been assigned yet in the define.xml (the item is colored blue), so we would like to have this done, we do check the checkbox and click "OK"

Page list (physic) Named destination Page list / List of national Page list / List of national	cal reference) tions		
Named destina Page list / List of na 15	tions		
Page list / List of na			
15	amed destinatio	ns	
13			
O Page range: first	st page - last pa	ige	
First page:			
Last page:			

The result upon inspection is:

<u>Conclusion</u>: this new feature in the ODM-Define-XML Designer allows you to extract page numbers from the annotated CRF semi-automatically, in such a way that the user still has full control of which page numbers are copied to the define.xml.

This feature will save you many hours of tedious and error prone work, for large studies maybe even days.