CDISC®

CLINICAL DATA INTERCHANGE STANDARDS CONSORTIUM

FH | JOANNEUM
University of Applied Sciences

The CDISC vision is to inform patient care & safety through higher quality medical research.

Strength *through Collaboration*

Define 2.0: What is new? How to switch from 1.0 to 2.0?

Presented by FH-Prof.Dr. Jozef Aerts
University of Applied Sciences FH Joanneum
Graz, Austria

Strength *through Collaboration*

What's new in define.xml 2.0?

- Based on ODM 1.3(.2) instead of on ODM 1.2
  - "Description" element replaces def:Label
  - "MethodDef" element replaces def:ComputationMethod
  - ItemRef/@KeySequence replaces def:DomainKeys
  - Machine-readable expressions in MethodDef
  - @Rank replaces CodeListItem/@def:Rank
  - EnumeratedItem for CDISC-CT
    - causing new problems …
  - def:Origin
  - reusable def:CommentDef
  - def:WhereClause

"Description" element replaces def:Label

Define.xml 1.0:

```
<ItemGroupDef OID="IG.DM"  Name="DM"  Repeating="No"
IsReferenceData="No"  Purpose="Tabulation"  def:Label="Demographics"
def:Structure="One record per event per subject" def:DomainKeys="STUDYID,
USUBJID" def:Class="Special Purpose" def:ArchiveLocationID="Location.DM">
```

Define.xml 2.0:

```
<ItemGroupDef OID="IG.DM"  Name="DM"  Repeating="No"
IsReferenceData="No"  Purpose="Tabulation"
def:Structure="One record per event per subject"
def:Class="Special Purpose" def:ArchiveLocationID="Location.DM">
        <Description>
                <TranslatedText xml:lang="en">Demographics</TranslatedText>
        </Description>
</ItemGroupDef>
```

"KeySequence" attributes replace def:DomainKeys

Define.xml 1.0:

```
<ItemGroupDef OID="IG.DM"  Name="DM"  Repeating="No" IsReferenceData="No"
Purpose="Tabulation"  def:Label="Demographics" def:Structure="One record per event per
subject"
def:DomainKeys="STUDYID, USUBJID" def:Class="Special Purpose"
def:ArchiveLocationID="Location.DM">
```

Define.xml 2.0:

```
<ItemGroupDef OID="IG.DM"  Name="DM"  Repeating="No" IsReferenceData="No"
Purpose="Tabulation"
def:Structure="One record per event per subject"
def:Class="Special Purpose" def:ArchiveLocationID="Location.DM">
        <ItemRef ItemOID="IT.STUDYID" OrderNumber="1" Mandatory="Yes"
                KeySequence="1"  />
        <ItemRef ItemOID="IT.DM.DOMAIN" OrderNumber="2"  Mandatory="Yes"/>
        <ItemRef ItemOID="IT.USUBJID" OrderNumber="3" Mandatory="Yes"
                KeySequence="2"  MethodOID="MT.USUBJID"/>
</ItemGroupDef>
```

"MethodDef" element replaces def:ComputationMethod

Define.xml 1.0:

```
<ItemDef OID="VS.VSTESTCD.MEANBP" Name="MEANBP" DataType="float" …
def:ComputationMethodOID="COMPMETHOD.MEANBP" >
</ItemDef>

<def:ComputationMethod OID="COMPMETHOD.MEANBP" …>
```

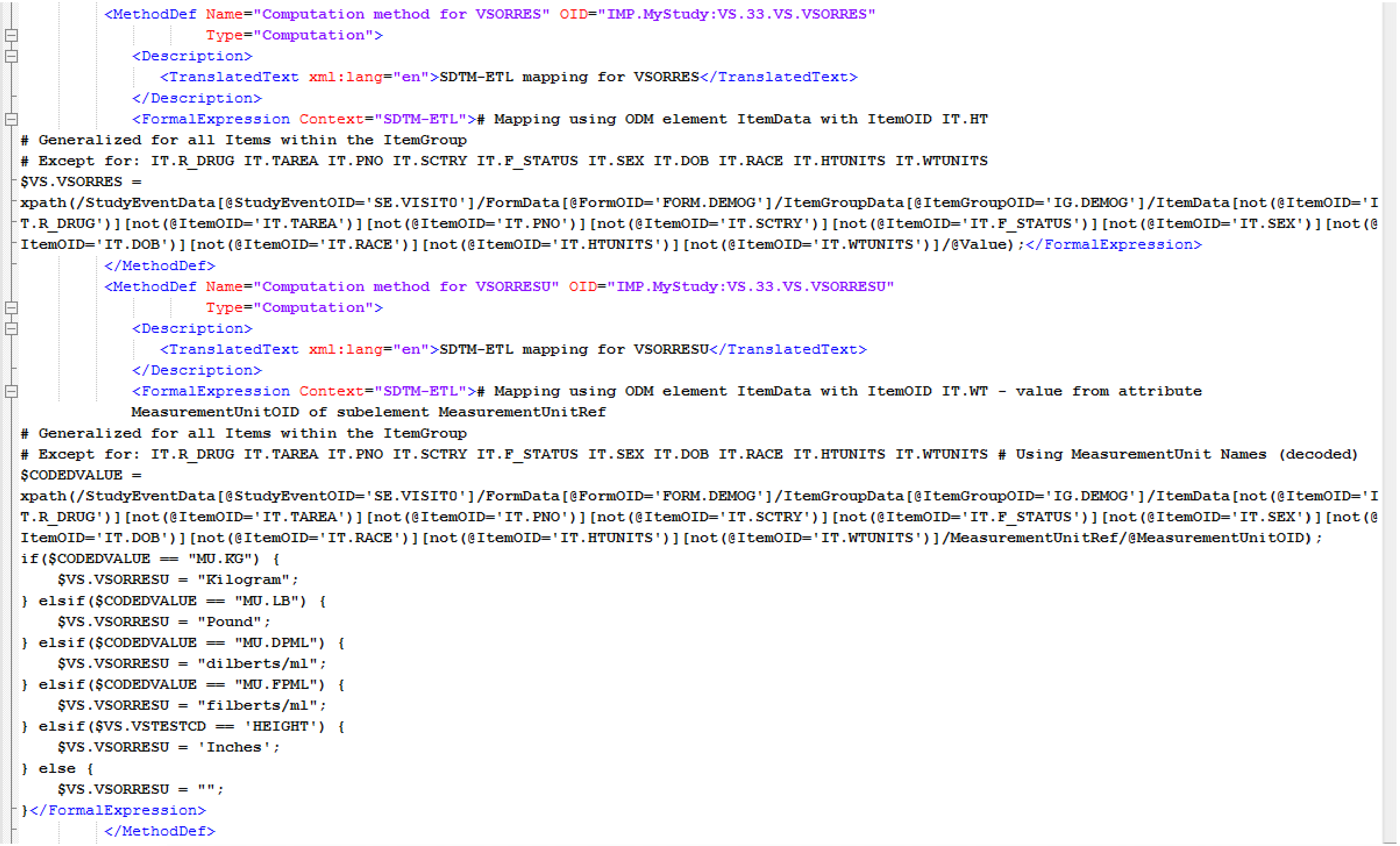
=> no way to add machine-readable algorithm, only text description

Define.xml 2.0:

```
<ItemDef OID="VS.VSTESTCD.MEANBP" Name="MEANBP" DataType="float" …
MethodOID="METHOD.MEANBP" >
</ItemDef>

<MethodDef OID="METHOD.MEANBP" Name="Mean BP" Type="Computation">
        <Description><TranslatedText …>…</TranslatedText></Description>
        <FormalExpression Context="MyScriptingLanguage">
        … </FormalExpression>
</MethodDef>
```

Machine-readable computation expressions example



```
         <MethodDef Name="Computation method for VSORRES" OID="IMP.MyStudy:VS.33.VS.VSORRES"
                    Type="Computation">
            <Description>
               <TranslatedText xml:lang="en">SDTM-ETL mapping for VSORRES</TranslatedText>
            </Description>
            <FormalExpression Context="SDTM-ETL"># Mapping using ODM element ItemData with ItemOID IT.HT
# Generalized for all Items within the ItemGroup
# Except for: IT.R_DRUG IT.TAREA IT.PNO IT.SCTRY IT.F_STATUS IT.SEX IT.DOB IT.RACE IT.HTUNITS IT.WTUNITS
$VS.VSORRES =
xpath(/StudyEventData[@StudyEventOID='SE.VISIT0']/FormData[@FormOID='FORM.DEMOG']/ItemGroupData[@ItemGroupOID='IG.DEMOG']/ItemData[not(@ItemOID='I
T.R_DRUG')][not(@ItemOID='IT.TAREA')][not(@ItemOID='IT.PNO')][not(@ItemOID='IT.SCTRY')][not(@ItemOID='IT.F_STATUS')][not(@ItemOID='IT.SEX')][not(@
ItemOID='IT.DOB')][not(@ItemOID='IT.RACE')][not(@ItemOID='IT.HTUNITS')][not(@ItemOID='IT.WTUNITS')]/@Value);</FormalExpression>
         </MethodDef>
         <MethodDef Name="Computation method for VSORRESU" OID="IMP.MyStudy:VS.33.VS.VSORRESU"
                    Type="Computation">
            <Description>
               <TranslatedText xml:lang="en">SDTM-ETL mapping for VSORRESU</TranslatedText>
            </Description>
            <FormalExpression Context="SDTM-ETL"># Mapping using ODM element ItemData with ItemOID IT.WT - value from attribute
            MeasurementUnitOID of subelement MeasurementUnitRef
# Generalized for all Items within the ItemGroup
# Except for: IT.R_DRUG IT.TAREA IT.PNO IT.SCTRY IT.F_STATUS IT.SEX IT.DOB IT.RACE IT.HTUNITS IT.WTUNITS # Using MeasurementUnit Names (decoded)
$CODEDVALUE =
xpath(/StudyEventData[@StudyEventOID='SE.VISIT0']/FormData[@FormOID='FORM.DEMOG']/ItemGroupData[@ItemGroupOID='IG.DEMOG']/ItemData[not(@ItemOID='I
T.R_DRUG')][not(@ItemOID='IT.TAREA')][not(@ItemOID='IT.PNO')][not(@ItemOID='IT.SCTRY')][not(@ItemOID='IT.F_STATUS')][not(@ItemOID='IT.SEX')][not(@
ItemOID='IT.DOB')][not(@ItemOID='IT.RACE')][not(@ItemOID='IT.HTUNITS')][not(@ItemOID='IT.WTUNITS')]/MeasurementUnitRef/@MeasurementUnitOID);
if($CODEDVALUE == "MU.KG") {
    $VS.VSORRESU = "Kilogram";
} elsif($CODEDVALUE == "MU.LB") {
    $VS.VSORRESU = "Pound";
} elsif($CODEDVALUE == "MU.DPML") {
    $VS.VSORRESU = "dilberts/ml";
} elsif($CODEDVALUE == "MU.FPML") {
    $VS.VSORRESU = "filberts/ml";
} elsif($VS.VSTESTCD == 'HEIGHT') {
    $VS.VSORRESU = 'Inches';
} else {
    $VS.VSORRESU = "";
}</FormalExpression>
         </MethodDef>
```

@Rank replaces CodeListItem/@def:Rank

Define.xml 1.0:

```
<CodeList OID="CodeList.AESEVTXT"  Name="AECAUS" DataType="text"
                SASFormatName="$AESEV">
        <CodeListItem CodedValue="MILD" def:Rank="1.0" >
                <Decode>
                <TranslatedText xml:lang="en">Grade 1</TranslatedText>
                </Decode>
        </CodeListItem>
        <CodeListItem CodedValue="MODERATE" def:Rank="2.0" >
                <Decode>
                <TranslatedText xml:lang="en">Grade 2</TranslatedText>
                </Decode>
        </CodeListItem>
        <CodeListItem CodedValue="SEVERE" def:Rank="3.0" >
                <Decode>
                <TranslatedText xml:lang="en">Grade 3</TranslatedText>
                </Decode>
        </CodeListItem>
</CodeList>
```

@Rank replaces CodeListItem/@def:Rank

Define.xml 2.0:

```
<CodeList OID="CL.AESEV" Name="Severity/Intensity Scale for Adverse Events"
DataType="text" SASFormatName="$AESEV">
        <CodeListItem CodedValue="MILD" Rank="1" >
                <Decode>
                <TranslatedText xml:lang="en">Grade 1</TranslatedText>
                </Decode>
                <Alias Name="C41338" Context="nci:ExtCodeID"/>
        </CodeListItem>
        <CodeListItem CodedValue="MODERATE" Rank="2" >
                <Decode>
                <TranslatedText xml:lang="en">Grade 2</TranslatedText>
                </Decode>
                <Alias Name="C41339" Context="nci:ExtCodeID"/>
        </CodeListItem>
        <CodeListItem CodedValue="SEVERE" Rank="3" >
                <Decode>
                <TranslatedText xml:lang="en">Grade 3</TranslatedText>
                </Decode>
                <Alias Name="C41340" Context="nci:ExtCodeID"/>
        </CodeListItem>
        <Alias Name="C66769" Context="nci:ExtCodeID"/>
</CodeList>
```

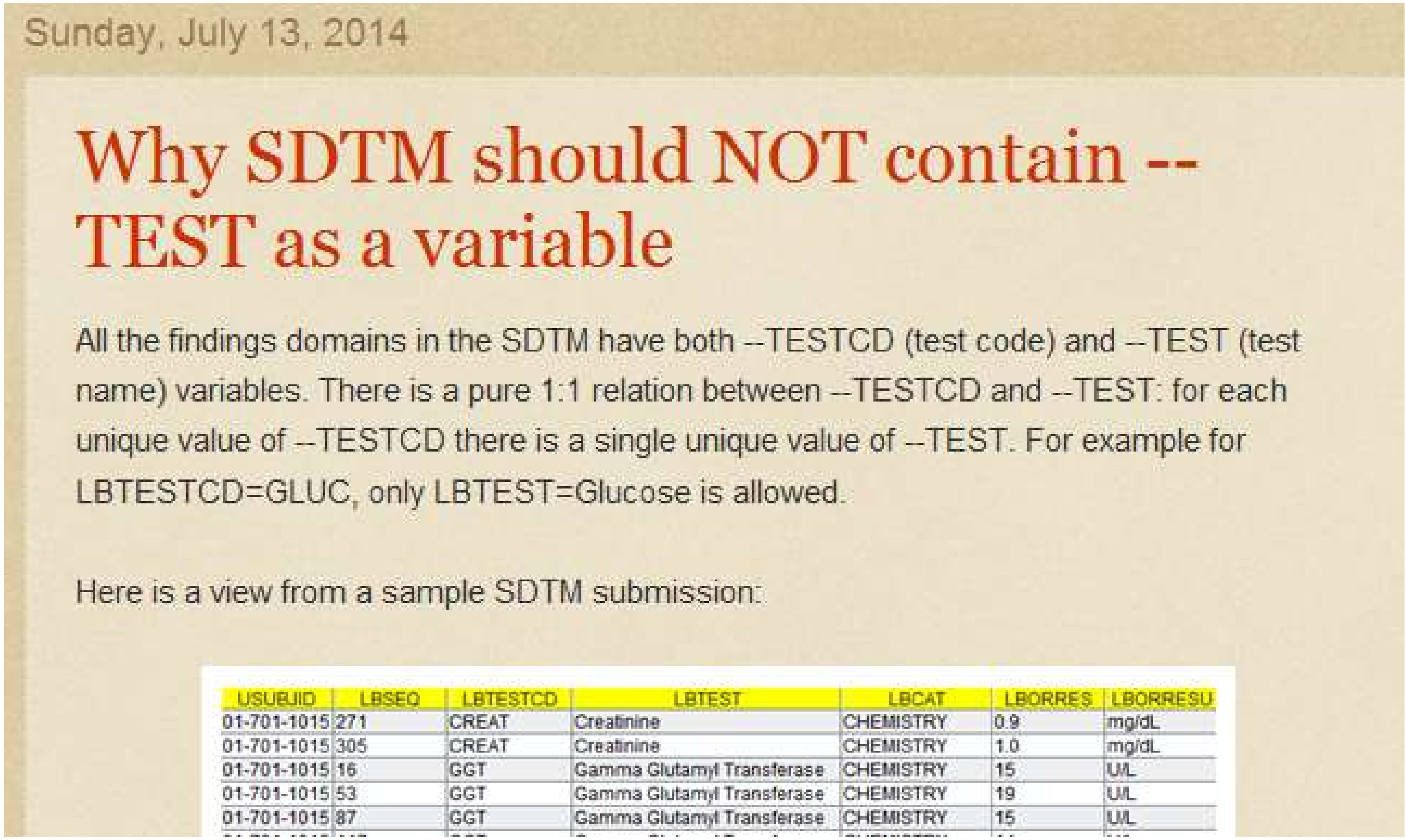
Usage of EnumeratedItem in CodeList

define.xml 2.0:

```
<CodeList OID="CL.VSTESTCD" Name="Vital Signs Test Code"
DataType="text" SASFormatName="$VSTESTC">
        <EnumeratedItem CodedValue="DIABP">
                <Alias Name="C25299" Context="nci:ExtCodeID"/>
        </EnumeratedItem >
        <EnumeratedItem CodedValue="FRMSIZE">
                <Alias Name="C49680" Context="nci:ExtCodeID"/>
        </EnumeratedItem>
…
</CodeList>

<CodeList OID="CL.VSTEST" Name="Vital Signs Test Name" DataType="text">
        <EnumeratedItem CodedValue="Diastolic Blood Pressure">
                <Alias Name="C25299" Context="nci:ExtCodeID"/>
        </EnumeratedItem>
        <EnumeratedItem CodedValue="Body Frame Size">
                <Alias Name="C49680" Context="nci:ExtCodeID"/>
        </EnumeratedItem> …
</CodeList>
```

--TESTCD / --TEST problems



Sunday, July 13, 2014

Why SDTM should NOT contain --TEST as a variable

All the findings domains in the SDTM have both --TESTCD (test code) and --TEST (test name) variables. There is a pure 1:1 relation between --TESTCD and --TEST: for each unique value of --TESTCD there is a single unique value of --TEST. For example for LBTESTCD=GLUC, only LBTEST=Glucose is allowed.

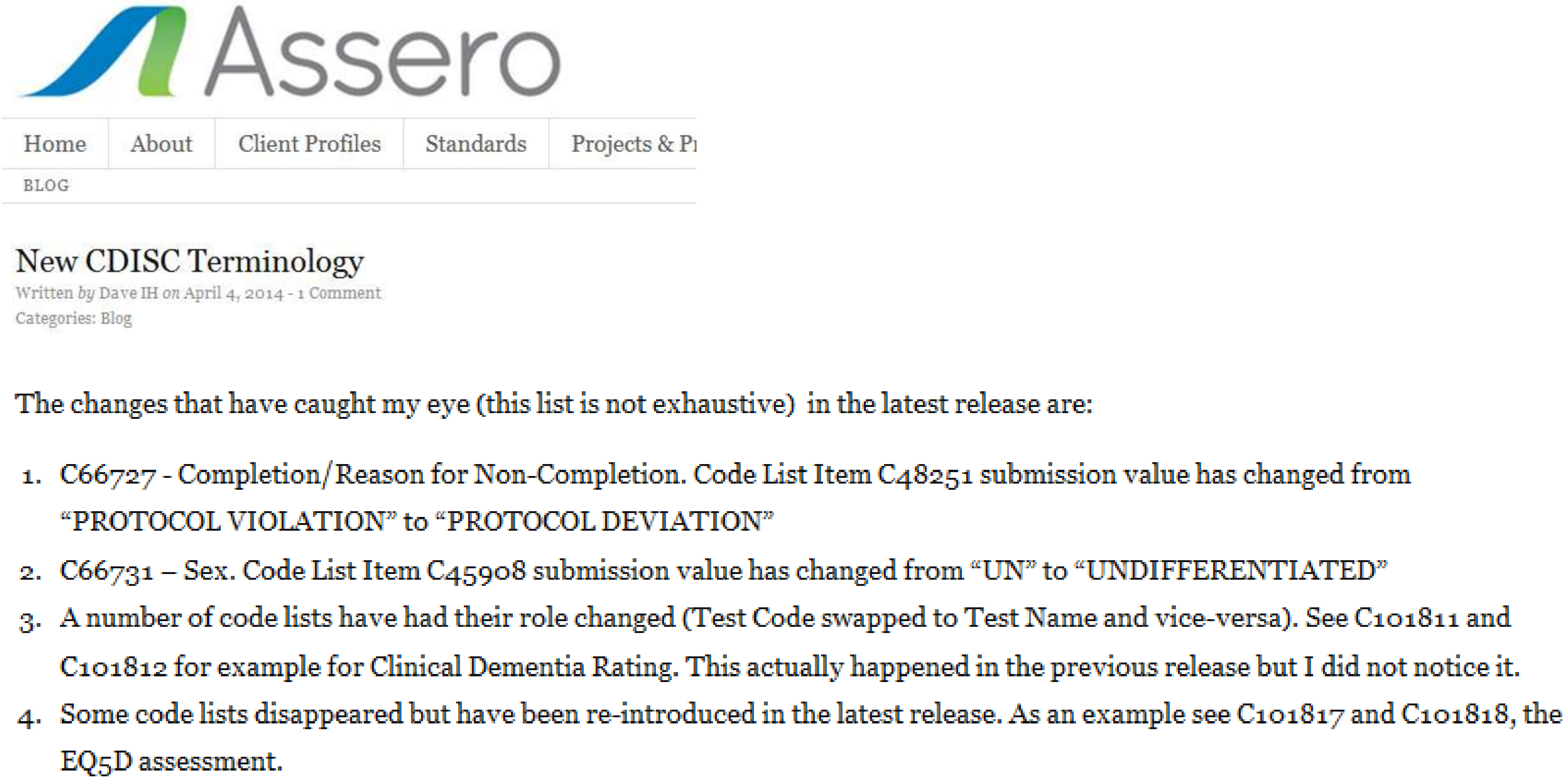
Here is a view from a sample SDTM submission:

USUBJID	LBSEQ	LBTESTCD	LBTEST	LBCAT	LBORRES	LBORRESU
01-701-1015	271	CREAT	Creatinine	CHEMISTRY	0.9	mg/dL
01-701-1015	305	CREAT	Creatinine	CHEMISTRY	1.0	mg/dL
01-701-1015	16	GGT	Gamma Glutamyl Transferase	CHEMISTRY	15	U/L
01-701-1015	53	GGT	Gamma Glutamyl Transferase	CHEMISTRY	19	U/L
01-701-1015	87	GGT	Gamma Glutamyl Transferase	CHEMISTRY	15	U/L

http://cdiscguru.blogspot.com/2014/07/why-sdtm-should-not-contain-test-as.html

--TESTCD / --TEST problems

- Depending on the version of the CDISC-CT, the --TESTCD / --TEST pairs can be different!



New CDISC Terminology

Written *by* Dave IH *on* April 4, 2014 - 1 Comment
Categories: Blog

The changes that have caught my eye (this list is not exhaustive) in the latest release are:

1. C66727 - Completion/Reason for Non-Completion. Code List Item C48251 submission value has changed from "PROTOCOL VIOLATION" to "PROTOCOL DEVIATION"
2. C66731 – Sex. Code List Item C45908 submission value has changed from "UN" to "UNDIFFERENTIATED"
3. A number of code lists have had their role changed (Test Code swapped to Test Name and vice-versa). See C101811 and C101812 for example for Clinical Dementia Rating. This actually happened in the previous release but I did not notice it.
4. Some code lists disappeared but have been re-introduced in the latest release. As an example see C101817 and C101818, the EQ5D assessment.

New "features" in define.xml 2.0

- def:Origin replaces Origin and has been enumerated
- def:CommentDef replaces Comment
  - reason: reusability

def:Origin replaces Origin and has been enumerated

- Each "ItemDef" (either on variable level or an valuelist level) MUST have a "def:Origin" when the context is an electronic submission
- E.g.:

```
<ItemDef OID="IT.AE.AEACN" Name="AEACN"  …>
            <Description>
            <TranslatedText xml:lang="en">Action Taken with Study
                        Treatment</TranslatedText>
            </Description>
            <CodeListRef CodeListOID="CL.ACN"/>
            <def:Origin Type="CRF">
                        <def:DocumentRef leafID="LF.blankcrf">
                        <def:PDFPageRef PageRefs="21" Type="PhysicalRef"/>
                        </def:DocumentRef>
            </def:Origin>
     </ItemDef>
```

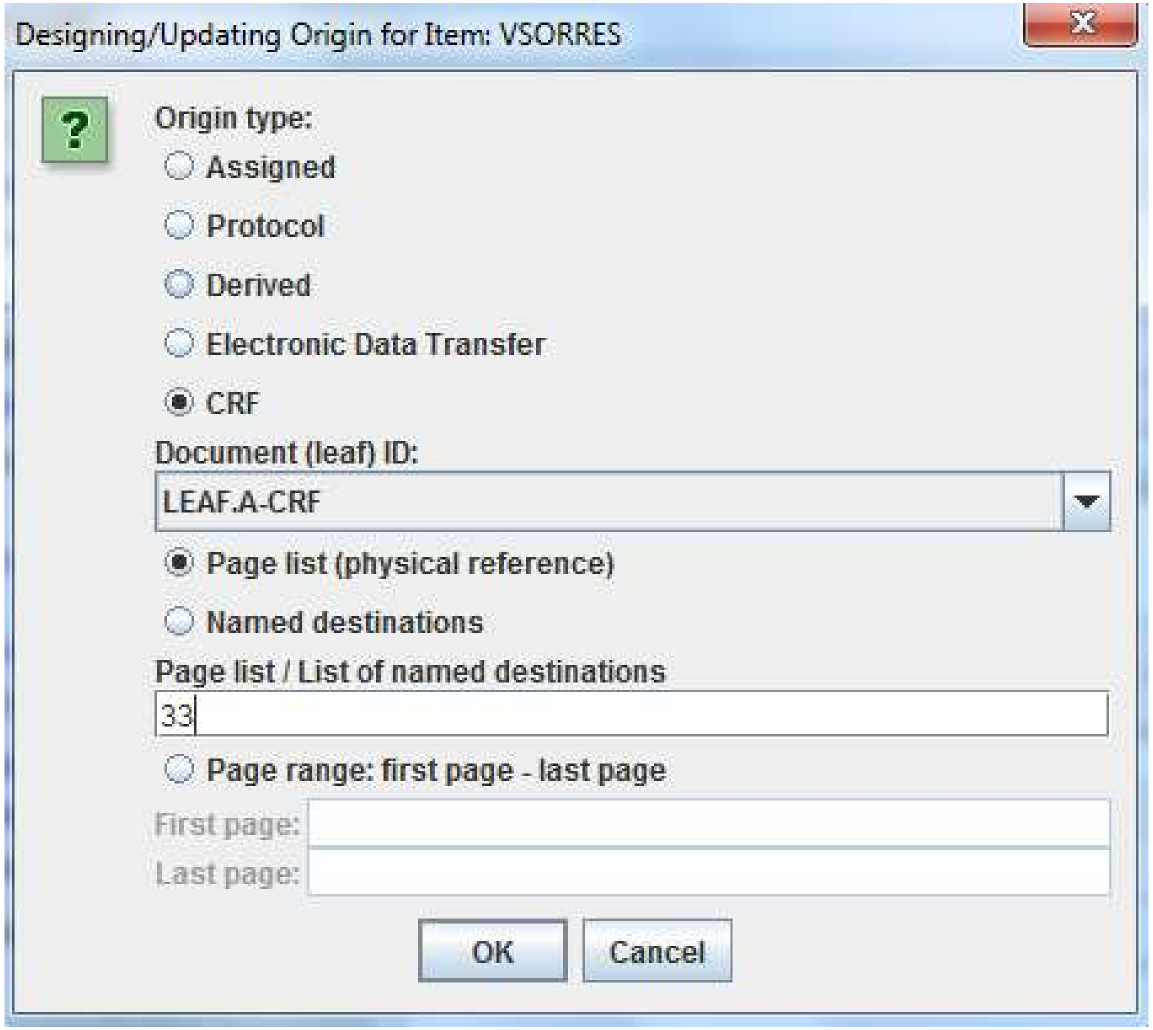
def:Origin

- Possible values for the "Type" attribute:
  - CRF => Case report form data
    - => def:DocumentRef MUST be present with reference to the annotated-CRF document with pages or named destination
  - Derived => Data is calculated (e.g. BMI)
  - Assigned => Data is obtained from individual judgement (e.g. --DECOD)
  - Protocol => taken or predefined from the protocol (e.g. VSPOS does not appear as choice on the CRF, but is fixed by the protocol)
  - eDT: electronic data transfer. E.g. data from ECG, Lab, …
  - Predecessor => Data is copied from another data set (usage mostly limited to ADaM-SDTM traceability)

Other example def:Origin

```
<ItemDef OID="IT.IE.IECAT" Name="IECAT" DataType="text" Length="9"
SASFieldName="IECAT">
        <Description>
        <TranslatedText xml:lang="en">Inclusion/Exclusion
                Category</TranslatedText>
        </Description>
        <def:Origin Type="CRF">
                <def:DocumentRef leafID="LF.blankcrf">
                <def:PDFPageRef FirstPage="4" LastPage="5"
                        Type="PhysicalRef"/>
                </def:DocumentRef>
        </def:Origin>
</ItemDef>
```

def:Origin: no fear: tools can do this for you



Designing/Updating Origin for Item: VSORRES

Origin type:

○ Assigned

○ Protocol

○ Derived

○ Electronic Data Transfer

● CRF

Document (leaf) ID: LEAF.A-CRF

● Page list (physical reference)

○ Named destinations

Page list / List of named destinations: 33

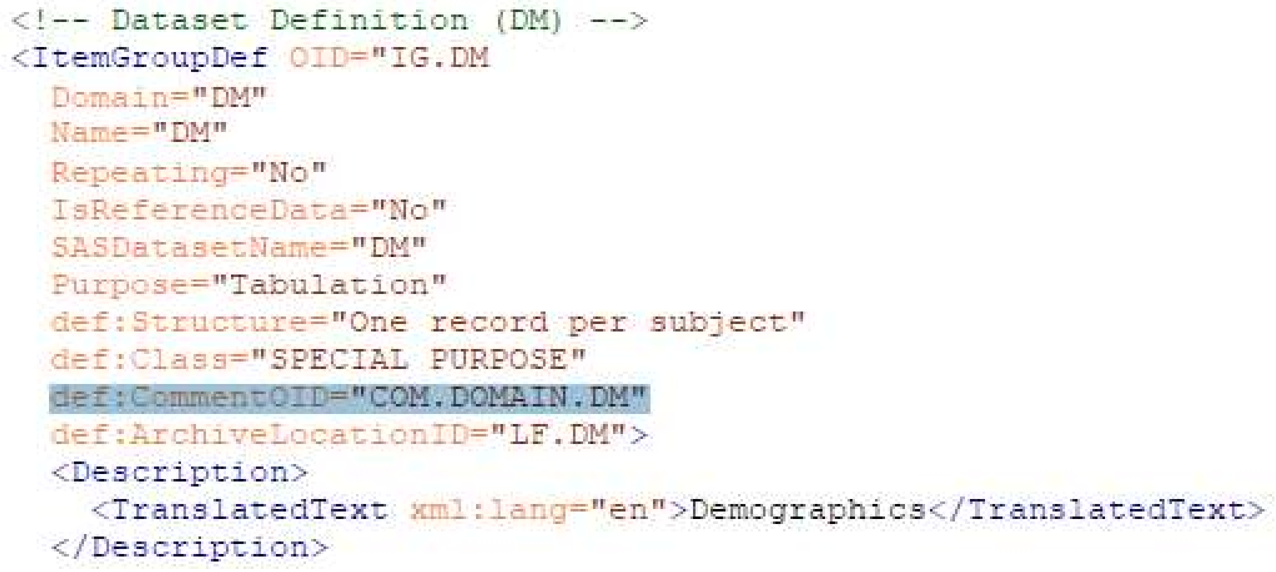
○ Page range: first page - last page

First page:

Last page:

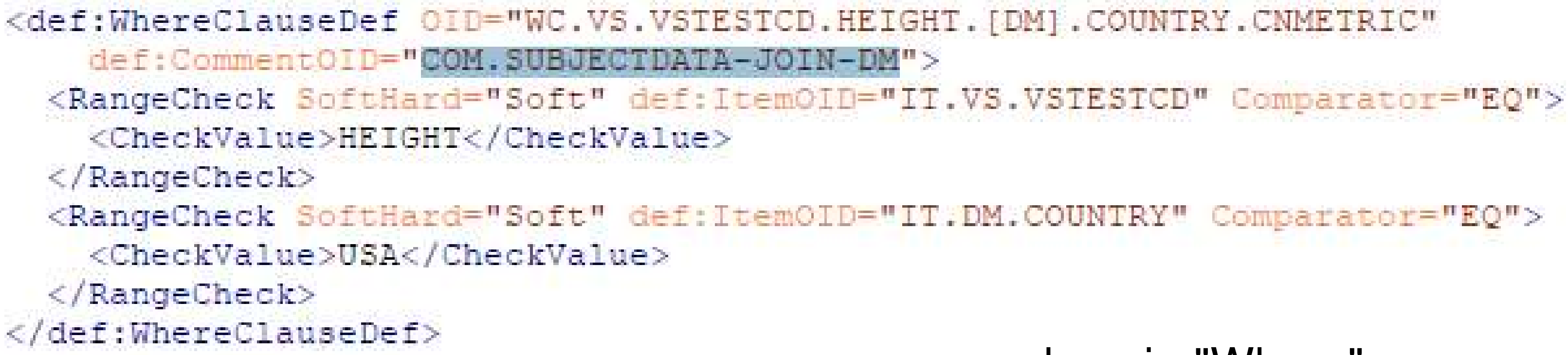
def:CommentDef replaces Comment

- can also contain a reference to an external document
- can be referenced from
  - ItemGroupDef (Domain/Dataset)
  - ItemDef (Variable)
  - def:WhereClauseDef (Valuelevel metadata)
    - required for cross-domain "where" expressions



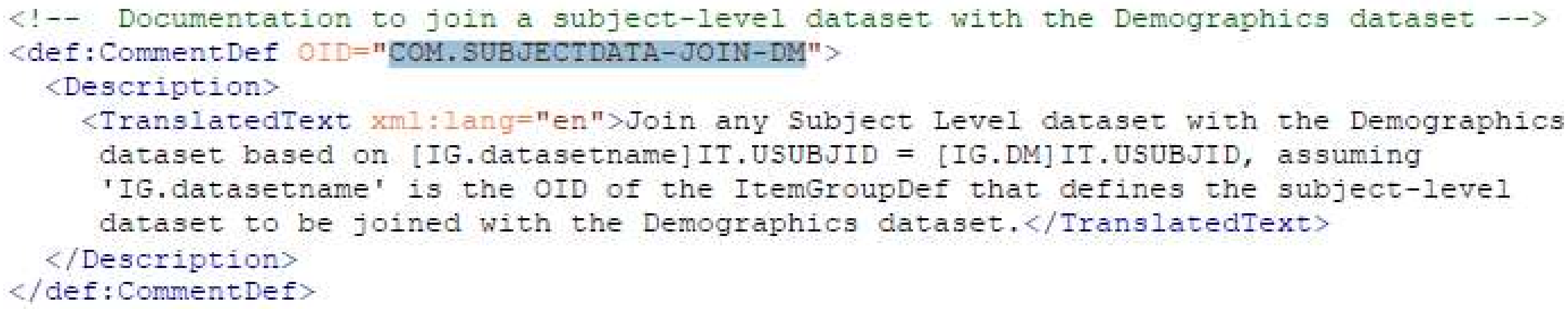
```
<!-- Dataset Definition (DM) -->
<ItemGroupDef OID="IG.DM
  Domain="DM"
  Name="DM"
  Repeating="No"
  IsReferenceData="No"
  SASDatasetName="DM"
  Purpose="Tabulation"
  def:Structure="One record per subject"
  def:Class="SPECIAL PURPOSE"
  def:CommentOID="COM.DOMAIN.DM"
  def:ArchiveLocationID="LF.DM">
  <Description>
    <TranslatedText xml:lang="en">Demographics</TranslatedText>
  </Description>
```

def:CommentDef



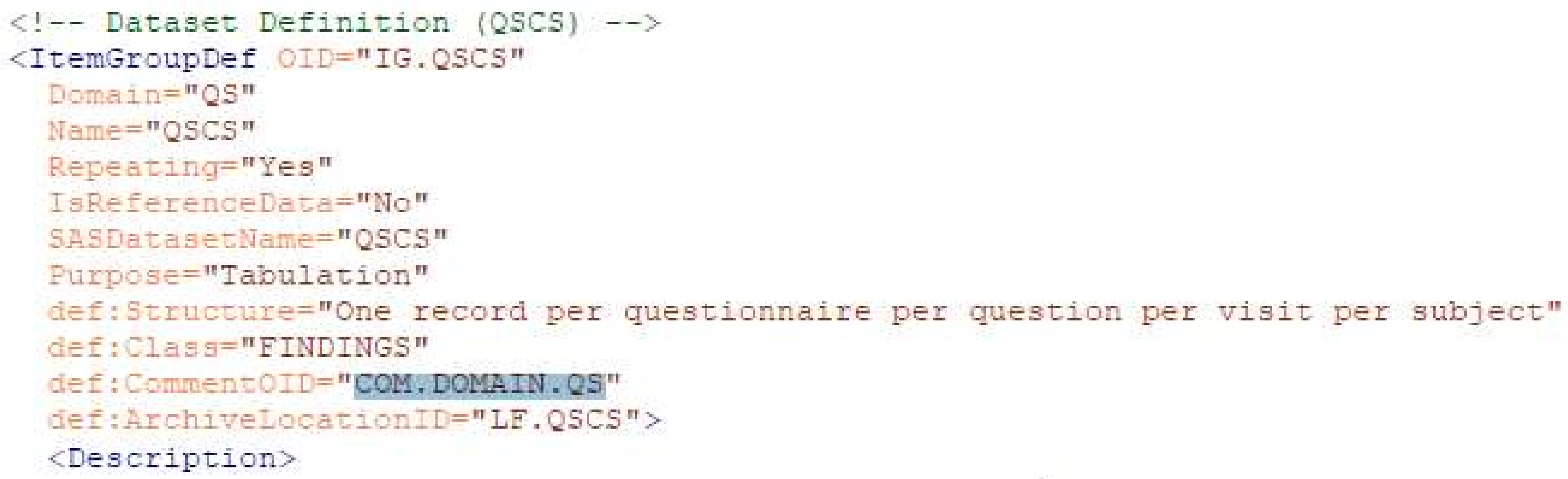
```
<def:WhereClauseDef OID="WC.VS.VSTESTCD.HEIGHT.[DM].COUNTRY.CNMETRIC"
    def:CommentOID="COM.SUBJECTDATA-JOIN-DM">
  <RangeCheck SoftHard="Soft" def:ItemOID="IT.VS.VSTESTCD" Comparator="EQ">
    <CheckValue>HEIGHT</CheckValue>
  </RangeCheck>
  <RangeCheck SoftHard="Soft" def:ItemOID="IT.DM.COUNTRY" Comparator="EQ">
    <CheckValue>USA</CheckValue>
  </RangeCheck>
</def:WhereClauseDef>
```

cross-domain "Where"

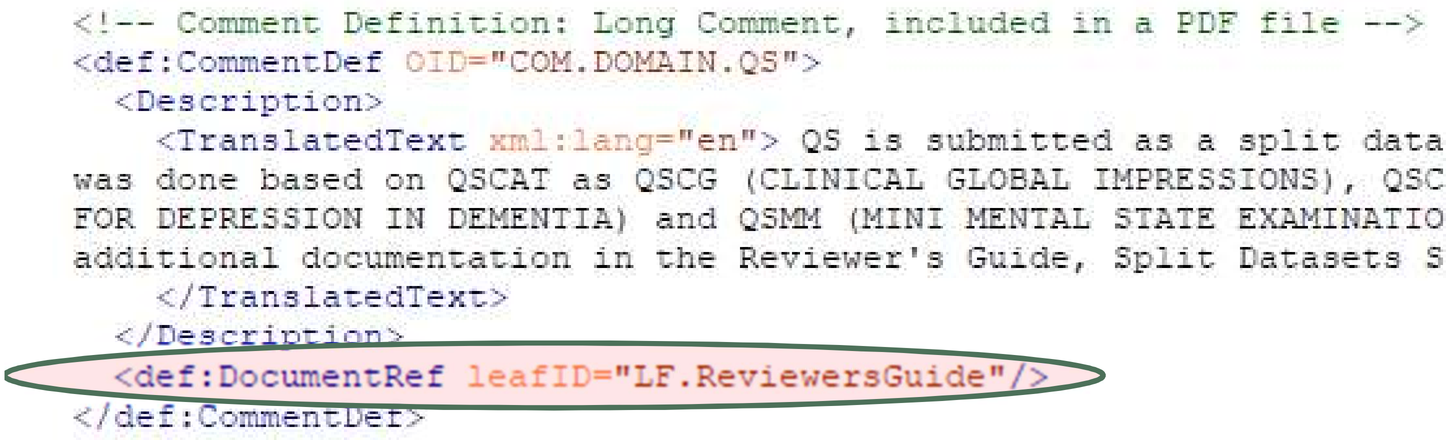


```
<!--  Documentation to join a subject-level dataset with the Demographics dataset -->
<def:CommentDef OID="COM.SUBJECTDATA-JOIN-DM">
  <Description>
    <TranslatedText xml:lang="en">Join any Subject Level dataset with the Demographics
     dataset based on [IG.datasetname]IT.USUBJID = [IG.DM]IT.USUBJID, assuming
     'IG.datasetname' is the OID of the ItemGroupDef that defines the subject-level
     dataset to be joined with the Demographics dataset.</TranslatedText>
  </Description>
</def:CommentDef>
```

def:CommentDef with link to ext. document

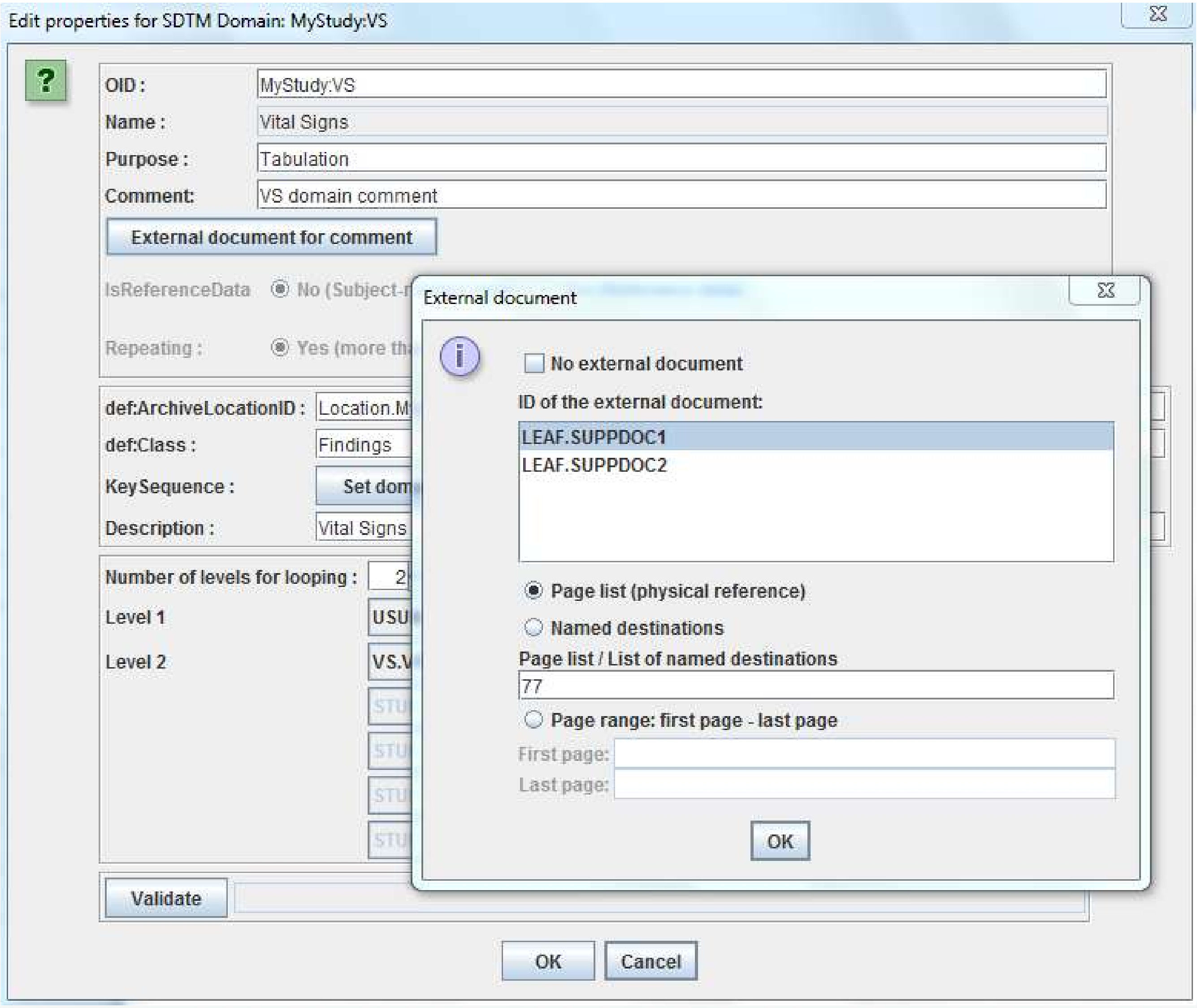


```
<!-- Dataset Definition (QSCS) -->
<ItemGroupDef OID="IG.QSCS"
  Domain="QS"
  Name="QSCS"
  Repeating="Yes"
  IsReferenceData="No"
  SASDatasetName="QSCS"
  Purpose="Tabulation"
  def:Structure="One record per questionnaire per question per visit per subject"
  def:Class="FINDINGS"
  def:CommentOID="COM.DOMAIN.QS"
  def:ArchiveLocationID="LF.QSCS">
  <Description>
```



```
<!-- Comment Definition: Long Comment, included in a PDF file -->
<def:CommentDef OID="COM.DOMAIN.QS">
  <Description>
    <TranslatedText xml:lang="en"> QS is submitted as a split dataset. The split
was done based on QSCAT as QSCG (CLINICAL GLOBAL IMPRESSIONS), QSCS (CORNELL SCALE
FOR DEPRESSION IN DEMENTIA) and QSMM (MINI MENTAL STATE EXAMINATION). See
additional documentation in the Reviewer's Guide, Split Datasets Section.
    </TranslatedText>
  </Description>
  <def:DocumentRef leafID="LF.ReviewersGuide"/>
</def:CommentDef>
```

def:CommentDef:
no fear: tools can do this for you



Edit properties for SDTM Domain: MyStudy:VS

OID : MyStudy:VS

Name : Vital Signs

Purpose : Tabulation

Comment: VS domain comment

IsReferenceData ● No (Subject-r

Repeating : ● Yes (more th

def:ArchiveLocationID : Location.M

def:Class : Findings

KeySequence : Set dom

Description : Vital Signs

Number of levels for looping : 2

Level 1 USU

Level 2 VS.V

External document

□ No external document

ID of the external document:

LEAF.SUPPDOC1
LEAF.SUPPDOC2

● Page list (physical reference)

○ Named destinations

Page list / List of named destinations

77

○ Page range: first page - last page

First page:

Last page:

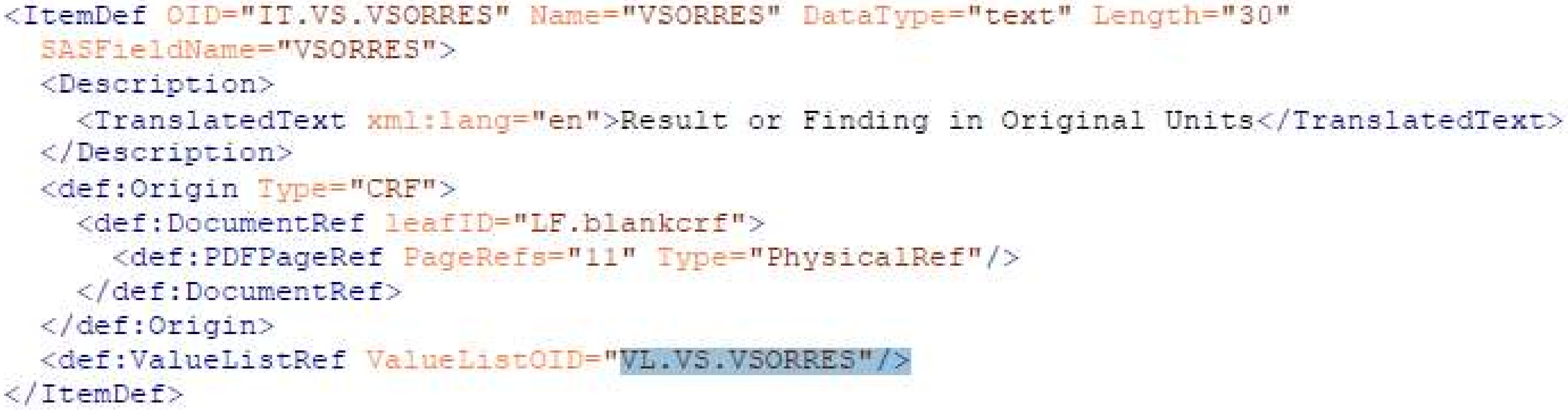
ValueLists

- could be added to any variable in define.xml 1.0 but was conventionally added to --TESTCD (assuming application on --ORRES)

- may only be added to the variables on which they apply directly in define.xml 2.0, e.g.:
    - --ORRES
    - --ORRESU
    - --STRES
    - --STRESU
    - --DSTERM, --DSDECOD based on --DSCAT
    - ADaM: AVAL or AVALC in BDS data structures based on values of PARAMCD

ValueList examples

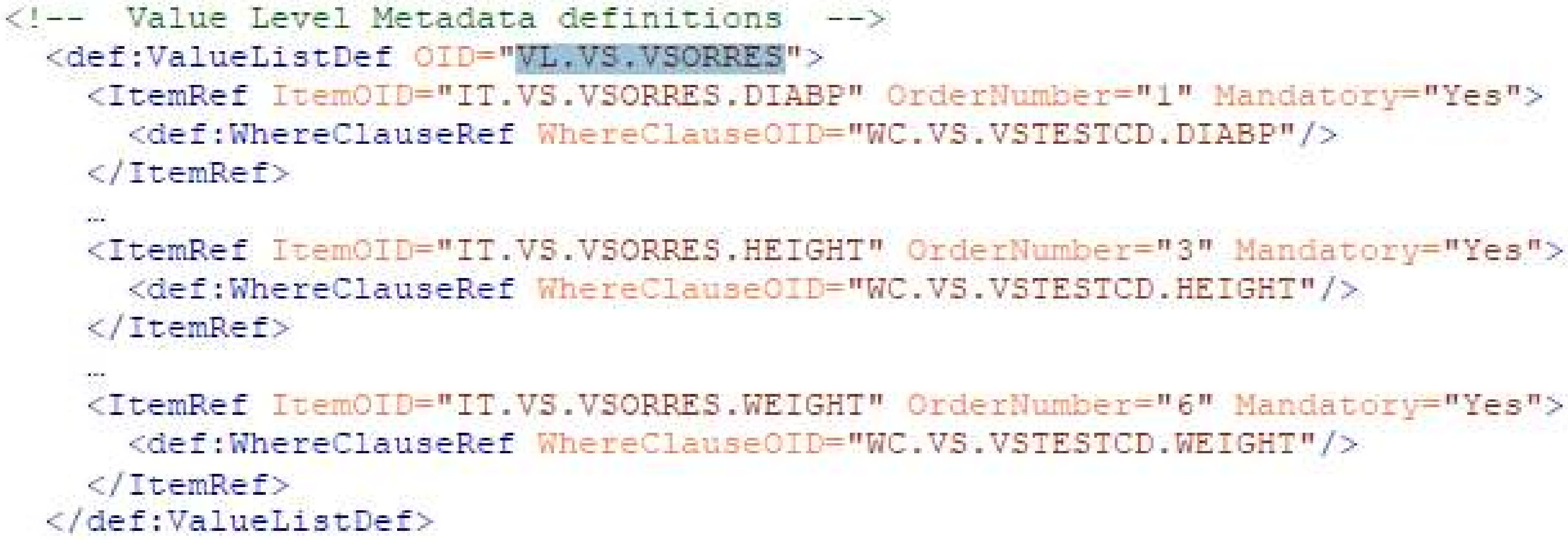
- Typical case: VSORRES
- data type and (max.) length depend on VSTESTCD



```
<ItemDef OID="IT.VS.VSORRES" Name="VSORRES" DataType="text" Length="30"
  SASFieldName="VSORRES">
  <Description>
    <TranslatedText xml:lang="en">Result or Finding in Original Units</TranslatedText>
  </Description>
  <def:Origin Type="CRF">
    <def:DocumentRef leafID="LF.blankcrf">
      <def:PDFPageRef PageRefs="11" Type="PhysicalRef"/>
    </def:DocumentRef>
  </def:Origin>
  <def:ValueListRef ValueListOID="VL.VS.VSORRES"/>
</ItemDef>
```

ValueList examples

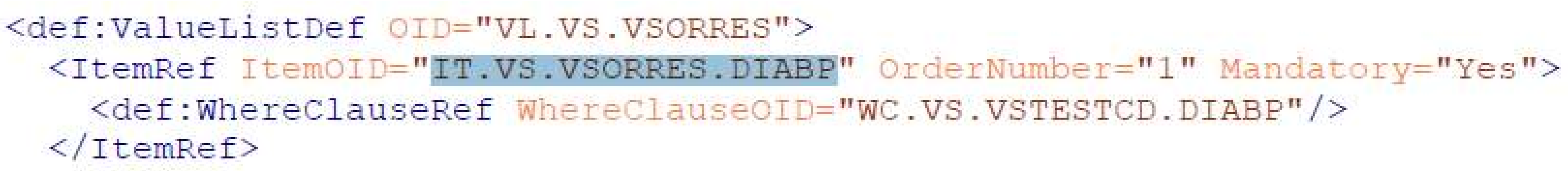
- Typical case: VSORRES
- data type and (max.) length depend on VSTESTCD



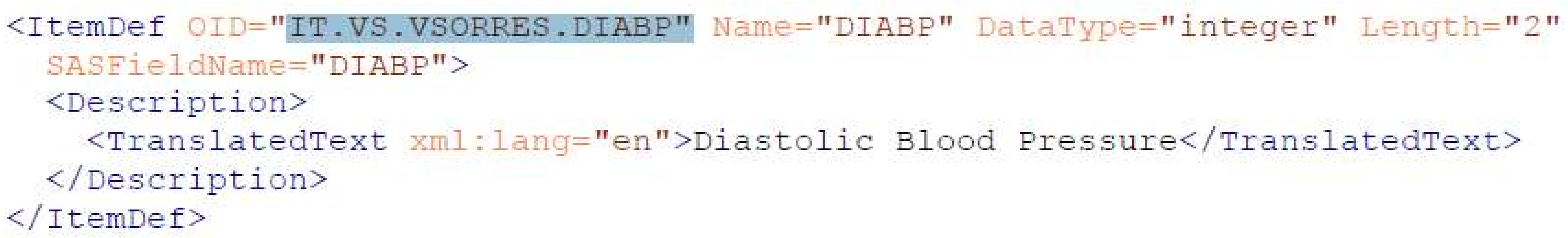
```
<!--  Value Level Metadata definitions  -->
  <def:ValueListDef OID="VL.VS.VSORRES">
    <ItemRef ItemOID="IT.VS.VSORRES.DIABP" OrderNumber="1" Mandatory="Yes">
      <def:WhereClauseRef WhereClauseOID="WC.VS.VSTESTCD.DIABP"/>
    </ItemRef>
    …
    <ItemRef ItemOID="IT.VS.VSORRES.HEIGHT" OrderNumber="3" Mandatory="Yes">
      <def:WhereClauseRef WhereClauseOID="WC.VS.VSTESTCD.HEIGHT"/>
    </ItemRef>
    …
    <ItemRef ItemOID="IT.VS.VSORRES.WEIGHT" OrderNumber="6" Mandatory="Yes">
      <def:WhereClauseRef WhereClauseOID="WC.VS.VSTESTCD.WEIGHT"/>
    </ItemRef>
  </def:ValueListDef>
```

ValueList examples

- Typical case: VSORRES
- ItemDef "IT.VS.VSORRES.DIABP" applies when …



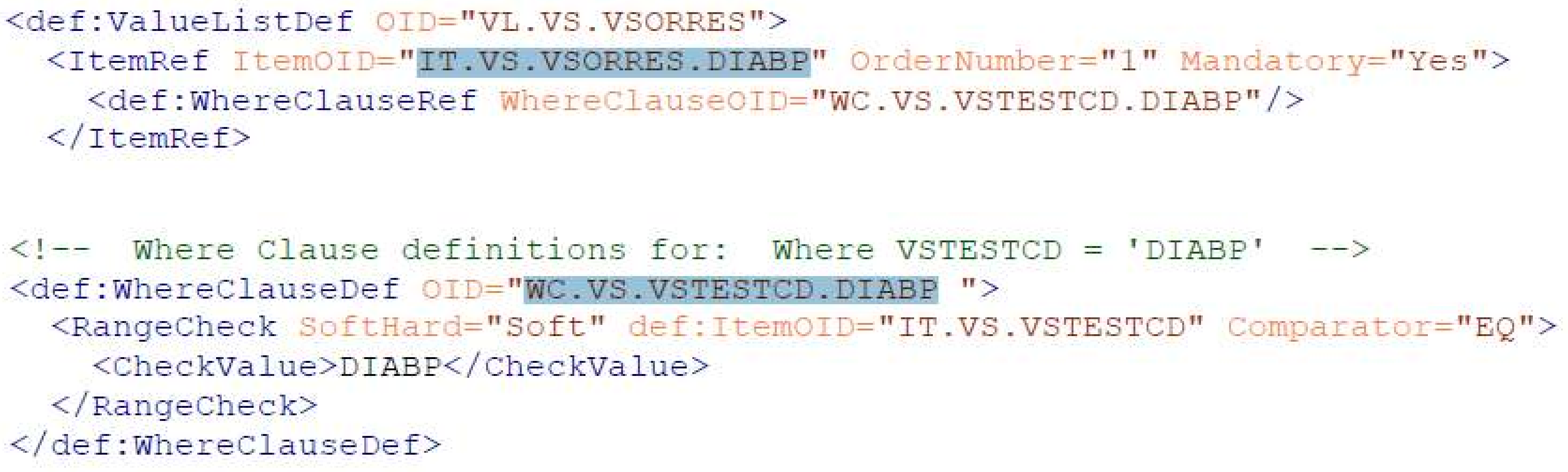
```
<def:ValueListDef OID="VL.VS.VSORRES">
  <ItemRef ItemOID="IT.VS.VSORRES.DIABP" OrderNumber="1" Mandatory="Yes">
    <def:WhereClauseRef WhereClauseOID="WC.VS.VSTESTCD.DIABP"/>
  </ItemRef>
```



```
<ItemDef OID="IT.VS.VSORRES.DIABP" Name="DIABP" DataType="integer" Length="2"
  SASFieldName="DIABP">
  <Description>
    <TranslatedText xml:lang="en">Diastolic Blood Pressure</TranslatedText>
  </Description>
</ItemDef>
```

ValueList examples

- Typical case: VSORRES
- ItemDef "IT.VS.VSORRES.DIABP" applies when …
- VSTESTCD = 'DIABP'



```
<def:ValueListDef OID="VL.VS.VSORRES">
  <ItemRef ItemOID="IT.VS.VSORRES.DIABP" OrderNumber="1" Mandatory="Yes">
    <def:WhereClauseRef WhereClauseOID="WC.VS.VSTESTCD.DIABP"/>
  </ItemRef>


<!--  Where Clause definitions for:  Where VSTESTCD = 'DIABP'  -->
<def:WhereClauseDef OID="WC.VS.VSTESTCD.DIABP ">
  <RangeCheck SoftHard="Soft" def:ItemOID="IT.VS.VSTESTCD" Comparator="EQ">
    <CheckValue>DIABP</CheckValue>
  </RangeCheck>
</def:WhereClauseDef>
```

"IT.VS.VSTESTCD" is compared with 'DIABP'

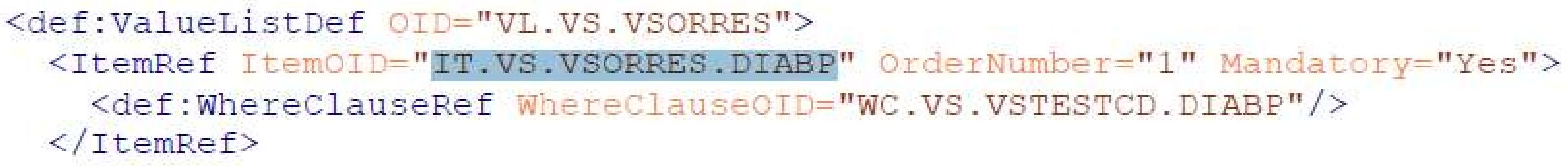
def:WhereClause

- used for value level metadata

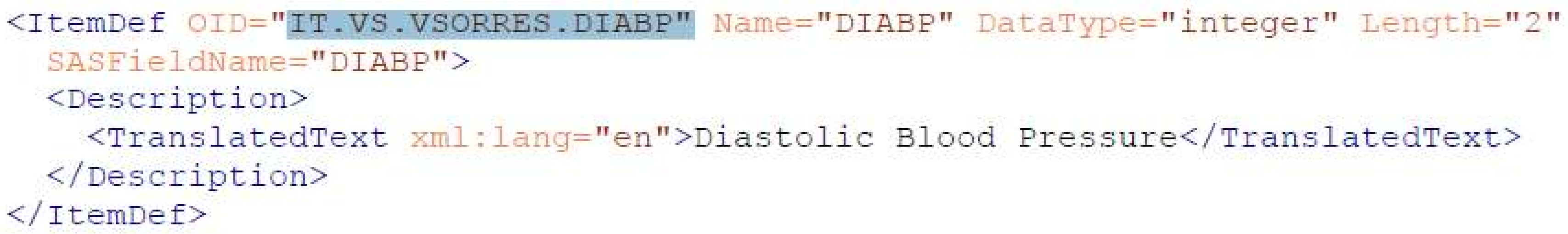
- describe the conditions under which the definition of a value applies
- in a machine-readable form

- With def:WhereClause there is no need anymore for nested ValueLists
  - nested ValueLists should NOT be used in define.xml 2.0

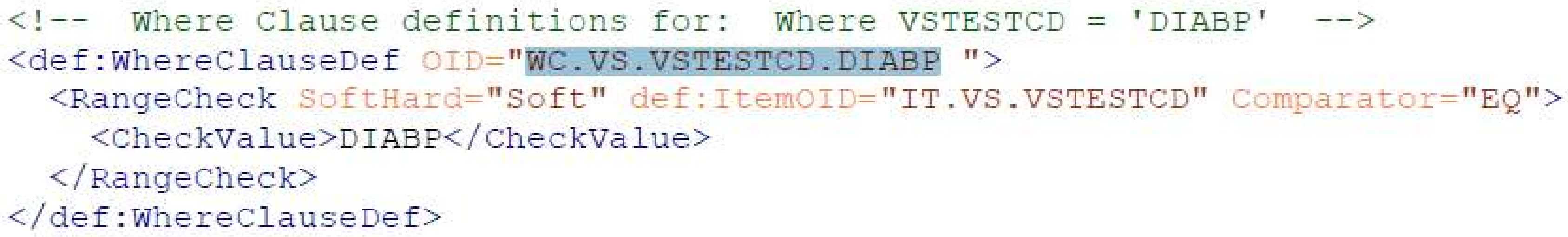
def:WhereClause in detail



```
<def:ValueListDef OID="VL.VS.VSORRES">
  <ItemRef ItemOID="IT.VS.VSORRES.DIABP" OrderNumber="1" Mandatory="Yes">
    <def:WhereClauseRef WhereClauseOID="WC.VS.VSTESTCD.DIABP"/>
  </ItemRef>
```



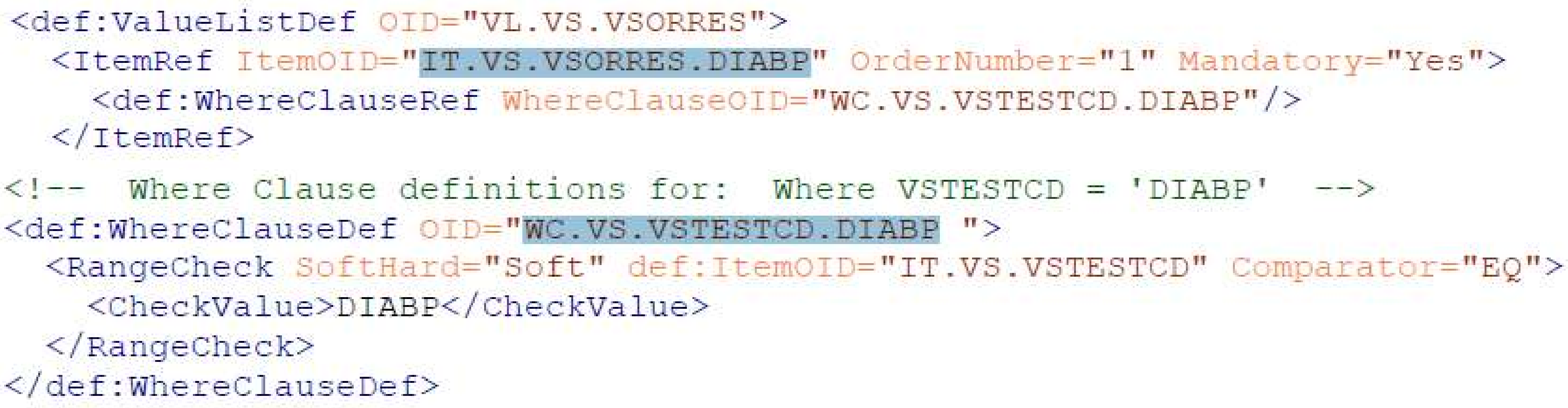
```
<ItemDef OID="IT.VS.VSORRES.DIABP" Name="DIABP" DataType="integer" Length="2"
  SASFieldName="DIABP">
  <Description>
    <TranslatedText xml:lang="en">Diastolic Blood Pressure</TranslatedText>
  </Description>
</ItemDef>
```



```
 <!--  Where Clause definitions for:  Where VSTESTCD = 'DIABP'  -->
 <def:WhereClauseDef OID="WC.VS.VSTESTCD.DIABP ">
   <RangeCheck SoftHard="Soft" def:ItemOID="IT.VS.VSTESTCD" Comparator="EQ">
     <CheckValue>DIABP</CheckValue>
   </RangeCheck>
 </def:WhereClauseDef>
```

Or: use integer for VSORRES when VSTESTCD=DIABP

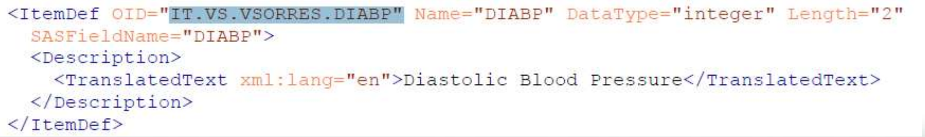
WhereClause in detail



```
<def:ValueListDef OID="VL.VS.VSORRES">
  <ItemRef ItemOID="IT.VS.VSORRES.DIABP" OrderNumber="1" Mandatory="Yes">
    <def:WhereClauseRef WhereClauseOID="WC.VS.VSTESTCD.DIABP"/>
  </ItemRef>
<!--  Where Clause definitions for:  Where VSTESTCD = 'DIABP'  -->
<def:WhereClauseDef OID="WC.VS.VSTESTCD.DIABP ">
  <RangeCheck SoftHard="Soft" def:ItemOID="IT.VS.VSTESTCD" Comparator="EQ">
    <CheckValue>DIABP</CheckValue>
  </RangeCheck>
</def:WhereClauseDef>
```

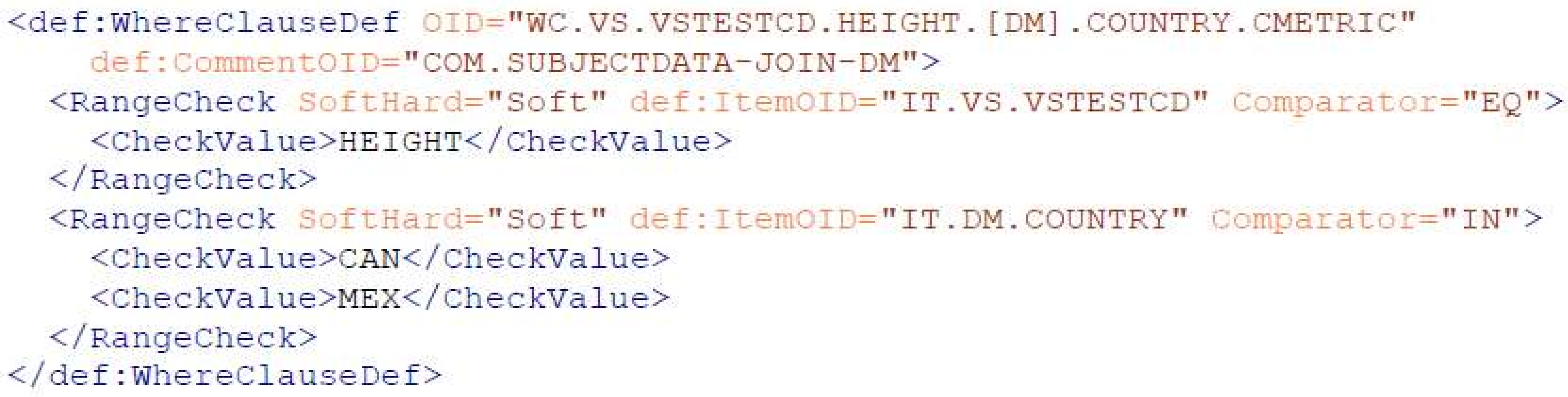
- RangeCheck => describes the condition
- Comparator => how values are compared
  - in this case "EQ" (equals)
- def:ItemOID => left-hand site of comparison - ref
- CheckValue => right-hand site of comparison

Length = 2; DataType = integer; SASFieldName = DIABP
applies WHEN VSTESTCD = DIABP



```
<ItemDef OID="IT.VS.VSORRES.DIABP" Name="DIABP" DataType="integer" Length="2"
  SASFieldName="DIABP">
  <Description>
    <TranslatedText xml:lang="en">Diastolic Blood Pressure</TranslatedText>
  </Description>
</ItemDef>
```

WhereClause - a more complicated example

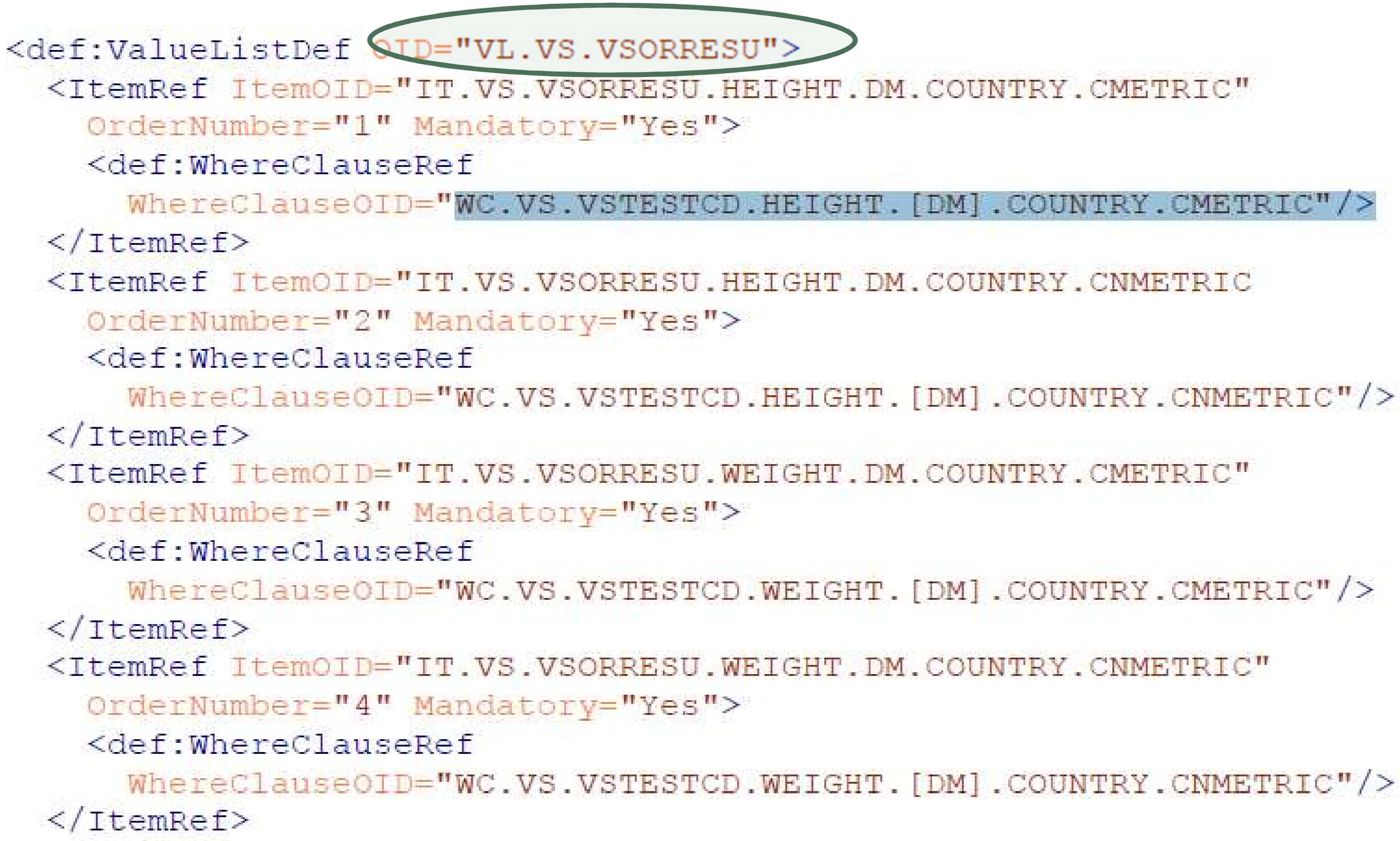


```
<def:WhereClauseDef OID="WC.VS.VSTESTCD.HEIGHT.[DM].COUNTRY.CMETRIC"
    def:CommentOID="COM.SUBJECTDATA-JOIN-DM">
  <RangeCheck SoftHard="Soft" def:ItemOID="IT.VS.VSTESTCD" Comparator="EQ">
    <CheckValue>HEIGHT</CheckValue>
  </RangeCheck>
  <RangeCheck SoftHard="Soft" def:ItemOID="IT.DM.COUNTRY" Comparator="IN">
    <CheckValue>CAN</CheckValue>
    <CheckValue>MEX</CheckValue>
  </RangeCheck>
</def:WhereClauseDef>
```

Describes a cross-domain condition that:

- the value of VSTESTCD equals "HEIGHT"
  AND
- the value of DM.COUNTRY is either
  - CAN
    OR
  - MEX

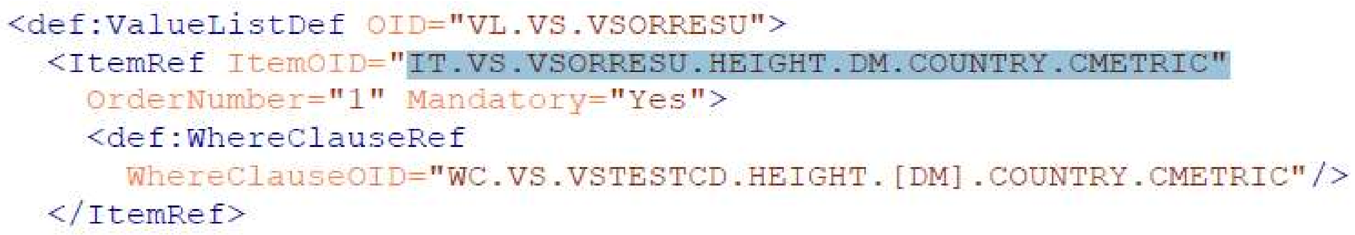
WhereClause - a more complicated example



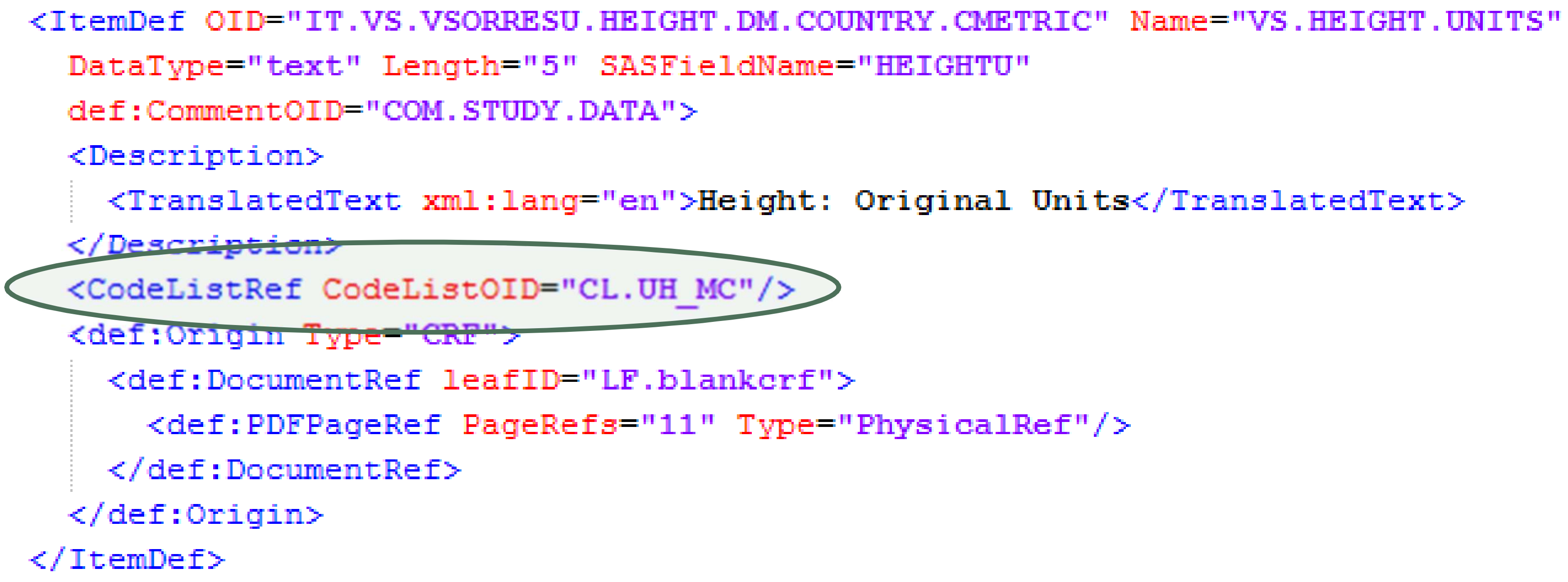
```
<def:ValueListDef OID="VL.VS.VSORRESU">
  <ItemRef ItemOID="IT.VS.VSORRESU.HEIGHT.DM.COUNTRY.CMETRIC"
    OrderNumber="1" Mandatory="Yes">
    <def:WhereClauseRef
      WhereClauseOID="WC.VS.VSTESTCD.HEIGHT.[DM].COUNTRY.CMETRIC"/>
  </ItemRef>
  <ItemRef ItemOID="IT.VS.VSORRESU.HEIGHT.DM.COUNTRY.CNMETRIC
    OrderNumber="2" Mandatory="Yes">
    <def:WhereClauseRef
      WhereClauseOID="WC.VS.VSTESTCD.HEIGHT.[DM].COUNTRY.CNMETRIC"/>
  </ItemRef>
  <ItemRef ItemOID="IT.VS.VSORRESU.WEIGHT.DM.COUNTRY.CMETRIC"
    OrderNumber="3" Mandatory="Yes">
    <def:WhereClauseRef
      WhereClauseOID="WC.VS.VSTESTCD.WEIGHT.[DM].COUNTRY.CMETRIC"/>
  </ItemRef>
  <ItemRef ItemOID="IT.VS.VSORRESU.WEIGHT.DM.COUNTRY.CNMETRIC"
    OrderNumber="4" Mandatory="Yes">
    <def:WhereClauseRef
      WhereClauseOID="WC.VS.VSTESTCD.WEIGHT.[DM].COUNTRY.CNMETRIC"/>
  </ItemRef>
```

The condition applies to the SDTM variable VSORRESU

WhereClause - a more complicated example



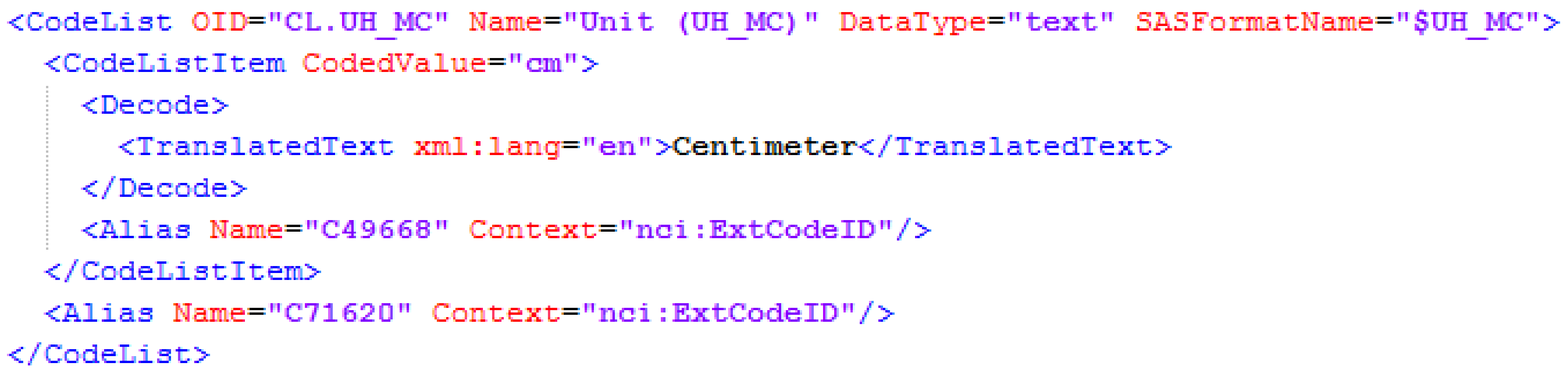
```
<def:ValueListDef OID="VL.VS.VSORRESU">
  <ItemRef ItemOID="IT.VS.VSORRESU.HEIGHT.DM.COUNTRY.CMETRIC"
    OrderNumber="1" Mandatory="Yes">
    <def:WhereClauseRef
      WhereClauseOID="WC.VS.VSTESTCD.HEIGHT.[DM].COUNTRY.CMETRIC"/>
  </ItemRef>
```



```
<ItemDef OID="IT.VS.VSORRESU.HEIGHT.DM.COUNTRY.CMETRIC" Name="VS.HEIGHT.UNITS"
  DataType="text" Length="5" SASFieldName="HEIGHTU"
  def:CommentOID="COM.STUDY.DATA">
  <Description>
    <TranslatedText xml:lang="en">Height: Original Units</TranslatedText>
  </Description>
  <CodeListRef CodeListOID="CL.UH_MC"/>
  <def:Origin Type="CRF">
    <def:DocumentRef leafID="LF.blankcrf">
      <def:PDFPageRef PageRefs="11" Type="PhysicalRef"/>
    </def:DocumentRef>
  </def:Origin>
</ItemDef>
```

The condition applies to the SDTM variable VSORRESU
and the value must be one from the codelist CL.UH_MC

WhereClause - a more complicated example

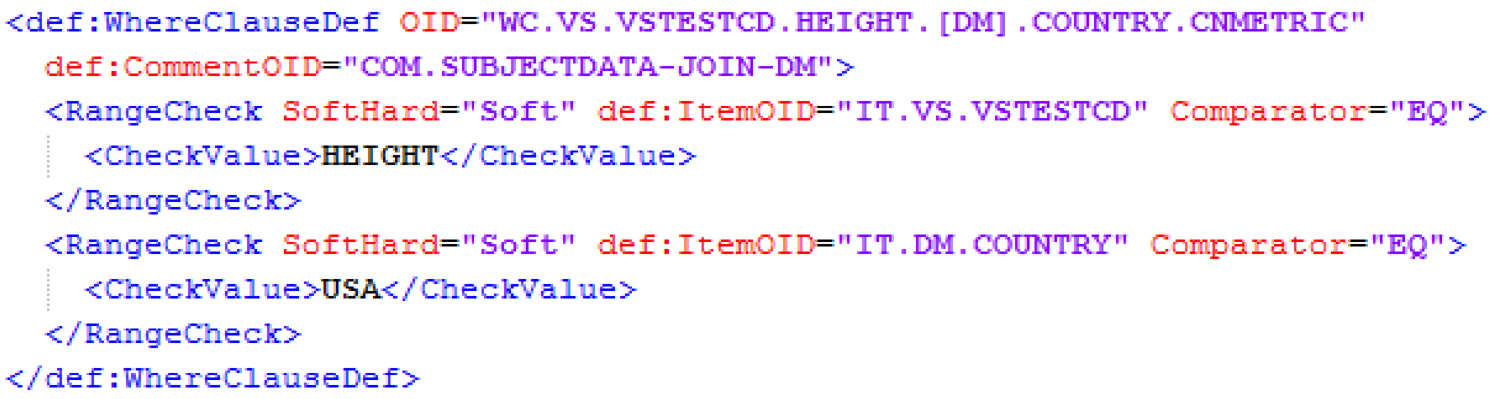


```
<CodeList OID="CL.UH_MC" Name="Unit (UH_MC)" DataType="text" SASFormatName="$UH_MC">
  <CodeListItem CodedValue="cm">
    <Decode>
      <TranslatedText xml:lang="en">Centimeter</TranslatedText>
    </Decode>
    <Alias Name="C49668" Context="nci:ExtCodeID"/>
  </CodeListItem>
  <Alias Name="C71620" Context="nci:ExtCodeID"/>
</CodeList>
```

The condition applies to the SDTM variable VSORRESU
and the value must be one from the codelist CL.UH_MC
which has only "centimeters" in it

So what is described here is:
When VSTESTCD = 'HEIGHT'
and the country (DM.COUNTRY) is either 'MEX' (Mexico) or 'CAN' (Canada)
Then the only allowed value for VSORRESU is 'cm' (centimeters)

Similar rule for VSTESTCD=HEIGHT and DM.COUNTRY=USA



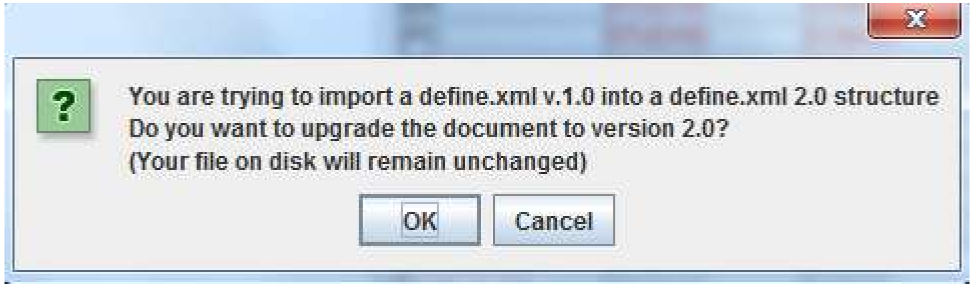
```
<def:WhereClauseDef OID="WC.VS.VSTESTCD.HEIGHT.[DM].COUNTRY.CNMETRIC"
  def:CommentOID="COM.SUBJECTDATA-JOIN-DM">
  <RangeCheck SoftHard="Soft" def:ItemOID="IT.VS.VSTESTCD" Comparator="EQ">
    <CheckValue>HEIGHT</CheckValue>
  </RangeCheck>
  <RangeCheck SoftHard="Soft" def:ItemOID="IT.DM.COUNTRY" Comparator="EQ">
    <CheckValue>USA</CheckValue>
  </RangeCheck>
</def:WhereClauseDef>
```

- further stating that in that case, VSORRESU can only have the value "IN" (inches)
- Remark that the rule (def:WhereClauseDef) could be reused:
  - to state that for LBTESTCD=WEIGHT
  - VSORRESU must be "kg" when DM.COUNTRY=MEX,CAN
  - VSORRESU must be "LB" (Pounds) when DM.COUNTRY=USA

How to move from define.xml 1.0 to define.xml 2.0 ?

How to move from define.xml 1.0 to define.xml 2.0 ?

- FDA does not force you to submit using v.2.0
- You can still use define.xml v.1.0 with Dataset-XML (though define.xml 2.0 is recommended)
- Some tools automatically update define.xml 1.0 into (partial) define.xml 2.0



You are trying to import a define.xml v.1.0 into a define.xml 2.0 structure
Do you want to upgrade the document to version 2.0?
(Your file on disk will remain unchanged)

- You can obtain an XSLT stylesheet that generates a prototype define.xml 2.0 from a define.xml 1.0

What does the stylesheet do?

- Moves everything to ODM 1.3 / define.xml namespaces
- def:Label => Description
- @def:Rank => @Rank
- def:ComputationMethod => MethodDef
- ItemDef/@Comment => def:CommentDef

What does the stylesheet NOT do?

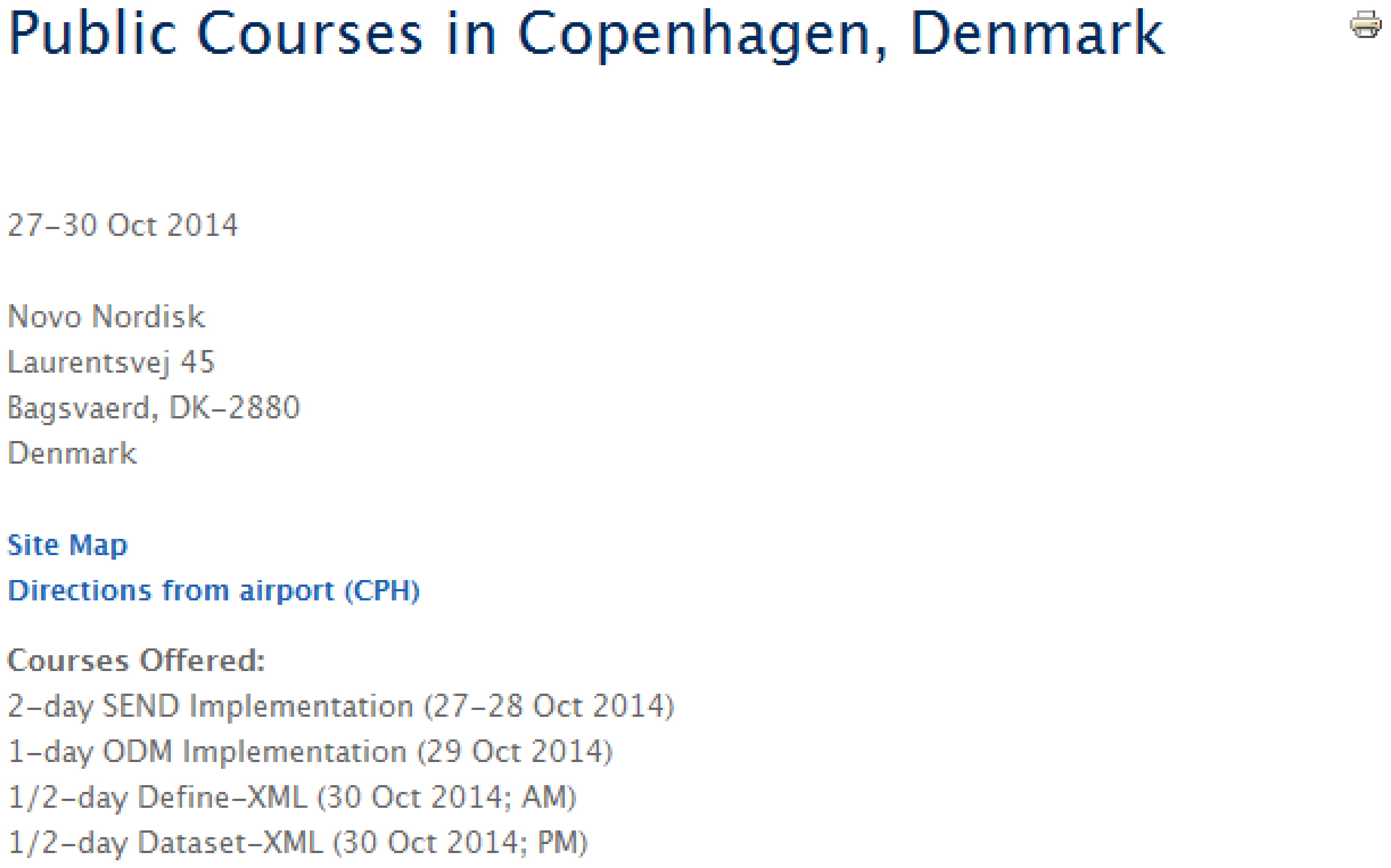
- Does not create page numbers for a-CRFs using information from the old @Origin attribute
- Does not generate @def:KeySequence attributes from old @def:DomainKeys attribute
- Does not generate information that was not in define.xml v.1.0
  - you will need to add this yourself

How to move from define.xml 1.0 to define.xml 2.0 ?

- Get a process in place
- Ideally: you should generate your define.xml (1.0 or 2.0) in parallel with your mappings / dataset generation automatically

- Points to take special care of:
  - Where clauses for ValueLists
  - for regulatory submissions:
    - each variable MUST have an Origin

Learning more about define.xml 2.0



Public Courses in Copenhagen, Denmark

27-30 Oct 2014

Novo Nordisk
Laurentsvej 45
Bagsvaerd, DK-2880
Denmark

Site Map
Directions from airport (CPH)

Courses Offered:
2-day SEND Implementation (27-28 Oct 2014)
1-day ODM Implementation (29 Oct 2014)
1/2-day Define-XML (30 Oct 2014; AM)
1/2-day Dataset-XML (30 Oct 2014; PM)

http://cdisc.org/public-courses-copenhagen-denmark

Thank you for your attention