

# SDTM-ETL 4.5 User Manual and Tutorial

Author: Jozef Aerts, XML4Pharma

Last update: 2024-03-18

## Mapping Units of Measure

### Table of Contents

Table of Contents .....	1
Introduction.....	1
Copying ODM units of measure to SDTM/SEND define .....	1
Mapping ODM Units of Measurement to SDTM/SEND CDISC Controlled Terminology units .....	5
Standardized units and UCUM .....	11
Conclusions.....	16

### Introduction

Units of measure can be handled and mapped to SDTM/SEND units of measure in different ways.

Main reason is that ODM has several mechanisms to store and handle units of measure.

The classic one, but not used by many EDC systems, is to store the units in the ODM element

"BasicDefinitions", and then referencing these, as well in the ODM metadata as in the clinical data, using the element "MeasurementUnitRef".

The second one is to just define the unit of measure as a data point, using "ItemDef" in the metadata, and "ItemData" in the clinical data.

And then there is [UCUM](#) (Unified Code for Units of Measure), which is used in healthcare and a standard in the medical informatics world, but unfortunately (due to the "not-invented-here syndrome) not (yet) used by CDISC.

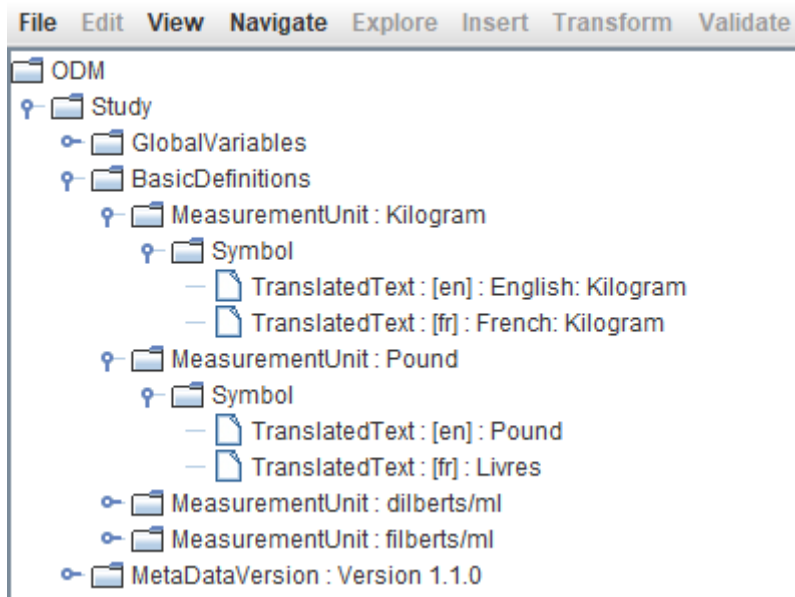
On the SDTM/SEND side, in many cases, variables representing units of measure (e.g. –ORRESU) is under CDISC controlled terminology , i.e. having a CDISC codelist. Better were that CDISC moves to UCUM, as unit conversions can easily be automated using UCUM, which is not possible at all with CDISC units.

In some cases, one will however also want its own codes for units of measure, also on the SDTM/SEND side.

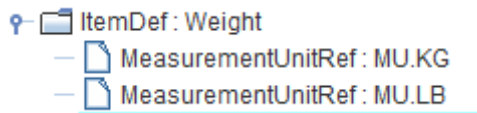
We will discuss each of these options in detail.

### Copying ODM units of measure to SDTM/SEND define

In ODM, in some implementations, the units of measure to be used are stored in the "BasicDefinitions" element. When this is so, one will find these in the ODM tree on the left side, e.g.:



For data points that use these units, one will find a reference to them e.g.



Essentially meaning that, in the (e)CRF, there was a choice between using "kg" or "pounds" as the unit of measure for "weight".

This also means that in the clinical data, one may find data points for which "kg" was used, and others for which "pounds" was used. For example, when using the menu "View - ODM Clinical Data":

View Clinical Data: Item OID: IT.WT - Name: Weight

File with ODM Clinical Data: D:\SDTM-ETLTestFiles\ODM1-3-1WMyStudy\_ODM\_1\_3\_1.xml

Generalize for all Items  
 Generalize for all ItemGroups  
 Generalize for all Forms  
 Generalize for all StudyEvents

Limit Results to first  Results  
 Also display RepeatKeys

ODM uses non-typed ItemData  ODM uses TYPED ItemData

Filter subjects

001  
002  
003  
004

Include selected  
 Exclude selected

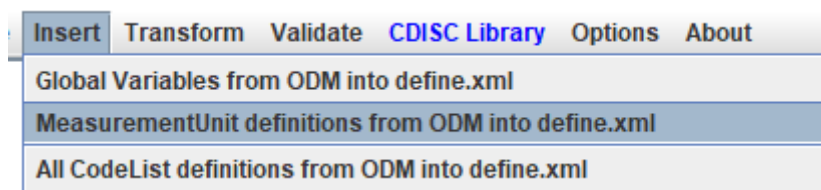
Subject	StudyEvent	Form	ItemGroup	Item	Name	Value	MeasurementUnit
001	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	204	Pound
002	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	77	Kilogram
003	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	122	Pound
004	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	185	Pound
005	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	244	Pound
006	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	175	Pound
007	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	168	Pound
008	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	97	Pound
009	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	171	Pound
010	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	80	Kilogram
011	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	114	Pound
012	SE.VISIT0	FORM.DEMOG	IG.DEMOG	IT.WT	Weight	193	Pound

Max.Length expected: 4 - Max.Length encountered: -1

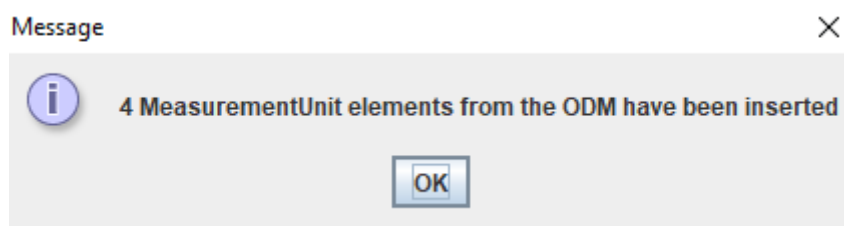
OK

One may now want to copy these units of measure from the ODM side to the SDTM/SEND define.xml and then transform this into a codelist. Very often, this will not be necessary, as one wants to use the already associated CDISC codelist. However, and especially in SEND, there will be no suitable associated codelist (e.g. for PK measurements) and one wants to use the units from the ODM.

To copy the units of measure from the ODM into the define.xml representing the SDTM/SEND data, use the menu "Insert - Measurement Unit definitions from ODM into define.xml":

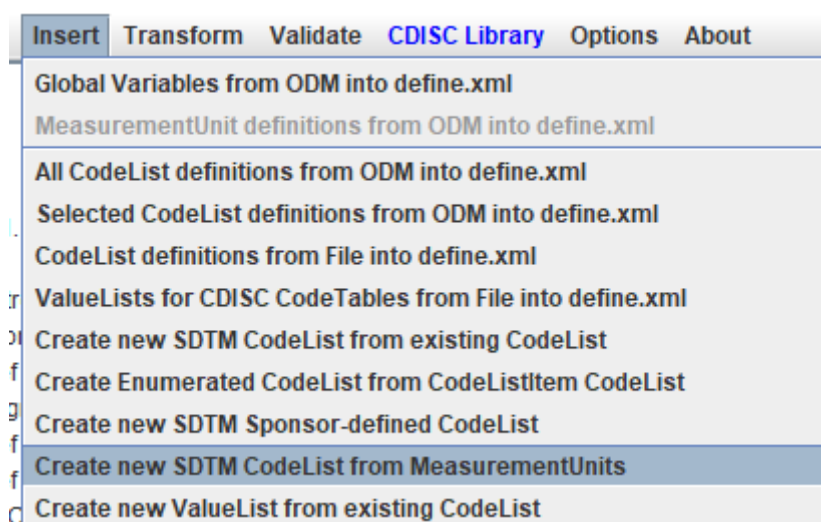


leading e.g. to:



As define.xml does not formally support the "BasicDefinitions" element, in most cases, one will want to transform these into an SDTM or SEND codelist.

This can be done by using the menu "Insert - Create new SDTM<sup>1</sup> CodeList from MeasurementUnits":



Which will pop up a new dialog:

---

<sup>1</sup> In the case SEND is used, the menu item will state "Create new SEND CodeList from MeasurementUnits"

Create new SDTM CodeList from set of MeasurementUnits

**?** New OID:

New Name:

New SASFormatName:

Use 'Name' for Code/Decode  Use 'Text' for Code/Decode

Remove rows until only those remain that you want to appear in the CodeList.  
Several rows can be removed simultaneously.

Remove row		
OID	Name	Text
MyStudy:MU.KG	Kilogram	English: Kilogram
MyStudy:MU.LB	Pound	Pound
MyStudy:MU.DPML	dilberts/ml	dilberts/ml
MyStudy:MU.FPML	filberts/ml	filberts/ml

OK Cancel

One then needs to at least fill the fields "New OID" and "New Name". For example:

Create new SDTM CodeList from set of MeasurementUnits

**?** New OID:

New Name:

New SASFormatName:

Use 'Name' for Code/Decode  Use 'Text' for Code/Decode

Remove rows until only those remain that you want to appear in the CodeList.  
Several rows can be removed simultaneously.

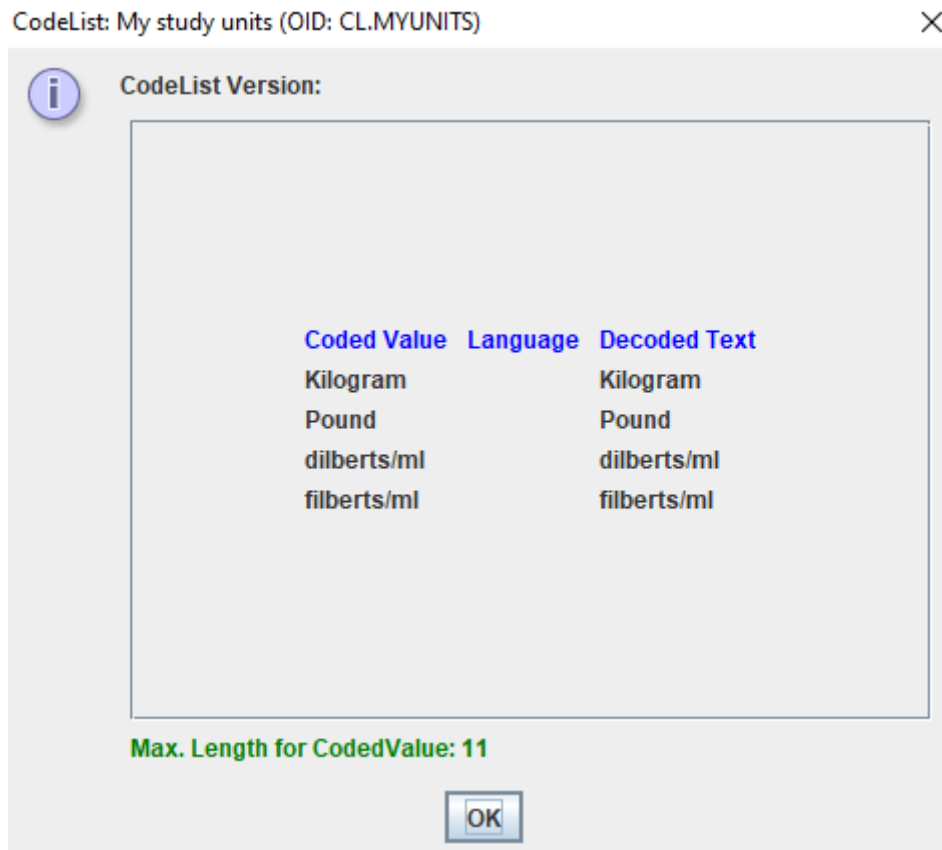
Remove row		
OID	Name	Text
MyStudy:MU.KG	Kilogram	English: Kilogram
MyStudy:MU.LB	Pound	Pound
MyStudy:MU.DPML	dilberts/ml	dilberts/ml
MyStudy:MU.FPML	filberts/ml	filberts/ml

One can still add or remove rows in this table, and has the option to either use the value of "Name" or of "Text" to appear in the "Decode" child elements of the codelist.

After clicking OK, one can use the generated codelist, e.g. assigning it to SDTM/SEND variables.

One can also still edit it, make subset codelists (menu "Insert - Create new SDTM CodeList from existing CodeList"), or use it as a basis for other codelists.

For visualization of the generated codelist, when one has assigned it to an SDTM/SEND variable, use the menu "View - SDTM Codelist" or "View - SEND CodeList", e.g. leading to:



CodeList: My study units (OID: CL.MYUNITS)

CodeList Version:

Coded Value	Language	Decoded Text
Kilogram		Kilogram
Pound		Pound
dilberts/ml		dilberts/ml
filberts/ml		filberts/ml

Max. Length for CodedValue: 11

OK

## Mapping ODM Units of Measurement to SDTM/SEND CDISC Controlled Terminology units

For EDC systems that support the "BasicDefinitions" mechanism for the units of measure in the ODM export, one can map these to the CDISC controlled terms used in SDTM or SEND directly.

Let us try to do this for VS, i.e. map the units from the ODM to the CDISC codelist VSRESU (NCI code C66770). According to the [CDISC Library](#):

cdisc LIBRARY Data Standards Browser

Controlled Terminology Package Effective 2023-12-15

Packages: DDF CT 2023-12-15, Define-XML CT 2023-12-15, Glossary CT 2023-12-15, Protocol CT 2023-12-15, **SDTM CT 2023-12-15**, SEND CT 2023-12-15

← C66770

**Extensible:** Yes  
**Submission Value:** VSRESU  
**Definition:** The unit used to record and describe the result of a test investigating a vital sign. (NCI)  
**NCI Preferred Term:** CDISC SDTM Unit for Vital Sign Result Terminology  
**Synonyms:** Units for Vital Signs Results

Filter results

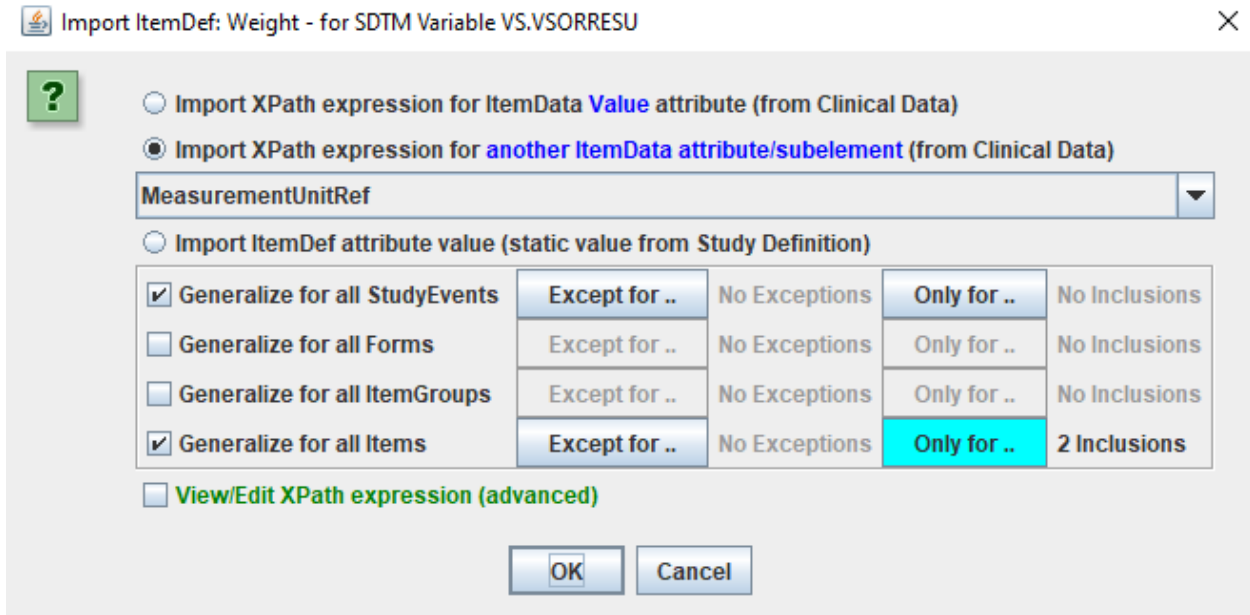
Term	Submission Value	Synonyms	Definition	NCI Preferred Term
C25613	%	Percentage	One hundred times the quotient of one quantity divided by another, with the same units of measurement.	Percentage
C49673	beats/min	Beats per Minute; BPM; bpm	The number of heartbeats measured per minute time. (NCI)	Beats per Minute
C49674	breaths/min	Breaths per Minute	The number of breaths (inhalation and exhalation) taken per minute time. (NCI)	Breaths per Minute
C42559	C	Degree Celsius	A unit of temperature of the temperature scale designed so that the freezing point of water is 0 degrees and the boiling point is 100 degrees at standard atmospheric pressure. The current official definition of the Celsius sets 0.01 C to be at the triple point of water and a degree to be	Degree Celsius

When we already have provided the mapping for VSTESTCD, VSTEST, VSORRES, we can also generate the mapping for VSORRESU. In order to do so, just drag-and-drop from "Weight" (OID="IT.HT") to the cell VS.VSORRESU:

The screenshot shows the CDISC Data Standards Browser interface. On the left is a tree view of the SDTM CT 2023-12-15 package. The 'ItemDef: Weight' node is highlighted in yellow. A red arrow originates from this node and points to the 'VS.VSORRESU' cell in the table on the right. The table displays a grid of code lists and their submission values. The 'VS.VSORRESU' cell is highlighted in blue, indicating it is the selected or active cell.

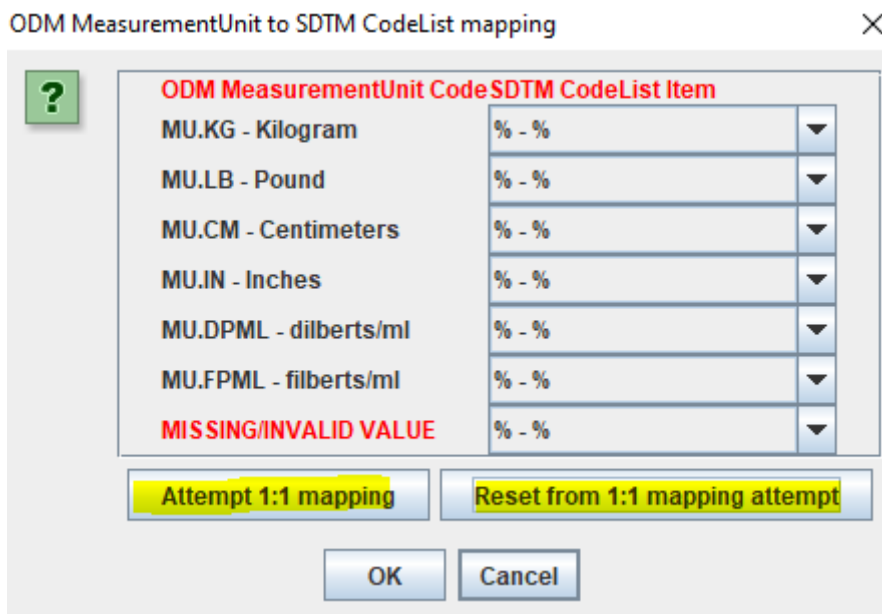
DAT	EG.EGSCAT	EG.EGPOS	EG.EGBEATH
RESC	IE.VISITNUM	IE.VISIT	IE.VISITDY
AT	IS.ISORRES	IS.ISORRESU	IS.ISSTRESC
CAT	LB.LBORRES	LB.LBORRESU	LB.LBORNRL
ESTCD	MB.MBTEST	MB.MBTSTDTL	MB.MBCAT
EST	MS.MSTSTDTL	MS.MSAGENT	MS.MSMODIF
IT	MI.MISCAT	MI.MIORRES	MI.MIORRESI
CAT	MO.MOSCAT	MO.MOPOS	MO.MOORRE
EST	CV.CVCAT	CV.CVSCAT	CV.CVPOS
EST	MK.MKCAT	MK.MKSCAT	MK.MKPOS
ESTCD	NV.NVTEST	NV.NVCAT	NV.NVSCAT
STDTL	OE.OECAT	OE.OESCAT	OE.OEORRE
EST	RP.RPCAT	RP.RPSCAT	RP.RPORRE
ESTCD	RE.RETEST	RE.RECAT	RE.RESCAT
TEST	UR.URSTDTL	UR.URCAT	UR.URSCAT
SCAT	PC.PCORRES	PC.PCORRESU	PC.PCSTRES
ORRESU	PP.PPSTRES	PP.PPSTRESN	PP.PPSTRESI
SCAT	PE.PEBODSYS	PE.PEORRES	PE.PEORRESI
CAT	FT.FTPOS	FT.FTORRES	FT.FTORRESI
ORRES	QS.QSORRESU	QS.QSSTRESC	QS.QSSTRESI
EST	RS.RSCAT	RS.RSSCAT	RS.RSORRE
ORRES	SC.SCORRESU	SC.SCSTRESC	SC.SCSTRESI
ORRES	SS.SSSTRESC	SS.SSSTAT	SS.SSREASN
EST	TU.TUORRES	TU.TUSTRESC	TU.TUNAM
EST	TR.TRORRES	TR.TRORRESU	TR.TRSTRES
OS	VS.VSORRES	VS.VSORRESU	VS.VSSTRES
CAT	FA.FAORRES	FA.FAORRESU	FA.FASTRESI
DAT	SR.SRSCAT	SR.SRORRES	SR.SRORRE
CDVER			
OS	VS.VSORRES	VS.VSORRESU	VS.VSSTRES

The "mapping wizard" dialog shows up:

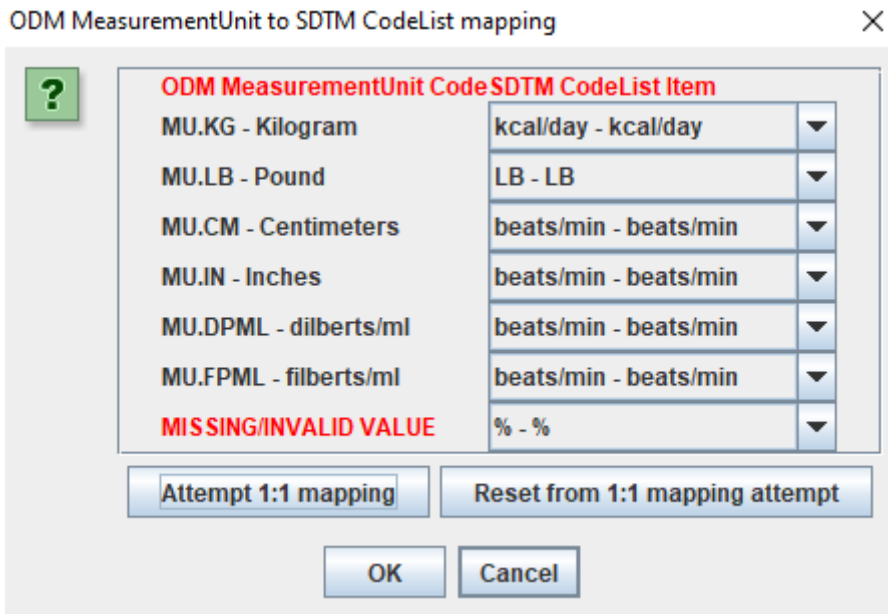


As the wizard sees that we have a "MeasurementUnitRef" child element on the ODM side, and we wish to map to an –ORRESU variable, it already suggests that we use the value of "MeasurementUnitRef". The "Only for ..." on "Generalize for all Items" is retained from our prior mappings for VSTESTCD, VSTEST and VSORRES, as we want to limit to "Height" and "Weight" ("Gender" in the same group does not go into VS).

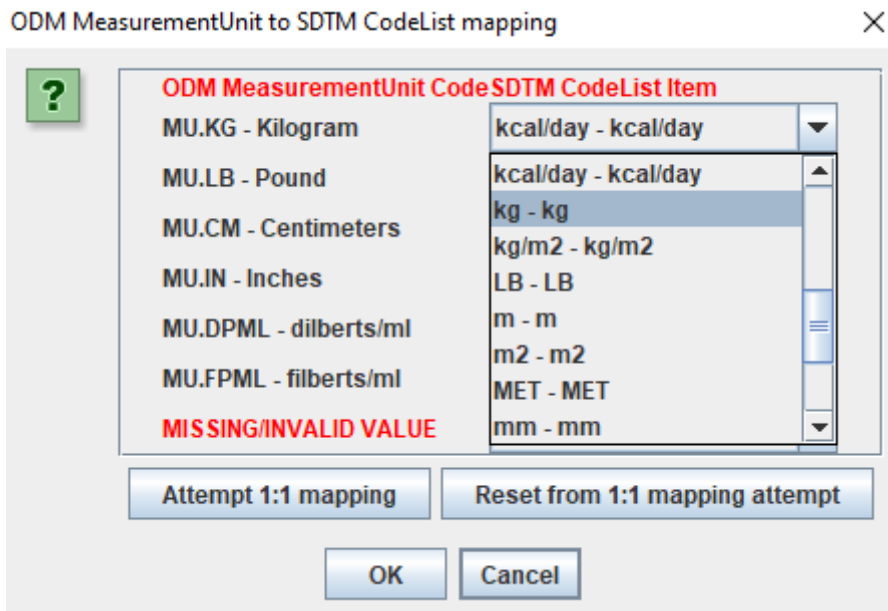
After clicking "OK" we get the part that allows us to define the mappings:



New in version 4.5 are the buttons "Attempt 1:1 mapping" and "Reset from 1:1 mapping attempt". When "Attempt 1:1 mapping" is clicked, the system will try to generate a mapping based world similarity, e.g. with the result:



with the "best guessed" values ordered within each dropdown, e.g.:



In our case, the "best guessed" mapping is not very good, so we may still need to do some "manual" mapping, e.g.:



## ODM MeasurementUnit to SDTM CodeList mapping



?

**ODM MeasurementUnit Code SDTM CodeList Item**

MU.KG - Kilogram	kg - kg	▼
MU.LB - Pound	LB - LB	▼
MU.CM - Centimeters	cm - cm	▼
MU.IN - Inches	in - in	▼
MU.DPML - dilberts/ml		▼
MU.FPML - filberts/ml		▼
<b>MIS SING/INVALID VALUE</b>		▼

Attempt 1:1 mapping

Reset from 1:1 mapping attempt

OK

Cancel

where we leave mappings for "dilberts/ml" and "filberts/ml" out as they will not be used for the vital signs units mappings anyway.

Leading to the automatically created mapping script:

```

The Transformation Script
1 # Mapping using ODM element ItemData with ItemOID IT.HT - value from attribute MeasurementUnitOID of subelement MeasurementUnitRef
2 # Generalized for all Items within the ItemGroup
3 # Mapped to SDTM CodeList: CL.C66770.VSRESU
4 $CODEVALUE = xpath(/StudyEventData[@StudyEventOID='SE.VISIT0']/FormData[@FormOID='FORM.DEMO0']/ItemGroupData[@ItemGroupOID='IG.DEMO0']/ItemData[@ItemOID='IT.HT' or @ItemOID='IT.WT']/MeasurementUnitRef/@MeasurementUnitOID)
5 $NEWCODEVALUE = ''
6 if ($CODEVALUE == 'MU.KG') {
7   $NEWCODEVALUE = 'kg';
8 } elseif ($CODEVALUE == 'MU.LB') {
9   $NEWCODEVALUE = 'LB';
10 } elseif ($CODEVALUE == 'MU.CM') {
11   $NEWCODEVALUE = 'cm';
12 } elseif ($CODEVALUE == 'MU.IN') {
13   $NEWCODEVALUE = 'in';
14 } elseif ($CODEVALUE == 'MU.DPML') {
15   $NEWCODEVALUE = '';
16 } elseif ($CODEVALUE == 'MU.FPML') {
17   $NEWCODEVALUE = '';
18 } else {
19   $NEWCODEVALUE = '';
20 }
21 $VS.VSORRESU = $NEWCODEVALUE;
22
23
  
```

Notice the `"/MeasurementUnitRef/@MeasurementUnitOID"` in the XPath selection in line 4. The next lines then transform the ODM unit into the SDTM unit from the CDISC VSRESU codelist.

When then executing the mappings (using "Transform - Generate Transformation (XSLT) Code for ...", the final result is e.g.:

MyStudy:VS						
USUBJID	VS.VSSEQ	VS.VSTESTCD	VS.VSTEST	VS.VSORRES	VS.VSORRESU	
001		1	HEIGHT	Height	73	
001		2	WEIGHT	Weight	204	LB
002		1	HEIGHT	Height	164	
002		2	WEIGHT	Weight	77	kg
003		1	HEIGHT	Height	65	
003		2	WEIGHT	Weight	122	LB
004		1	HEIGHT	Height	69	
004		2	WEIGHT	Weight	185	LB
005		1	HEIGHT	Height	71	
005		2	WEIGHT	Weight	244	LB
006		1	HEIGHT	Height	71	
006		2	WEIGHT	Weight	175	LB
007		1	HEIGHT	Height	72	
007		2	WEIGHT	Weight	168	LB
008		1	HEIGHT	Height	62	
008		2	WEIGHT	Weight	97	LB
009		1	HEIGHT	Height	66	
009		2	WEIGHT	Weight	171	LB
010		1	HEIGHT	Height	69	
010		2	WEIGHT	Weight	80	kg
011		1	HEIGHT	Height	61	
011		2	WEIGHT	Weight	114	LB
012		1	HEIGHT	Height	66	

Number of records: 24  
Number of subjects: 12  
Number of distinct tests: 2

Where we see that for most subjects, weight was collected in pound (CDISC-code "LB") but for a few, weight was collected in kilograms (CDISC-code "kg").

We also see that no unit was assigned for "HEIGHT". Reason for this that there wasn't a choice for the unit for "height", but was always collected in inches (CDISC-code "in"). So we need to extend the code somewhat, and have the assignment being based on the value of VSTESTCD. For example:

```

The Transformation Script
1 # Mapping using ODM element ItemData with ItemOID IT.WT - value from attribute MeasurementUnitOID of subelement
2 # Generalized for all StudyEvents
3 # Generalized for all Items within the ItemGroup
4 # Mapped to SDIM CodeList CL.C66770.VSRESU
5 $CODEDVALUE = xpath(/StudyEventData/FormData[@FormOID='FORM.DEMOG']/ItemGroupData[@ItemGroupOID='IG.DEMOG']/Ite
6 $NEWCODEDVALUE = '';
7 if ($CODEDVALUE == 'MU.KG') {
8   $NEWCODEDVALUE = 'kg';
9 } elseif ($CODEDVALUE == 'MU.LB') {
10  $NEWCODEDVALUE = 'LB';
11 } elseif ($CODEDVALUE == 'MU.DPML') {
12  $NEWCODEDVALUE = '';
13 } elseif ($CODEDVALUE == 'MU.FPML') {
14  $NEWCODEDVALUE = '';
15 } else {
16  $NEWCODEDVALUE = '';
17 }
18 if($VS.VSTESTCD = 'WEIGHT') {
19   $VS.VSORRESU = $NEWCODEDVALUE;
20 } elseif($VS.VSTESTCD = 'HEIGHT') {
21   $VS.VSORRESU = 'in';
22 } else {
23   $VS.VSORRESU = 'TODO';
24 }

```

Essentially hardcoding that for the case of "HEIGHT", always "inches" (CDISC-code "in") needs to be assigned. This then ultimately leads to the results:

MyStudy:VS						
USUBJID	VS.VSSEQ	VS.VSTESTCD	VS.VSTEST	VS.VSORRES	VS.VSORRESU	
001		1 HEIGHT	Height	73	in	
001		2 WEIGHT	Weight	204	LB	
002		1 HEIGHT	Height	164	in	
002		2 WEIGHT	Weight	77	kg	
003		1 HEIGHT	Height	65	in	
003		2 WEIGHT	Weight	122	LB	
004		1 HEIGHT	Height	69	in	
004		2 WEIGHT	Weight	185	LB	
005		1 HEIGHT	Height	71	in	
005		2 WEIGHT	Weight	244	LB	
006		1 HEIGHT	Height	71	in	
006		2 WEIGHT	Weight	175	LB	
007		1 HEIGHT	Height	72	in	
007		2 WEIGHT	Weight	168	LB	
008		1 HEIGHT	Height	62	in	
008		2 WEIGHT	Weight	97	LB	
009		1 HEIGHT	Height	66	in	
009		2 WEIGHT	Weight	171	LB	
010		1 HEIGHT	Height	69	in	
010		2 WEIGHT	Weight	80	kg	
011		1 HEIGHT	Height	61	in	
011		2 WEIGHT	Weight	114	LB	
012		1 HEIGHT	Height	66	in	

Number of records: 24  
Number of subjects: 12  
Number of distinct tests: 2

## Standardized units and UCUM

A big problem with the CDISC controlled terminology for units is that it does not allow to automate unit conversion, which is often needed for populating –STRESN variables. For example for our case, in case we want to have "cm" and "kg" for the "standardized" results, there is no other way than to hardcode the conversion factors. This means a lot of manual work. For example, for VSSTRESN, we will need a (too) long manually generated script like:

```

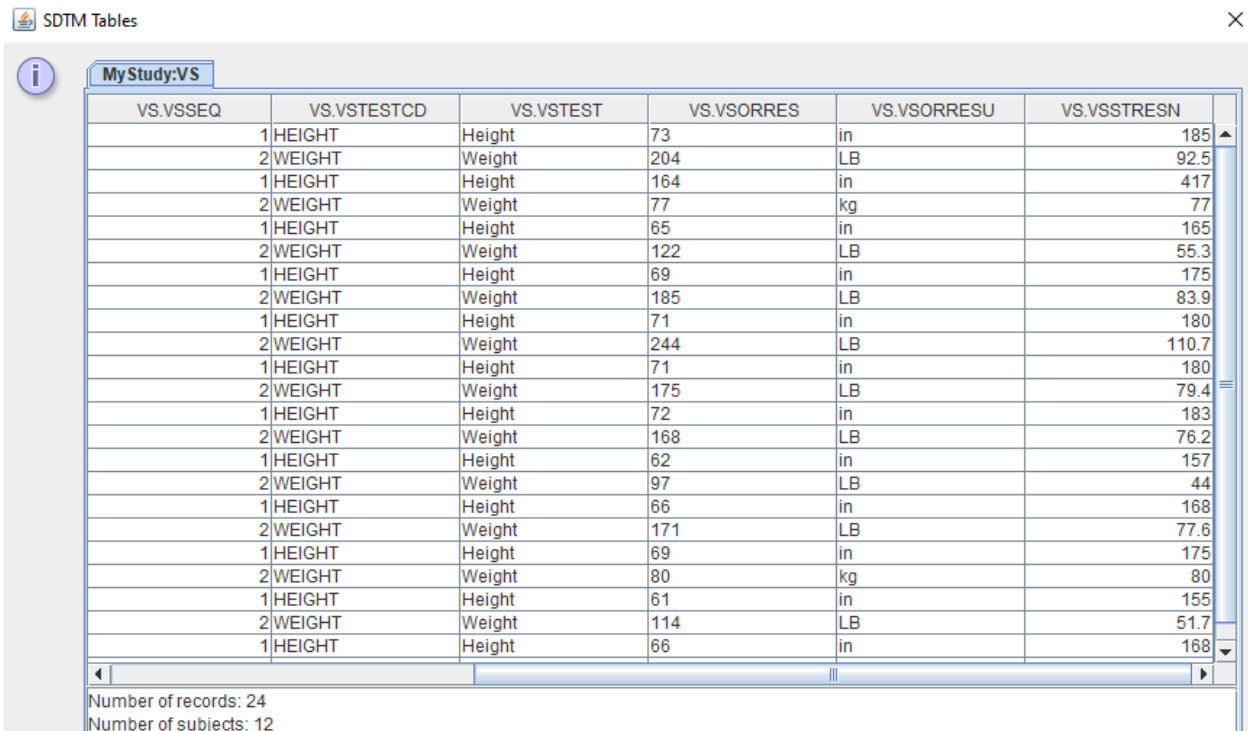
The Transformation Script
1 # Conversion to standardized units kg (for weight) and cm (for height)
2 if($VS.VSTESTCD = 'HEIGHT' and $VS.VSORRESU = 'in') {
3   # convert to centimeters
4   $TEMP = number($VS.VSORRES) * 2.54;
5   $VS.VSSTRESN = round($TEMP,0);
6 } elseif($VS.VSTESTCD = 'WEIGHT' and $VS.VSORRESU = 'LB') {
7   # convert to kilograms
8   $TEMP = number($VS.VSORRES) * 0.4536;
9   $VS.VSSTRESN = round($TEMP,1);
10 } elseif($VS.VSTESTCD = 'WEIGHT' and $VS.VSORRESU = 'kg') {
11   $VS.VSSTRESN = $VS.VSORRES;
12 } else {
13   $VS.VSSTRESN = 'TODO';
14 }

```

where we use rounding to an integer for conversion to centimeters and to one character after the decimal point for conversion to kilograms.

Also remark the use of the "number()" function, as in our mapping language, variable values are essentially always strings, and need to be converted to a number for mathematical operations.

The total result then is:



VS.VSSEQ	VS.VSTESTCD	VS.VSTEST	VS.VSORRES	VS.VSORRESU	VS.VSSTRESN
1	HEIGHT	Height	73	in	185
2	WEIGHT	Weight	204	LB	92.5
1	HEIGHT	Height	164	in	417
2	WEIGHT	Weight	77	kg	77
1	HEIGHT	Height	65	in	165
2	WEIGHT	Weight	122	LB	55.3
1	HEIGHT	Height	69	in	175
2	WEIGHT	Weight	185	LB	83.9
1	HEIGHT	Height	71	in	180
2	WEIGHT	Weight	244	LB	110.7
1	HEIGHT	Height	71	in	180
2	WEIGHT	Weight	175	LB	79.4
1	HEIGHT	Height	72	in	183
2	WEIGHT	Weight	168	LB	76.2
1	HEIGHT	Height	62	in	157
2	WEIGHT	Weight	97	LB	44
1	HEIGHT	Height	66	in	168
2	WEIGHT	Weight	171	LB	77.6
1	HEIGHT	Height	69	in	175
2	WEIGHT	Weight	80	kg	80
1	HEIGHT	Height	61	in	155
2	WEIGHT	Weight	114	LB	51.7
1	HEIGHT	Height	66	in	168

Number of records: 24  
Number of subjects: 12

but we then still need to add the mapping for VSSTRESU too:

```
The Transformation Script
1 if($VS.VSTESTCD = 'HEIGHT') {
2     $VS.VSSTRESU = 'cm';
3 } elseif($VS.VSTESTCD = 'WEIGHT') {
4     $VS.VSSTRESU = 'kg';
5 } else {
6     $VS.VSSTRESU = 'TODO';
7 }
```

and the total result being:

MyStudy:VS									
STUDYID	DOMAIN	USUBJID	VS.VSSEQ	VS.VSTESTCD	VS.VSTEST	VS.VSORRES	VS.VSORRESU	VS.VSSTRESN	VS.VSSTRESU
MyStudy	VS	001	1	HEIGHT	Height	73	in	185	cm
MyStudy	VS	001	2	WEIGHT	Weight	204	LB	92.5	kg
MyStudy	VS	002	1	HEIGHT	Height	164	in	417	cm
MyStudy	VS	002	2	WEIGHT	Weight	77	kg	77	kg
MyStudy	VS	003	1	HEIGHT	Height	65	in	165	cm
MyStudy	VS	003	2	WEIGHT	Weight	122	LB	55.3	kg
MyStudy	VS	004	1	HEIGHT	Height	69	in	175	cm
MyStudy	VS	004	2	WEIGHT	Weight	185	LB	83.9	kg
MyStudy	VS	005	1	HEIGHT	Height	71	in	180	cm
MyStudy	VS	005	2	WEIGHT	Weight	244	LB	110.7	kg
MyStudy	VS	006	1	HEIGHT	Height	71	in	180	cm
MyStudy	VS	006	2	WEIGHT	Weight	175	LB	79.4	kg
MyStudy	VS	007	1	HEIGHT	Height	72	in	183	cm
MyStudy	VS	007	2	WEIGHT	Weight	168	LB	76.2	kg
MyStudy	VS	008	1	HEIGHT	Height	62	in	157	cm
MyStudy	VS	008	2	WEIGHT	Weight	97	LB	44	kg
MyStudy	VS	009	1	HEIGHT	Height	66	in	168	cm
MyStudy	VS	009	2	WEIGHT	Weight	171	LB	77.6	kg
MyStudy	VS	010	1	HEIGHT	Height	69	in	175	cm
MyStudy	VS	010	2	WEIGHT	Weight	80	kg	80	kg
MyStudy	VS	011	1	HEIGHT	Height	61	in	155	cm
MyStudy	VS	011	2	WEIGHT	Weight	114	LB	51.7	kg
MyStudy	VS	012	1	HEIGHT	Height	66	in	168	cm

Number of records: 24  
Number of subjects: 12  
Number of distinct tests: 2


So, a lot of manual code, and that only for two vital signs tests!

The whole medical world is using [UCUM](#) notation for units, except for CDISC (reason: "not-invented-here syndrome"). UCUM is not a "list" (CDISC-CT for units just is a list) and does, in contradiction to CDISC, allow for automated unit conversions.

There is even a free RESTful web service for it provided by the National Library of Medicine (NLM) for which the description can be found at <https://ucum.nlm.nih.gov/ucum-service.html>:

← → ↻ <https://ucum.nlm.nih.gov/ucum-service.html> 110% ☆

**NIH** National Library of Medicine  
Lister Hill National Center for Biomedical Communications

 **Unified Code for Units of Measure (UCUM)**

UCUM Resources from the Lister Hill National Center for Biomedical Communications

## UCUM Web Service

This is a set of web services (APIs) for programs to use when working with units from the Unified Code for Units of Measure ([UCUM](#)) system. These are the same APIs as those that are running at [xml4pharmaserver.com](http://xml4pharmaserver.com), and are based on that website's web service code which has been donated to the U.S. National Library of Medicine by FH-Prof. Jozef Aerts and Mr. Milos Ilic MSc, Institute of eHealth, University of Applied Sciences FH Joanneum in Graz Austria.

Currently, three web services are available:

- [UCUM unit Conversion web service](#)
- [UCUM unit validation web service](#)
- [UCUM unit to base units conversion](#)

Such RESTful web services can be used in any software, even in SAS or R.

Just suppose that our collection system already uses UCUM for units, which will usually (>99%) the case when the data comes from electronic health records<sup>2</sup>.

The UCUM notation for the above units is:

Designation	UCUM notation
centimeters	cm
inches	[in i]
kilograms	kg
pounds	[lb av]

Where one sees that for some cases, the UCUM notation is equal to the CDISC controlled term ("cm" and "kg") whereas it is very different for others.

A more complete oversight of "Commonly Used UCUM Codes for Healthcare Units" can be found at: <https://download.hl7.de/documents/ucum/ucumdata.html>

**Commonly Used UCUM Codes for Healthcare Units**

**Document Instructions**

This document contains a list of UCUM codes for commonly used units in clinical lab value reporting. The tables below display the valid UCUM code, descriptive n general type. You may browse the list, use the links to the left to jump to a specific unit type, or use the search box to the left to search for a particular unit.

**Additional Resources**

To save a copy of these tables to your computer, right click on the desired link and choose "Save Target As..." or "Save Link As..."

[Commonly Used UCUM Codes \(XML Format\)](#)  
[Commonly Used UCUM Codes \(TSV Format\)](#)  
[How To Download UCUM Codes from CDC PHIN VADS](#)

**Unit Types**

**Most Common Healthcare Units**

Valid UCUM Code	Descriptive Name
%	Percent
/uL	PerMicroLiter
[IU]/L	InternationalUnitsPerLiter
10 <sup>3</sup> /uL	ThousandsPerMicroLiter
10 <sup>6</sup> /uL	MillionsPerMicroLiter
fL	FemtoLiter
g/dL	GramsPerDeciLiter
g/L	GramsPerLiter
g/mL	GramsPerMilliLiter
kPa	KiloPascal
m[IU]/mL	MilliInternationalUnitsPerMilliLiter
meq/L	MilliEquivalentsPerLiter
ma/dL	MilliGramsPerDeciLiter

Our scripting language has a number of functions that use the NLM RESTful web service to have automated unit conversion using UCUM notation for the units.

These functions can be found when, in the mapping editor, clicking the button (located near the bottom) "RESTful WS", then popping up a list of available functions with tooltips with explanations:

---

<sup>2</sup> As well HL7 CDA as FHIR normally use UCUM notation for all units.

Origin: No Origin has been added yet!

The Transformation Script

```

1 if($VS.VSTESTCD = 'HEIGHT')
2   $VS.VSSTRESU = 'cm';
3 } elseif($VS.VSTESTCD = 'WEI
4   $VS.VSSTRESU = 'kg';
5 } else {
6   $VS.VSSTRESU = 'TODO'
7 }

```

RESTful Web Service Functions

sdmLabel

unitconversion

unitconversionLoinc

unitconver

loinc2lbtes

loinc2lbtes

loinc2lbsp

loinc2lbme

Performs a simple UCUM unit conversion using the NLM RESTful web service, described at: <https://ucum.nlm.nih.gov/ucum-service.html>. NOT for Conventional to SI units and vice versa.

Three parameters:

- quantity: the number quantity
- source UCUM unit
- target UCUM unit

OK

Scripting Language Functions

<	>	<=	>=	==	!=	
contains	starts-with	ends-with	matches	not		
abs	sqrt	log	log10	exp	exp10	
min	max	avg	sum	count	is-a-number	
ceiling	floor	round	modulus	number	string	
date	year	month-in-year	day-in-year	day-in-month	day-in-week	
time	hour-in-day	minute-in-hour	second-in-minute	createdatetime	datediff	
timediff	datetimediff	elementname	more date/time ...	RESTful WS	My Functions	

For example, although this is fully counterproductive in this case, we could use:

The Transformation Script

```

1 # Conversion to standardized units kg (for weight) and cm (for height)
2 if($VS.VSTESTCD = 'HEIGHT' and $VS.VSORRESU = 'in') {
3   # convert to centimeters
4   # $TEMP = number($VS.VSORRES) * 2.54;
5   $TEMP = rws:unitconversion($VS.VSORRES, '[in_i]', 'cm');
6   $VS.VSSTRESN = round($TEMP, 0);
7 } elseif($VS.VSTESTCD = 'WEIGHT' and $VS.VSORRESU = 'LB') {
8   # convert to kilograms
9   # $TEMP = number($VS.VSORRES) * 0.4536;
10  $TEMP = rws:unitconversion($VS.VSORRES, '[lb_av]', 'kg');
11  $VS.VSSTRESN = round($TEMP, 1);
12 } elseif($VS.VSTESTCD = 'WEIGHT' and $VS.VSORRESU = 'kg') {
13   $VS.VSSTRESN = $VS.VSORRES;
14 } else {
15   $VS.VSSTRESN = 'TODO';
16 }
17

```

Using the "rws:unitconversion" function.

Essentially, doing so only makes sense when the units come in UCUM notation already in the source data.

It however shows once again that CDISC should move to UCUM notation for units instead of mandating to use it's "own invented wheel".

## Conclusions

There are several possibilities for mapping units of measure from the source (ODM) to the target SDTM or SEND variables. When the units of measure are defined as datapoints (i.e. using "ItemDef"), the mapping is as for any other data point, with the difference that usually, a mapping to CDISC units will be needed, for which there is a wizard. Alternatively, the units defined in the ODM "BasicDefinitions" can be imported into the define.xml and transformed to a codelist. When the ODM uses the "MeasurementUnitRef" mechanism for units, when doing drag-and-drop from the "ItemDef" to an SDTM/SEND cell, the software will already suggest to use the value of "ItemData/MeasurementUnitRef/@MeasurementUnitOID" that can then be mapped to CDISC units using the provided wizard. SDTM-ETL also supports automatic unit conversion when the units are provided in UCUM notation. Due to the nature of the CDISC-CT (just lists) this is unfortunately not possible with CDISC units.