GENERATING Define.xml

Uma Sarath Annapareddy, Sr.Statistical Programmer Consultant, Independent Consultant, Woodbridge, NJ
Sandeep Kottam, Sr.Statistical/SDTM Programmer consultant, Independent Consultant, NJ.
Sree Lakshmi K Tripuraneni, Sr.Statistical Programmer consultant, Purdue Pharma, Stamford, CT.

ABSTRACT:

CDISC's requirement of an XML-based metadata document introduces new challenges to Programmers in the pharmaceutical Industry.SAS is used to generate it and this task most likely need more of planning what other Metadata files needs to be required in order to generate define.xml. This paper addresses these challenges. After a brief discussion of XML basics, we will study carefully the structure of Define.xml.

This paper will discuss how the required metadata files are created and after these files in place, we'll see how the SAS can be used as powerful tool for generating the final document with the familiar procedures and statements.

INTRODUCTION:

A data definition file, called Case Report Tabulation Data Definitions (CRT DD), is necessary to facilitate the review of the study data submitted to a regulatory authority. Well-defined, standardized metadata minimizes the time needed to familiarize with the data, which can speed up the review process.

The Case Report Tabulation Data Definition Specification (define.xml) V1.0 Standard as prepared by the CDISC define.xml team has been referenced in the FDA's eCTD Study Data Specifications as the preferred data definition file when data are submitted in the SDTM format. The benefit of define.xml compared to define.pdf, is that an XML file is both human- and machine-readable. Whereas the human-readability helps the reviewer to understand and work with the data, machine-readable metadata can be exploited when transferring data between different systems.

Even the generation of define.xml can be automated, if metadata required for generating define.xml are already in place.

XML BASICS:

A basic understanding of the XML technology is required. There are three different file types that work together in XML technology:

• Schema
Definition and declaration of elements and their attributes file with extension: .XSD
• XML file

Description of data and metadata in machine-readable format using the elements defined in the schema file with extension: .XML
• Style sheet

Definition of the layout in a browser tool for the human readable representation of the XML file, file with extension: .XSL

Figure 1 and 2 Show the Difference between when the XML is opened with out/with XSL file in the same directory.

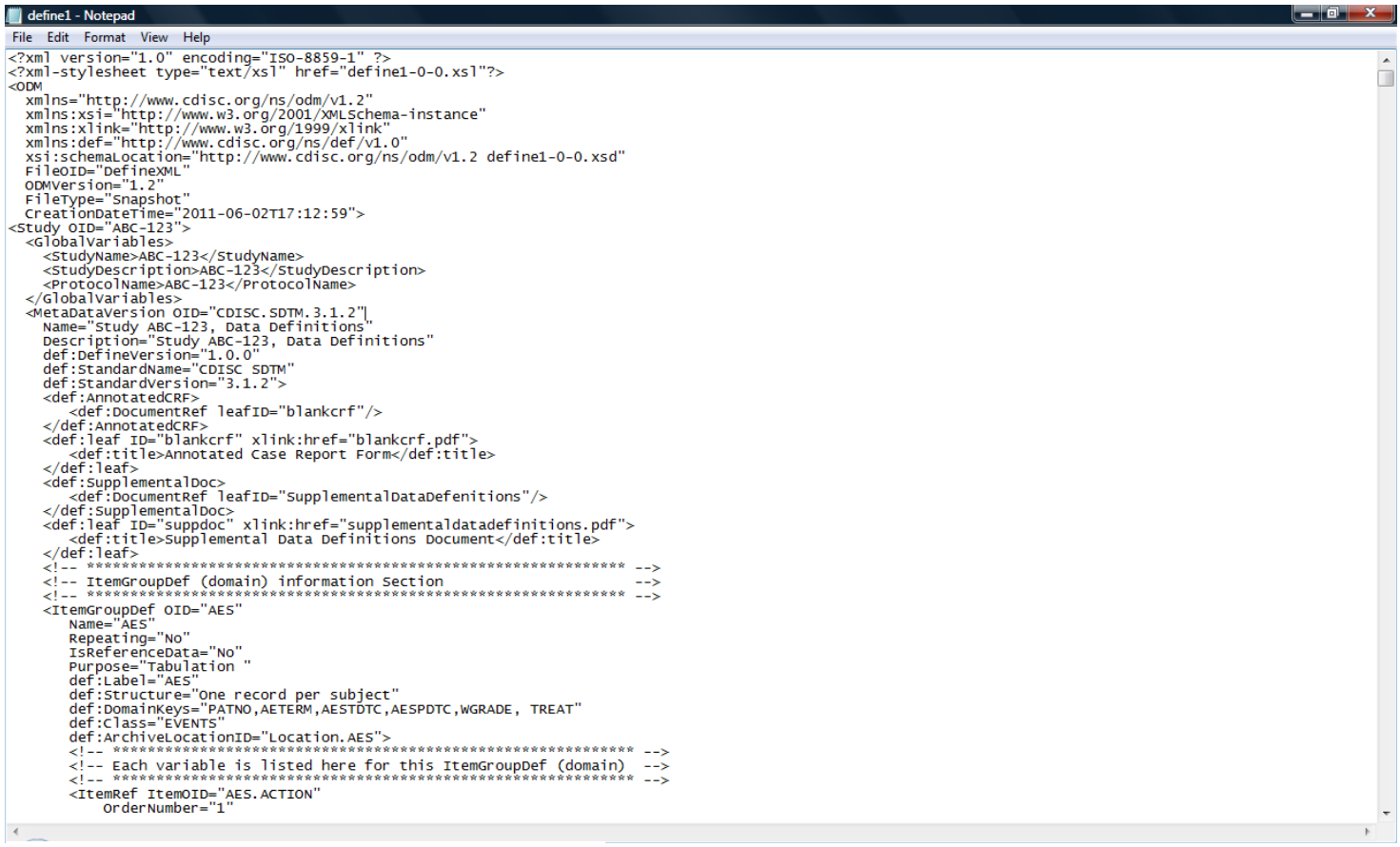


Figure 1: Define.xml opened without using XSL.

When the XSL is Present the Define.XML will look as shown in Figure 2, XSL is very important file while creating an XML Document.

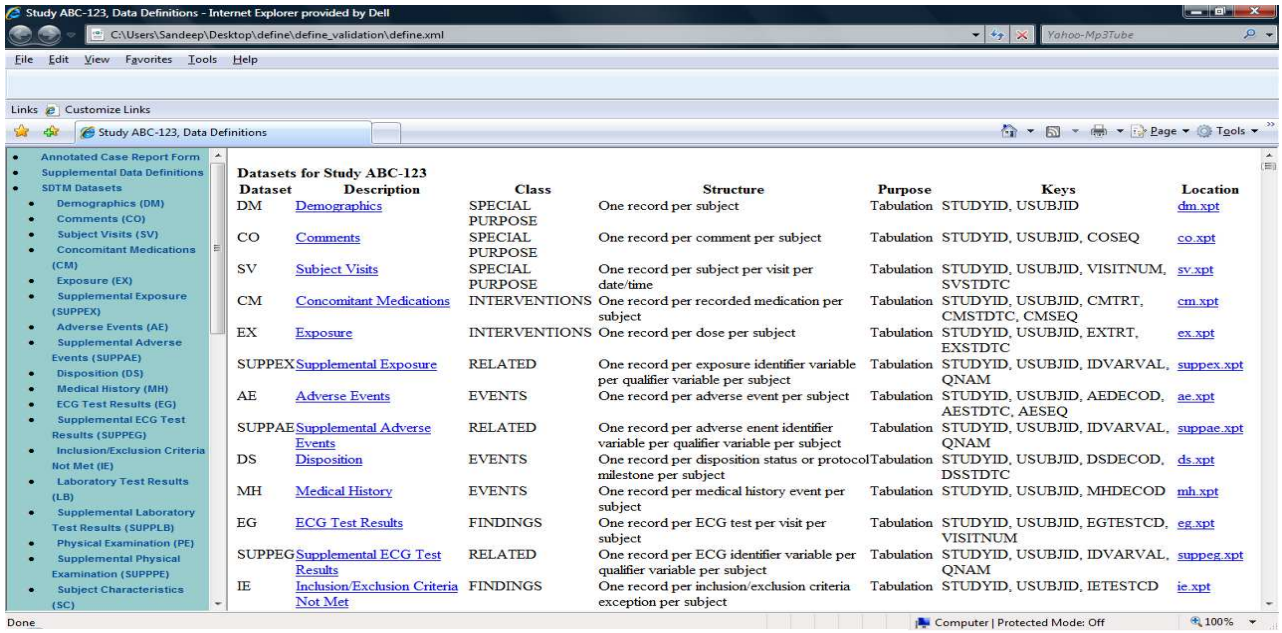


Figure 2 Define.xml Opened when XSL is present in the Same Folder.

Define.XML Metadata:

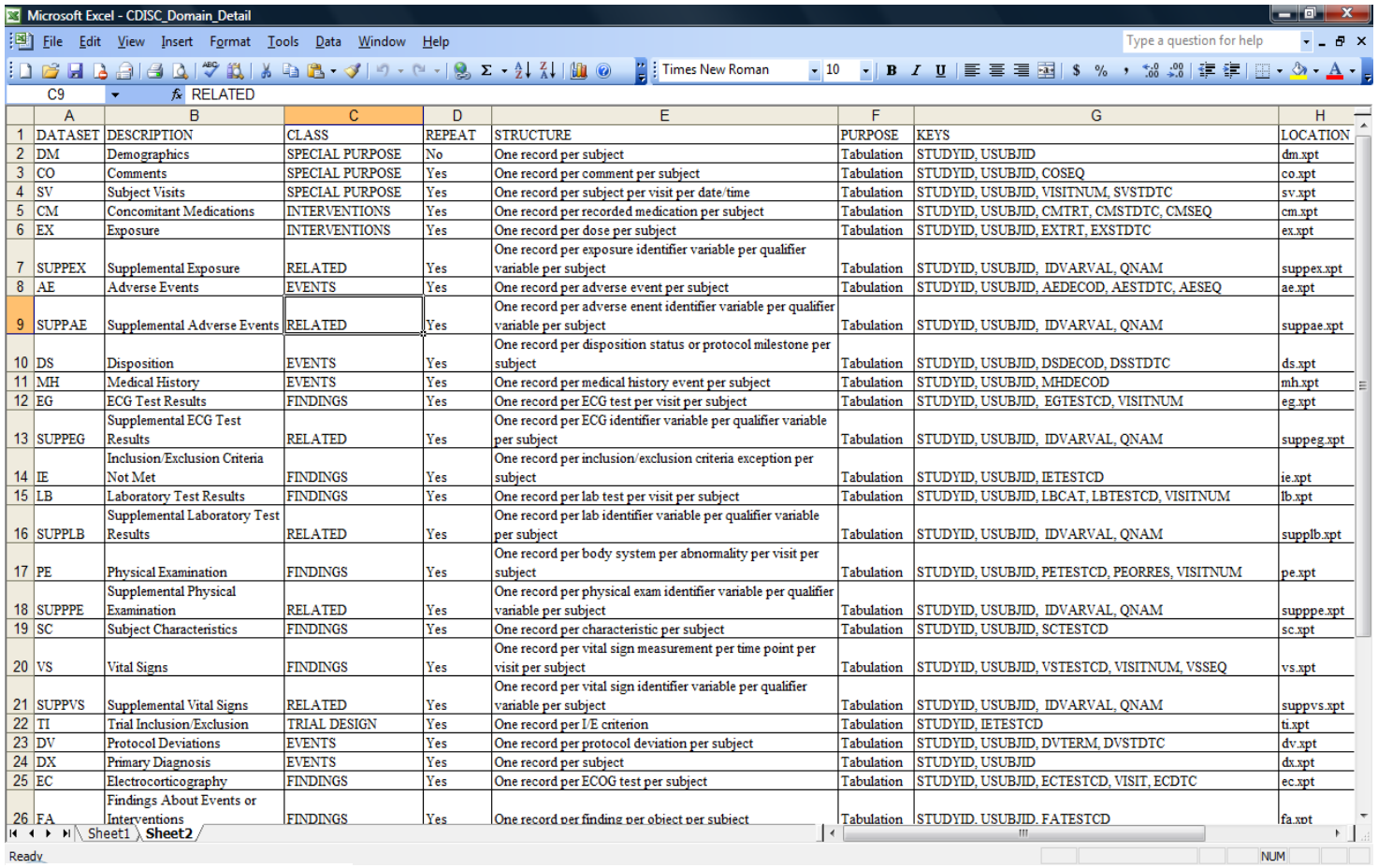
To generate Define.xml we need Several Excel files or worksheets i.e., Data Metadata (List of Datasets), Variable Metadata, Variable Value Level Metadata and Computational Algorithms. Only the Controlled Terminology/Code Lists Section can be created without additional look-up files.

Data Metadata

The Excel file structure chosen to capture the information for the Data Metadata looks very much like the Data Metadata Section of define.xml in a browser tool. It has the following columns:

- Dataset (name)
- Description (label)
- CDISC Domain Class
- Repeat
- Structure
- Purpose (’Tabulation’ for all SDTM datasets)
- Keys
- Location

A list of datasets is often generated at the design and specification phase of a programming project. Figure 3 shows how the file looks like


Microsoft Excel - CDISC_Domain_Detail

File Edit View Insert Format Tools Data Window Help

C9 RELATED

	A	B	C	D	E	F	G	H
1	DATASET	DESCRIPTION	CLASS	REPEAT	STRUCTURE	PURPOSE	KEYS	LOCATION
2	DM	Demographics	SPECIAL PURPOSE	No	One record per subject	Tabulation	STUDYID, USUBJID	dm.xpt
3	CO	Comments	SPECIAL PURPOSE	Yes	One record per comment per subject	Tabulation	STUDYID, USUBJID, COSEQ	co.xpt
4	SV	Subject Visits	SPECIAL PURPOSE	Yes	One record per subject per visit per date/time	Tabulation	STUDYID, USUBJID, VISITNUM, SVSTDTC	sv.xpt
5	CM	Concomitant Medications	INTERVENTIONS	Yes	One record per recorded medication per subject	Tabulation	STUDYID, USUBJID, CMTRT, CMSTDTC, CMSEQ	cm.xpt
6	EX	Exposure	INTERVENTIONS	Yes	One record per dose per subject	Tabulation	STUDYID, USUBJID, EXTRT, EXSTDTC	ex.xpt
7	SUPPEX	Supplemental Exposure	RELATED	Yes	One record per exposure identifier variable per qualifier variable per subject	Tabulation	STUDYID, USUBJID, IDVARVAL, QNAM	suppex.xpt
8	AE	Adverse Events	EVENTS	Yes	One record per adverse event per subject	Tabulation	STUDYID, USUBJID, AEDECOD, AESTDTC, AESEQ	ae.xpt
9	SUPPAE	Supplemental Adverse Events	RELATED	Yes	One record per adverse enent identifier variable per qualifier variable per subject	Tabulation	STUDYID, USUBJID, IDVARVAL, QNAM	suppae.xpt
10	DS	Disposition	EVENTS	Yes	One record per disposition status or protocol milestone per subject	Tabulation	STUDYID, USUBJID, DSDECOD, DSSTDTC	ds.xpt
11	MH	Medical History	EVENTS	Yes	One record per medical history event per subject	Tabulation	STUDYID, USUBJID, MHDECOD	mh.xpt
12	EG	ECG Test Results	FINDINGS	Yes	One record per ECG test per visit per subject	Tabulation	STUDYID, USUBJID, EGTESTCD, VISITNUM	eg.xpt
13	SUPPEG	Supplemental ECG Test Results	RELATED	Yes	One record per ECG identifier variable per qualifier variable per subject	Tabulation	STUDYID, USUBJID, IDVARVAL, QNAM	suppeg.xpt
14	IE	Inclusion/Exclusion Criteria Not Met	FINDINGS	Yes	One record per inclusion/exclusion criteria exception per subject	Tabulation	STUDYID, USUBJID, IETESTCD	ie.xpt
15	LB	Laboratory Test Results	FINDINGS	Yes	One record per lab test per visit per subject	Tabulation	STUDYID, USUBJID, LBCAT, LBTESTCD, VISITNUM	lb.xpt
16	SUPPLB	Supplemental Laboratory Test Results	RELATED	Yes	One record per lab identifier variable per qualifier variable per subject	Tabulation	STUDYID, USUBJID, IDVARVAL, QNAM	supplb.xpt
17	PE	Physical Examination	FINDINGS	Yes	One record per body system per abnormality per visit per subject	Tabulation	STUDYID, USUBJID, PETESTCD, PEORRES, VISITNUM	pe.xpt
18	SUPPPE	Supplemental Physical Examination	RELATED	Yes	One record per physical exam identifier variable per qualifier variable per subject	Tabulation	STUDYID, USUBJID, IDVARVAL, QNAM	supppe.xpt
19	SC	Subject Characteristics	FINDINGS	Yes	One record per characteristic per subject	Tabulation	STUDYID, USUBJID, SCTESTCD	sc.xpt
20	VS	Vital Signs	FINDINGS	Yes	One record per vital sign measurement per time point per visit per subject	Tabulation	STUDYID, USUBJID, VSTESTCD, VISITNUM, VSSEQ	vs.xpt
21	SUPPVS	Supplemental Vital Signs	RELATED	Yes	One record per vital sign identifier variable per qualifier variable per subject	Tabulation	STUDYID, USUBJID, IDVARVAL, QNAM	suppvs.xpt
22	TI	Trial Inclusion/Exclusion	TRIAL DESIGN	Yes	One record per I/E criterion	Tabulation	STUDYID, IETESTCD	ti.xpt
23	DV	Protocol Deviations	EVENTS	Yes	One record per protocol deviation per subject	Tabulation	STUDYID, USUBJID, DVTERM, DVSTDTC	dv.xpt
24	DX	Primary Diagnosis	EVENTS	Yes	One record per subject	Tabulation	STUDYID, USUBJID	dx.xpt
25	EC	Electrocorticography	FINDINGS	Yes	One record per ECOG test per subject	Tabulation	STUDYID, USUBJID, ECTESTCD, VISIT, ECDTC	ec.xpt
26	FA	Findings About Events or Interventions	FINDINGS	Yes	One record per finding per object per subject	Tabulation	STUDYID, USUBJID, FATESTCD	fa.xpt

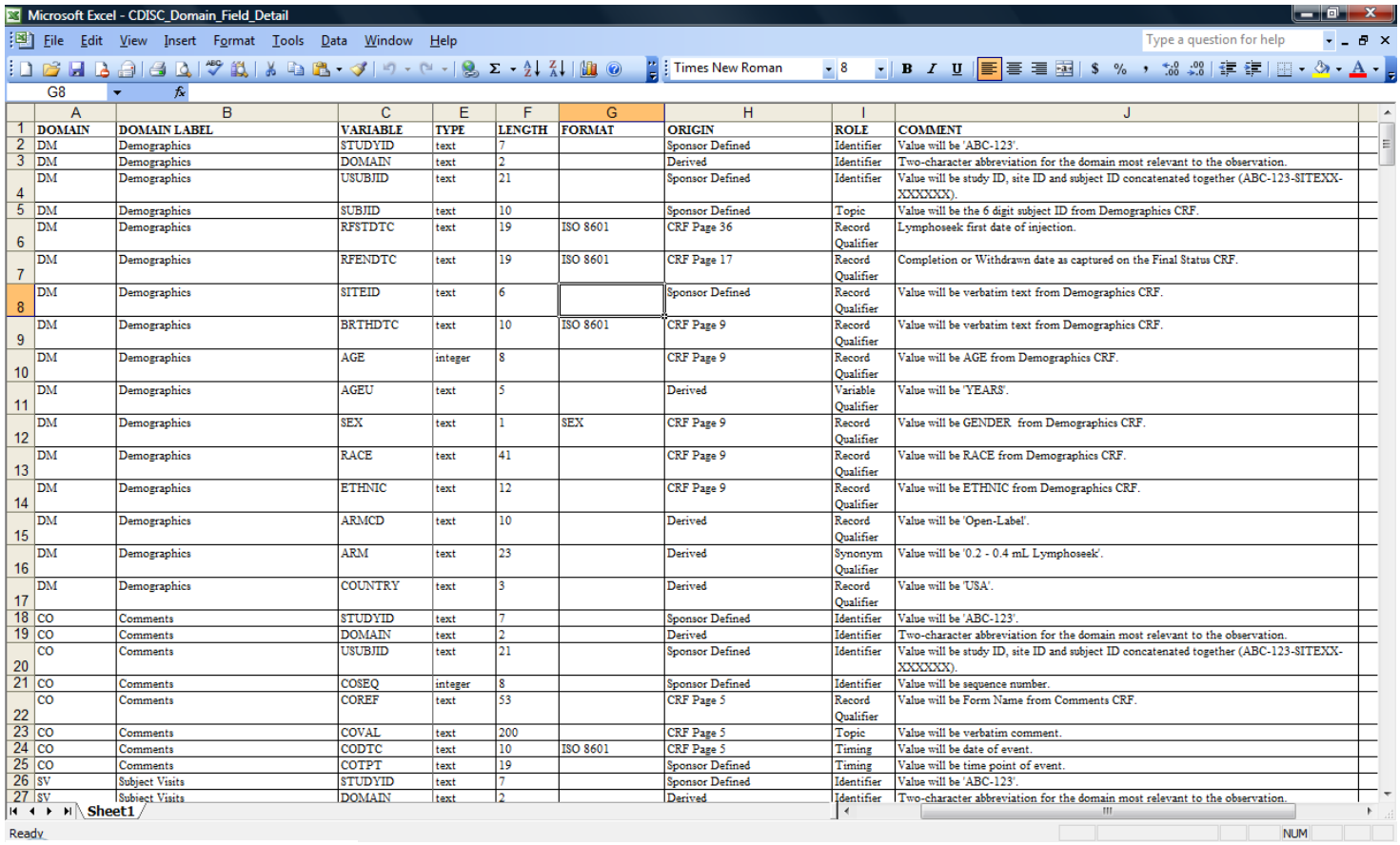
Sheet1 Sheet2

Ready NUM

Figure 3: Data Metadata file.

VARIABLE METADATA

All variables included in a certain dataset are described per dataset in the Variable Metadata Section. Figure 4 shows an example of the Variable Metadata.



	A	B	C	E	F	G	H	I	J
1	DOMAIN	DOMAIN LABEL	VARIABLE	TYPE	LENGTH	FORMAT	ORIGIN	ROLE	COMMENT
2	DM	Demographics	STUDYID	text	7		Sponsor Defined	Identifier	Value will be 'ABC-123'.
3	DM	Demographics	DOMAIN	text	2		Derived	Identifier	Two-character abbreviation for the domain most relevant to the observation.
4	DM	Demographics	USUBJID	text	21		Sponsor Defined	Identifier	Value will be study ID, site ID and subject ID concatenated together (ABC-123-SITEXX-XXXXXX).
5	DM	Demographics	SUBJID	text	10		Sponsor Defined	Topic	Value will be the 6 digit subject ID from Demographics CRF.
6	DM	Demographics	RFSTDTC	text	19	ISO 8601	CRF Page 36	Record Qualifier	Lymphoseek first date of injection.
7	DM	Demographics	RFENDTC	text	19	ISO 8601	CRF Page 17	Record Qualifier	Completion or Withdrawn date as captured on the Final Status CRF.
8	DM	Demographics	SITEID	text	6		Sponsor Defined	Record Qualifier	Value will be verbatim text from Demographics CRF.
9	DM	Demographics	BRTHDTC	text	10	ISO 8601	CRF Page 9	Record Qualifier	Value will be verbatim text from Demographics CRF.
10	DM	Demographics	AGE	integer	8		CRF Page 9	Record Qualifier	Value will be AGE from Demographics CRF.
11	DM	Demographics	AGEU	text	5		Derived	Variable Qualifier	Value will be 'YEARS'.
12	DM	Demographics	SEX	text	1	SEX	CRF Page 9	Record Qualifier	Value will be GENDER from Demographics CRF.
13	DM	Demographics	RACE	text	41		CRF Page 9	Record Qualifier	Value will be RACE from Demographics CRF.
14	DM	Demographics	ETHNIC	text	12		CRF Page 9	Record Qualifier	Value will be ETHNIC from Demographics CRF.
15	DM	Demographics	ARMCD	text	10		Derived	Record Qualifier	Value will be 'Open-Label'.
16	DM	Demographics	ARM	text	23		Derived	Synonym Qualifier	Value will be '0.2 - 0.4 mL Lymphoseek'.
17	DM	Demographics	COUNTRY	text	3		Derived	Record Qualifier	Value will be 'USA'.
18	CO	Comments	STUDYID	text	7		Sponsor Defined	Identifier	Value will be 'ABC-123'.
19	CO	Comments	DOMAIN	text	2		Derived	Identifier	Two-character abbreviation for the domain most relevant to the observation.
20	CO	Comments	USUBJID	text	21		Sponsor Defined	Identifier	Value will be study ID, site ID and subject ID concatenated together (ABC-123-SITEXX-XXXXXX).
21	CO	Comments	COSEQ	integer	8		Sponsor Defined	Identifier	Value will be sequence number.
22	CO	Comments	COREF	text	53		CRF Page 5	Record Qualifier	Value will be Form Name from Comments CRF.
23	CO	Comments	COVAL	text	200		CRF Page 5	Topic	Value will be verbatim comment.
24	CO	Comments	CODTC	text	10	ISO 8601	CRF Page 5	Timing	Value will be date of event.
25	CO	Comments	COTPT	text	19		Sponsor Defined	Timing	Value will be time point of event.
26	SV	Subject Visits	STUDYID	text	7		Sponsor Defined	Identifier	Value will be 'ABC-123'.
27	SV	Subject Visits	DOMAIN	text	2		Derived	Identifier	Two-character abbreviation for the domain most relevant to the observation.

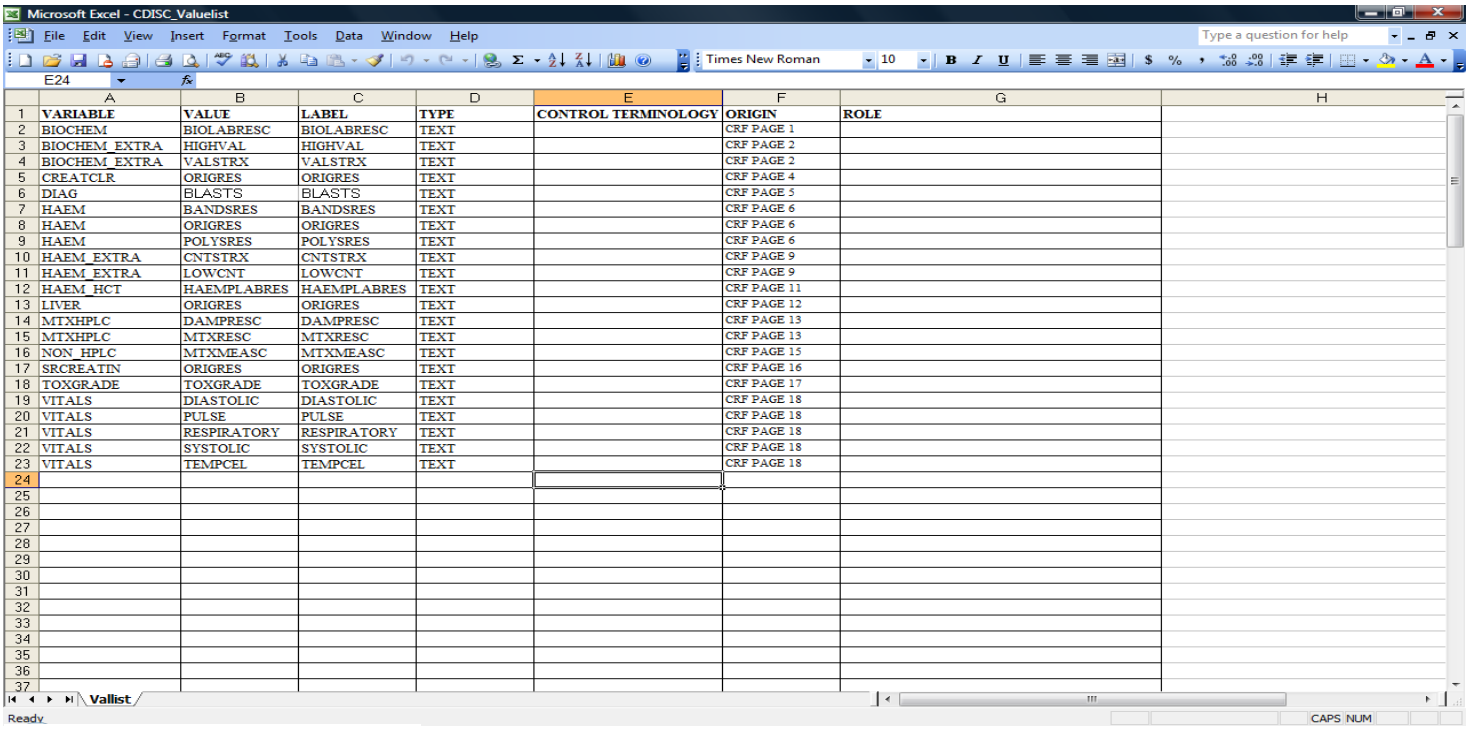
Figure 4: Variable Metadata.
The columns Variable and Label are same as Above Metadata.

- The Type conforms to the data type definitions of the ODM schema. Values are: text, integer, float, date time, date and time.
- For variables with a discrete list of values attached a format name is provided in the Controlled Terms or Format column that links to the Controlled Terms/Code Lists Section.
- The Origin of a variable may be Sponsor Defined, CRF, Assigned, Protocol or Derived
- The Role values correspond to the metadata definition of the SDTM.
- The Comment column contains a short explanation of Sponsor Defined variables or, for derived variables, the derivation rule or a computational method reference.

Value Level Metadata:

The Variable Value Level Metadata describes each unique value of a certain test short name similarly to the variables described in the Variable Metadata. For review and analysis a distinction between each of the single tests is required. Useful for transposing the vertical datasets to horizontal datasets, i.e. variables per test, if required.

Figure 5 shows an Example of Value Level Metadata file.



Microsoft Excel - CDISC_Valuelist

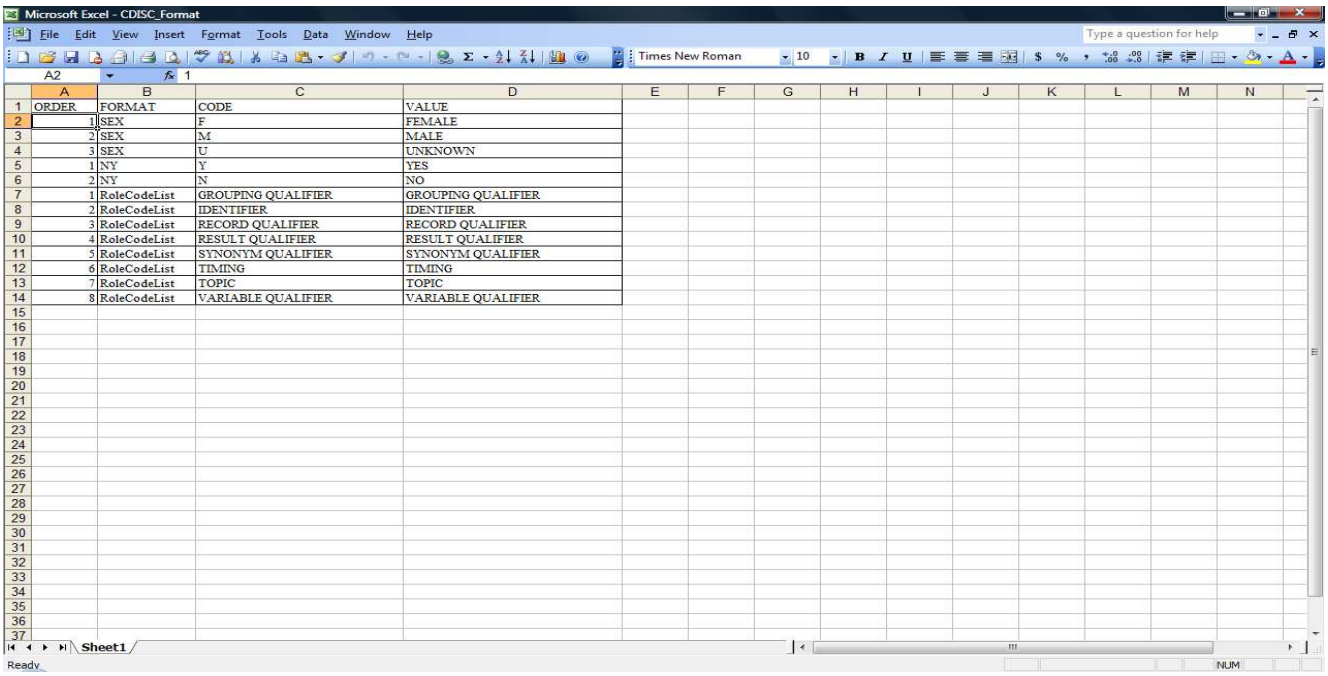
	A	B	C	D	E	F	G	H
1	VARIABLE	VALUE	LABEL	TYPE	CONTROL TERMINOLOGY	ORIGIN	ROLE	
2	BIOCHEM	BIOLABRESC	BIOLABRESC	TEXT		CRF PAGE 1		
3	BIOCHEM_EXTRA	HIGHVAL	HIGHVAL	TEXT		CRF PAGE 2		
4	BIOCHEM_EXTRA	VALSTRX	VALSTRX	TEXT		CRF PAGE 2		
5	CREATCLR	ORIGRES	ORIGRES	TEXT		CRF PAGE 4		
6	DIAG	BLASTS	BLASTS	TEXT		CRF PAGE 5		
7	HAEM	BANDSRES	BANDSRES	TEXT		CRF PAGE 6		
8	HAEM	ORIGRES	ORIGRES	TEXT		CRF PAGE 6		
9	HAEM	POLYSRES	POLYSRES	TEXT		CRF PAGE 6		
10	HAEM_EXTRA	CNTSTRX	CNTSTRX	TEXT		CRF PAGE 9		
11	HAEM_EXTRA	LOWCNT	LOWCNT	TEXT		CRF PAGE 9		
12	HAEM_HCT	HAEMPLABRES	HAEMPLABRES	TEXT		CRF PAGE 11		
13	LIVER	ORIGRES	ORIGRES	TEXT		CRF PAGE 12		
14	MTXHPLC	DAMPRESC	DAMPRESC	TEXT		CRF PAGE 13		
15	MTXHPLC	MTXRESC	MTXRESC	TEXT		CRF PAGE 13		
16	NON_HPLC	MTXMEASC	MTXMEASC	TEXT		CRF PAGE 15		
17	SRCREATIN	ORIGRES	ORIGRES	TEXT		CRF PAGE 16		
18	TOXGRADE	TOXGRADE	TOXGRADE	TEXT		CRF PAGE 17		
19	VITALS	DIASTOLIC	DIASTOLIC	TEXT		CRF PAGE 18		
20	VITALS	PULSE	PULSE	TEXT		CRF PAGE 18		
21	VITALS	RESPIRATORY	RESPIRATORY	TEXT		CRF PAGE 18		
22	VITALS	SYSTOLIC	SYSTOLIC	TEXT		CRF PAGE 18		
23	VITALS	TEMPCEL	TEMPCEL	TEXT		CRF PAGE 18		

Figure 5: Value Level Metadata.

CONTROLLED TERMINOLOGY:

All format names referred to by either the variable metadata or the variable value level metadata, their values and their decodes are displayed in the Controlled Terminology/Code Lists Section. As the code values correspond to the values stored in the datasets, they may be equal to the decode values, e.g., result codes of NEGATIVE or POSITIVE.

Figure 6 Shows an Example how the Control Terminology.



Microsoft Excel - CDISC_Format

	A	B	C	D
1	ORDER	FORMAT	CODE	VALUE
2	1	SEX	F	FEMALE
3	2	SEX	M	MALE
4	3	SEX	U	UNKNOWN
5	1	NY	Y	YES
6	2	NY	N	NO
7	1	RoleCodeList	GROUPING QUALIFIER	GROUPING QUALIFIER
8	2	RoleCodeList	IDENTIFIER	IDENTIFIER
9	3	RoleCodeList	RECORD QUALIFIER	RECORD QUALIFIER
10	4	RoleCodeList	RESULT QUALIFIER	RESULT QUALIFIER
11	5	RoleCodeList	SYNONYM QUALIFIER	SYNONYM QUALIFIER
12	6	RoleCodeList	TIMING	TIMING
13	7	RoleCodeList	TOPIC	TOPIC
14	8	RoleCodeList	VARIABLE QUALIFIER	VARIABLE QUALIFIER

Figure 6 Control Terminology Example

SAS CODE:

The Following code was used to Generate Define.xml with the supporting Documents as mentioned above sections.

```
*******************************************************************************;

*** Program: definexml_map.sas ***;

*** Version: 1.0 ***;

*** Programmer: Uma Sarath Annapareddy Sandeep Kottam, Sree Lakshmi K Tripuraneni ***;

*** Date: 06/04/2011 ***;

*** Purpose: Program used to create the Value list file for Define XML. ***;

*******************************************************************************;

proc datasets lib=work nolist memtype=data kill;
quit;


libname map 'C:\NESUG-2011' access=readonly;
*** Get dataset and variable information;

proc contents data=map._all_ noprint out=directory;
run;

Data directory;
set directory;
       length dataset variable $8 varlabel $40;
       dataset=memname;
       variable=name;
       varlabel=label;
keep dataset variable varlabel length;
run;


* Launch EXCEL and open the EXCEL file of source data;

options noxwait noxsync;
x %letxlname=%bquote ('"C:\NESUG-2011\CDISC_Domain_Detail.xls" ');
 %unquote(&xlname) ;
* Sleep for 5 second to give Excel time to start up;

data _null_;
       x=sleep(1);
run;

* read data;
```

```
filename f_data1 dde "excel|[CDISC_Domain_Detail.xls]sheet2!r2c1:r32c8" notab;

data MAIN;
       length dataset      description class
       repeat structure purpose keys location $100;
       infile f_data1 dlm='09'x dsd missover lrecl=1000;
       input dataset $ description$ class $ repeat $ structure $ purpose $ keys $ location $;

run;

options

noxwait noxsync;
%let xlname=%bquote ('"C:\NESUG-2011\CDISC_Domain_Field_Detail.xls" ');
x %unquote(&xlname) ;
* Sleep for 5 second to give Excel time to start up;

data _null_;
       x=sleep(1);
run;

* read data;

filename f_data2 dde "excel|[CDISC_Domain_Field_Detail.xls]sheet1!r2c1:r350c10" notab;
data MAINDET;
       length domain domainlabel variable label Type format origin role comment $ 100;
       infile f_data2 dlm='09'x dsd missover lrecl=1000;
       input domain $ domainlabel $ variable $ label $ Type $ length $ format $ origin $ role $
comment $;

run;


options noxwait noxsync;
%let xlname=%bquote ('"C:\NESUG-2011\CDISC_Format.xls" ');
x %unquote(&xlname) ;

* Sleep for 5 second to give Excel time to start up;

data _null_;
       x=sleep(1);
run;
* read data;

filename f_data3 dde "excel|[CDISC_Format.xls]sheet1!r2c1:r962c4" notab;
data FMTDET;
       length order format code value $100;
       infile f_data3 dlm='09'x dsd missover lrecl=1000;
       input order $ format $ code $ value $;

run;
options noxwait noxsync;
%let xlname=%bquote ('"C:\NESUG-2011\CDISC_Valuelist.xls" ');
x %unquote(&xlname) ;
* Sleep for 5 second to give Excel time to start up;
```

```
data _null_;
      x=sleep(1);
run;

* read data;

filename f_data4 dde "excel|[CDISC_Valuelist.xls]vallist!r2c1:r23c4" notab;
data VALORG;
      length dataset value order origin $ 100;
      infile f_data4 dlm='09'x dsd missover lrecl=1000;
      input dataset $ value $ order $ origin $;

run;


* Microsoft defines the DDE topic SYSTEM to enable commands to be invoked within Excel;

filename cmds dde 'excel|system';
* Close down EXCEL;

data _null_;

      file cmds;
      put '[QUIT()]';
run;

proc sort data=valorg;
      by dataset value;
run;

data main;
      length dataset $8;
      retain dsnord;
      set main;
      if dsnord eq . then dsnord=1;
      else dsnord=dsnord+1;
      format dataset;
      purpose=left(trim(purpose));

      keys=left(trim(tranwrd(keys,', ',', ')));
run;

proc sort data=main;
      by dsnord dataset;
run;

data maindet;
      length dataset variable $8;
      set maindet(rename=domain=dataset);
      comment=tranwrd(comment,'"','''');
      role=upcase(role);

      origin=upcase(origin);
```

```
      format dataset variable;
run;

proc sort data=maindet;
      by dataset;
run;

data maindet;
      set maindet;
      by dataset;
      retain fldord;
      if first.dataset then fldord=1;
      else fldord=fldord+1;
run;

proc sort data=maindet;
      by dataset variable;
run;

data page1;
      set main;
      by dsnord dataset;
      keep dataset dsnord;
run;

proc sort data=page1;
      by dataset;
run;

proc sort data=directory(rename=(varlabel=label));
      by dataset variable;
run;

data maindet;
      merge maindet(in=a) directory(in=b);
      by dataset variable;
      if a and b;
run;

data maindet;
      merge page1(in=a) maindet(in=b);
      by dataset;
      if a and b;
      length newvar $30;
      if variable in ('STUDYID' 'DOMAIN' 'USUBJID') then newvar=variable;
      else newvar=left(trim(dataset)) || '.' || left(trim(variable));
run;

proc sort data=maindet;
      by dsnord dataset fldord variable;
run;

data temp1;

      set maindet;
```

```
      length valnum valchar $10;
      if index(variable,'STRESN')>0;

      valnum=variable;

      valchar=left(trim(dataset)) ||'ORRES';
      keep dsnord dataset valnum valchar;
run;

data mainval;
      set maindet;
      length vallabl $10;
      by dsnord dataset fldord variable;
      if index(variable,'TESTCD')>0 or variable='QNAM';
      if variable='QNAM' then vallabl='QLABEL';
      else if dataset='TI' then vallabl='IETEST';
      else vallabl=left(trim(dataset)) || 'TEST';
      keep dsnord dataset variable vallabl;
run;

data mainval(drop=dsnord);
      merge mainval(in=a) temp1(in=b);
      by dsnord dataset;
      if a;
      if valnum ne '' then valfntp=1;
run;

data temp1;
      set mainval;
      valcode=1;
      keep dataset variable valcode;
run;

proc sort data=temp1;
      by dataset variable;
run;

proc sort data=maindet;
      by dataset variable;
run;

data maindet;
      merge maindet(in=a) temp1(in=b);
      by dataset variable;
      if a;
run;

proc sort data=maindet;
      by dsnord dataset fldord variable;
run;

*** XML Generation Steps ***;

filename outfile 'C:\NESUG-2011\define1.xml';
```

```
data _null_;
file outfile notitles lrecl=2000;
put '<?xml version="1.0" encoding="ISO-8859-1" ?>';
put '<?xml-stylesheet type="text/xsl" href="define1-0-0.xsl"?>';
put '<ODM';
put ' xmlns="http://www.cdisc.org/ns/odm/v1.2"';
put ' xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"';
put ' xmlns:xlink="http://www.w3.org/1999/xlink"';
put ' xmlns:def="http://www.cdisc.org/ns/def/v1.0"';
put ' xsi:schemaLocation="http://www.cdisc.org/ns/odm/v1.2 define1-0-0.xsd"';
put ' FileOID="DefineXML"';
put ' ODMVersion="1.2"';
put ' FileType="Snapshot"';
       d=date();

       format d yymmdd10.;
       t=time();

       format t time8.;
       call symput('tday',put(d,yymmdd10.) || 'T' || put(t,time8.));
run;
```

```
data _null_;

file outfile notitles lrecl=2000 mod;
put ' CreationDateTime="' "&tday" '">';
put '<Study OID="ABC-123">';
put ' <GlobalVariables>';
put ' <StudyName>ABC-123</StudyName>';
put ' <StudyDescription>ABC-123</StudyDescription>';
put ' <ProtocolName>ABC-123</ProtocolName>';
put ' </GlobalVariables>';
put ' <MetaDataVersion OID="CDISC.SDTM.3.1.2"';
put ' Name="Study ABC-123, Data Definitions"';
put ' Description="Study ABC-123, Data Definitions"';
put ' def:DefineVersion="1.0.0"';
put ' def:StandardName="CDISC SDTM"';
put ' def:StandardVersion="3.1.2">';
put ' <def:AnnotatedCRF>';
put ' <def:DocumentRef leafID="blankcrf"/>';
put ' </def:AnnotatedCRF>';
put ' <def:leaf ID="blankcrf" xlink:href="blankcrf.pdf">';
put ' <def:title>Annotated Case Report Form</def:title>';
put ' </def:leaf>';
put ' <def:SupplementalDoc>';
put ' <def:DocumentRef leafID="SupplementalDataDefenitions"/>';
put ' </def:SupplementalDoc>';
```

```
put ' <def:leaf ID="suppdoc" xlink:href="supplementaldatadefinitions.pdf">';
put ' <def:title>Supplemental Data Definitions Document</def:title>';
put ' </def:leaf>';
run

;
data vallist;
       length label origin valcomm $200 dataset variable value $8 type $10;
       label='';
       order=.;
       dataset='';
       variable='';
       value='';
       type='';
       origin='';
       valcomm='';
       if label ne '';
run;

%macro getval(dset=,fld=,ren=,kpvar=,findtyp=);

data &dset;
       set map.&dset;
       length dataset variable $8 type $10;
       if &fld ne'';
       rename &ren;
       dataset="&dset";
       variable="&fld";
       type='text';
       keep dataset variable type &kpvar;

run;

%if &findtyp=1 %then %do;
       data &dset char int float;
       set &dset;

       if &dset.stresn ne. then do;
       if &dset.stresn eq int(&dset.stresn) then output int;

       else output float;

       end;

       else if &dset.orres ne'' then output char;
       output &dset;

       keep dataset variable type value label;

       run;

       proc sort data=&dset nodup;

       by variable value label;
```

```
        run;

        proc sort data=char nodup;

        by variable value label;

        run;

        proc sort data=int nodup;

        by variable value label;

        run;

        proc sort data=float nodup;

        by variable value label;

        run;

        data &dset;

        merge &dset(in=a) char(in=b) int(in=c) float(in=d);

        by variable value label;

        length type $10;
        if a and b then type='text';
        else if a and d then type='float';
        else if a and c then type='integer';
        else if a then type='text';
        run;

%end;


proc sort data=&dset;

        by dataset value;

run;

data &dset;

        merge &dset(in=a) valorg(in=b);

        by dataset value;

        if a;

run;

proc sort data=&dset nodup;

        by variable order value label;
```

```
run;

proc append base=vallist data=&dset;

run;

data _null_;

      file outfile notitles lrecl=2000 mod;
      put' <def:ValueListDef OID="ValueList.' "&dset..&fld" '">';
run;

data _null_;

      file outfile notitles lrecl=2000 mod;
      set &dset;

      by variable order value label;

      put' <ItemRef ItemOID="' "&dset..&fld.." value +(-1) '" OrderNumber="1" Mandatory="No"/>';
run;

data _null_;

      file outfile notitles lrecl=2000 mod;
      put' </def:ValueListDef>';
run;

%mend;

data _null_;
      set mainval;
      call execute('%getval(dset=' || dataset || ',fld=' || variable || ',ren=' || variable ||
'=value ' ||
      vallabl ||'=label,kpvar=' || variable || ' ' || vallabl || ' ' || valchar || ' ' || valnum ||
',findtyp=' || valfntp || ')');
run;



proc   sort data=vallist;
      by dataset variable value;
run;

data vallist;

      merge page1(in=a) vallist(in=b);
      by dataset;
      if a and b;
      length newvar $30;
      if variable in ('STUDYID' 'DOMAIN' 'USUBJID') then newvar=variable;
      else newvar=left(trim(dataset)) || '.' || left(trim(variable)) || '.' || left(trim(value));
run;
```

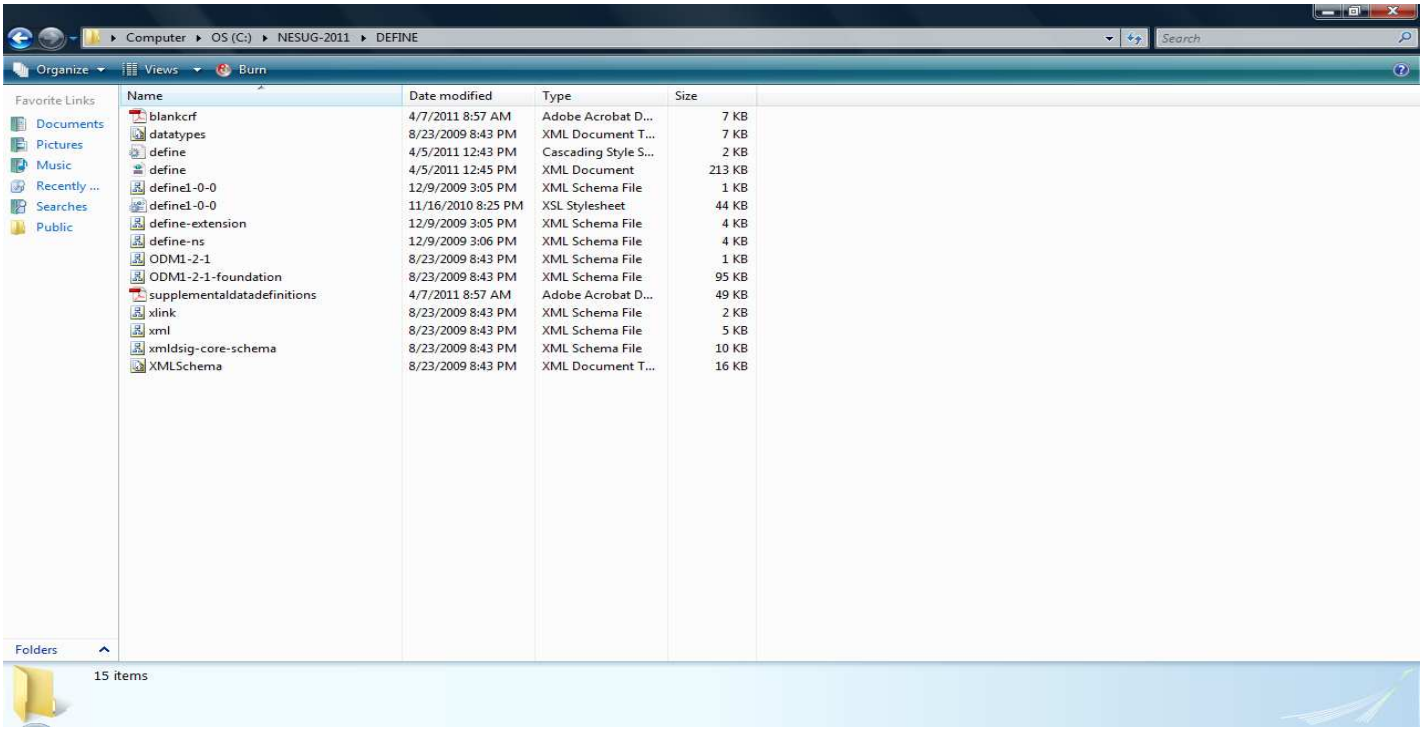
```
proc sort data=vallist;
      by dsnord dataset variable order value;
run;

data _null_;

file outfile notitles lrecl=2000 mod;
put ' <!-- ************************************************************ -->';
put ' <!-- ItemGroupDef (domain) information Section -->';
put ' <!-- ************************************************************ -->';
run;

%macro main;
      data _null_;

      file outfile notitles lrecl=2000 mod;
      set tset;

      put' <ItemGroupDef OID="' dataset +(-1) '"';
      put' Name="' dataset +(-1) '"';
      put' Repeating="' repeat +(-1) '"';
      put' IsReferenceData="No"';
      put' Purpose="' purpose +(-1) '"';
      put' def:Label="' description +(-1) '"';
      put' def:Structure="' structure +(-1) '"';
      put' def:DomainKeys="' keys +(-1) '"';
      put' def:Class="' class +(-1) '"';
      put' def:ArchiveLocationID="Location.' dataset +(-1) '">';
      put' <!-- ************************************************************ -->';
      put' <!-- Each variable is listed here for this ItemGroupDef (domain) -->';
      put' <!-- ************************************************************ -->';
run;

%mend;

%macro maindet;


data _null_;

      file outfile notitles lrecl=2000 mod;
      set tsetdet;

      put' <ItemRef ItemOID="' newvar +(-1) '"';
      put' OrderNumber="' fldord +(-1) '"';
      put' Mandatory="Yes"';
      put' Role="' role +(-1) '"';
      put' RoleCodeListOID="RoleCodeList"/>';
run;

%mend;

proc sql noprint;
      select max(dsnord) into :maxdsn from main;
quit;
```

```
%macro meta;

%doi=1 %to &maxdsn;

data tset;
       set main;

       by dsnord dataset;

       if dsnord=&i;

run;

%main;

data tsetdet;

       set maindet;

       by dsnord dataset fldord variable;

       if dsnord=&i;

run;

%maindet;

data _null_;

       file outfile notitles lrecl=2000 mod;
       set tset;

       put' <!-- ************************************************************ -->';
       put' <!-- def:leaf details for hypertext linking the dataset -->';
       put' <!-- ************************************************************ -->';
       put' <def:leaf ID="Location.' dataset +(-1) '" xlink:href="' location +(-1) '">';
       put' <def:title>' location +(-1) '</def:title>';
       put' </def:leaf>';
       put' </ItemGroupDef>';
run;

%end;

%mend;

%meta;

data _null_;

file outfile notitles lrecl=2000 mod;
put ' <ItemDef OID="STUDYID"';
put ' Name="STUDYID"';
put ' DataType="text"';
put ' Length="17"';
```

```
put ' Origin="SPONSOR DEFINED"';
put ' def:Label="Study Identifier">';
put ' </ItemDef>';
run;

data _null_;

file outfile notitles lrecl=2000 mod;

       set maindet;
             by dsnord dataset fldord variable;
       if newvar ne 'STUDYID';
       put ' <ItemDef OID="' newvar +(-1) '"';
       put ' Name="' variable +(-1) '"';
       put ' DataType="' type +(-1) '"';
             put ' Length="' length +(-1) '"';
       put ' Origin="' origin +(-1) '"';
       if comment ne '' then do;
       put ' Comment="' comment +(-1) '"';
       end;

       put ' def:Label="' label +(-1) '">';
       if format ne '' then do;
       put ' <CodeListRef CodeListOID="' format +(-1) '"/>';
       end;
       if valcode eq 1 then do;
       put ' <def:ValueListRef ValueListOID="ValueList.' newvar +(-1) '"/>';
       end;
       put ' </ItemDef>';
run;

data _null_;

file outfile notitles lrecl=2000 mod;
set vallist;
       by dsnord dataset variable order value;
       put ' <ItemDef OID="' newvar +(-1) '"';
       put ' Name="' value +(-1) '"';
       put ' DataType="' type +(-1) '"';
       put ' Origin="' origin '"';
       if valcomm ne '' then do;
       put ' Comment="' valcomm +(-1) '"';
end;
put ' def:Label="' label +(-1) '">';
put ' </ItemDef>';
run;
proc sort data=fmtdet;
by format order;
run;
data _null_;
       file outfile notitles lrecl=2000 mod;
       put ' <!-- ************************************************************ -->';
       put ' <!-- The Codelist details are here for all domains -->';
       put ' <!-- ************************************************************ -->';
run;
```

```
data _null_;
file outfile notitles lrecl=2000 mod;
       set fmtdet;
       by format order;
       if first.format then do;
       put ' <CodeList OID="' format +(-1) '" Name="' format +(-1) '" DataType="text">';
       end;
       put ' <CodeListItem CodedValue="' code +(-1) '">';
       put ' <Decode>';
       put ' <TranslatedText xml:lang="en">' value +(-1) '</TranslatedText>';
       put ' </Decode>';
       put ' </CodeListItem>';
       if last.format then do;
       put ' </CodeList>';
       end;
run;

data _null_;
       file outfile notitles lrecl=2000 mod;
       put ' </MetaDataVersion>';
       put ' </Study>';
       put '</ODM>';
run;
```

OUTPUT:



The Define.xml is stored in the above mentioned folder with blankcrf and supplementaldatadefinitions pdf which were linked in the program mentioned above.

CONCLUSIONS:

The above method, this compare macro is designed to make your life easier. You can use this code to use it over and over again. We can also Automate the process by passing excel file name and path or Use PIPE to read Excel Sheets in the particular Directory.

REFERENCES:

CDISC Case Report Tabulation Data Definition Specification (define.xml), Version 1.0, February 9, 2005 (http://www.cdisc.org/models/def/v1.0/index.html)

CDISC Metadata Submission Guidelines, Appendix to the SDTM IG V3.1.1, Draft Version 0.9, July 25, 2007 (http://www.cdisc.org/models/sdtm/v1.1/index.html)

CDISC Study Data Tabulation Model (SDTM) Implementation Guide, Final Version 3.1.1, September 8, 2005 (http://www.cdisc.org/models/sdtm/v1.1/index.html)

CDISC Operational Data Model (ODM), Version 1.3 (http://www.cdisc.org/models/odm/v1.3/index.html)

ACKNOWLEDGMENTS:

CONTACT INFORMATION:

Your comments and questions are valued and encouraged. Contact the authors at:

Uma Sarath Annapareddy
Sr.Statistical Programmer Consultant,
Independent Consultant,
Woodbridge, NJ
908-548-3187
sarathannapareddy@yahoo.com
Sandeep Kottam
Sr.Statistical/SDTM Programming consultant,
605-691-3274
kottamsandeep@gmail.com
Sree Lakhsmi K Tripuraneni
Sr.Statistical programmer,
Purdue Pharma,
Stamford, CT-06905
605-691-3312
sreektripuraneni@gmail.com