Accessing and using the metadata from the define.xml

Lex Jansen
Octagon Research Solutions, Inc.

Leading the Electronic Transformation of Clinical R&D

PharmaSUG 2010, Orlando, FL
• What is XML
• Reading XML with SAS (*exercises*)
• What is the define.xml
• Why convert the define.xml to SAS ???
  – printing issue
  – Validating the define.xml
  – Other use cases
• define.xml as a relational data model
• SAS XML mapper
• Converting define.xml to SAS datasets (*exercises*)
What is XML
• **XML** stands for eXtensible Markup Language

• **XML** is a markup language much like HTML, with tags, elements and attributes

• **XML** was designed to transport and store data, not to display data (display and content separated)

• **XML** does not DO anything

• **XML** tags are not predefined. You must define your own (self-descriptive!) tags
• Like HTML, XML makes use of **tags** (words bracketed by '<' and '>') and **attributes** (of the form name="value")

• **BUT ... , HTML** specifies what each tag and attribute means, and often how the text between them will look in a browser

• **XML** uses the tags only to delimit pieces of data, and leaves the interpretation of the data completely to the application that reads it

• In other words, if you see "<p>" in an XML file, do not assume it is a paragraph. Depending on the context, it may be a price, a parameter, a person, (or maybe something that does not start with with a "p"?)
• An XML file is **well-formed** if it conforms to the rules of XML syntax
  – A single element (root element) contains all other elements in the document (define.xml : <ODM> )
  – Elements have to be properly opened and closed
  – Elements do not overlap, e.g. properly nested
  – Attributes are properly quoted
  – The document does not contain illegal characters. Example: if the left opening bracket “<“ is part of the content it should be substituted as “&lt;”
  – A conforming XML parser is not allowed to process an XML document that is not well-formed
• Predefined entities:

<table>
<thead>
<tr>
<th>Character</th>
<th>Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
<td>&amp;</td>
</tr>
<tr>
<td>&lt;</td>
<td>&lt;</td>
</tr>
<tr>
<td>&gt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>'</td>
<td>'</td>
</tr>
</tbody>
</table>

Entities &amp; and &lt; MUST be used within text value of an element or attribute
• An XML file is **valid** if it conforms to a specific **XML schema**
• An **XML schema** is a description of a type of XML document
• A Schema defines constraints on the structure and contents of documents of that type
• A schema might describe that the content of a certain element, that contains a datetime value, is only valid if the value conforms to the ISO8601 standard
SAS can read generic XML files, that have the following characteristics:

- The enclosing root element is comparable to a SAS library
- A second-level element is translated to a data set name
- Other elements within that second level become SAS variables
Reading XML with SAS
Reading XML files with SAS

The example XML file we use contains data from "THE HEART OF ROCK & SOUL, the 1001 Greatest Singles Ever Made" by Dave Marsh.

See also: http://www.lexjansen.com/marsh
A “rectangular” XML file

```xml
<?xml version="1.0" encoding="ISO-8859-1" ?>
<heartofrockandsoul>
  <entry>
    <rank>556</rank>
    <artist>The Sheppards</artist>
    <title>Tragic</title>
    <producer>Bunky Sheppard</producer>
    <writer>Kermit Chandler &amp; O.C. Perkins</writer>
    <label>Apex 7762</label>
    <year>1961</year>
    <billboard>Did not make pop charts</billboard>
  </entry>
  <entry>
    <rank>938</rank>
    <artist>Gino Washington</artist>
    <title>Gino is a Coward</title>
    <producer>Sonny Saunders</producer>
    <writer>Ronald Davis</writer>
    <label>Ric Tic</label>
    <year>1963</year>
    <billboard>Did not make pop charts</billboard>
  </entry>
</heartofrockandsoul>
```
* Exercise01-A_ReadXML.sas ;

FILENAME rocksoul "&WorkShop\xml\heartofrocknsoul-1.xml";
LIBNAME rs xml XMLFILEREF=rocksoul;

PROC CONTENTS DATA=rs._ALL_ VARNUM;
RUN;

DATA work.RocknSoul1;
  SET rs.entry;
RUN;

PROC CONTENTS DATA=work.RocknSoul1 VARNUM;
RUN;

PROC PRINT DATA=work.RocknSoul1;
RUN;
### The CONTENTS Procedure

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>Variables</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS.ENTRY</td>
<td>.</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Member Type</th>
<th>Indexes</th>
<th>Observation Length</th>
<th>Deleted Observations</th>
<th>Compressed</th>
<th>Sorted</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>0</td>
<td>0</td>
<td>NO</td>
<td>NO</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Engine</th>
<th>Protection</th>
<th>Data Set Type</th>
<th>Label</th>
<th>Data Representation</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML</td>
<td></td>
<td></td>
<td></td>
<td>Default</td>
<td>Default</td>
</tr>
</tbody>
</table>

### Variables in Creation Order

<table>
<thead>
<tr>
<th>#</th>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Format</th>
<th>Informat</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BILLBOARD</td>
<td>Char</td>
<td>23</td>
<td>$23.</td>
<td>$23.</td>
<td>BILLBOARD</td>
</tr>
<tr>
<td>2</td>
<td>YEAR</td>
<td>Num</td>
<td>8</td>
<td>F8.</td>
<td>F8.</td>
<td>YEAR</td>
</tr>
<tr>
<td>3</td>
<td>LABEL</td>
<td>Char</td>
<td>14</td>
<td>$14.</td>
<td>$14.</td>
<td>LABEL</td>
</tr>
<tr>
<td>4</td>
<td>WRITER</td>
<td>Char</td>
<td>44</td>
<td>$44.</td>
<td>$44.</td>
<td>WRITER</td>
</tr>
<tr>
<td>5</td>
<td>PRODUCER</td>
<td>Char</td>
<td>29</td>
<td>$29.</td>
<td>$29.</td>
<td>PRODUCER</td>
</tr>
<tr>
<td>6</td>
<td>TITLE</td>
<td>Char</td>
<td>36</td>
<td>$36.</td>
<td>$36.</td>
<td>TITLE</td>
</tr>
<tr>
<td>7</td>
<td>ARTIST</td>
<td>Char</td>
<td>28</td>
<td>$28.</td>
<td>$28.</td>
<td>ARTIST</td>
</tr>
<tr>
<td>8</td>
<td>RANK</td>
<td>Num</td>
<td>8</td>
<td>F8.</td>
<td>F8.</td>
<td>RANK</td>
</tr>
</tbody>
</table>
&amp; transformed to &

<entry>
  <rank>556</rank>
  <artist>The Sheppards</artist>
  <title>Tragic</title>
  <producer>Bunky Sheppard</producer>
  <writer>Kermit Chandler & O.C. Perkins</writer>
  <label>Apex 7762</label>
  <year>1961</year>
  <billboard>Did not make pop charts</billboard>
</entry>
Reading XML files with SAS

data set

variables

```xml
< xml version="1.0" encoding="ISO-8859-1" >
  < heartofrockandsoul >
    < entry >
      < rank > 556 </ rank >
      < artist > The Sheppards </ artist >
      < title > Tragic </ title >
      < producer > Bunky Sheppard </ producer >
      < writer > Kermit Chandler & O.C. Perkins </ writer >
      < label > Apex 7762 </ label >
      < year > 1961 </ year >
      < billboard > Did not make pop charts </ billboard >
    </ entry >
    < entry >
      < rank > 938 </ rank >
      < artist > Gino Washington </ artist >
      < title > Gino is a Coward </ title >
      < producer > Sonny Saunders </ producer >
      < writer > Ronald Davis </ writer >
      < label > Ric Tic </ label >
      < year > 1963 </ year >
      < billboard > Did not make pop charts </ billboard >
    </ entry >
  </ heartofrockandsoul >
</xml>
* Exercise01-B_ReadXML.sas ;

FILENAME rocksoul "&WorkShop/xml/heartofrocknsoul-2.xml";
LIBNAME rs xml XMLFILEREF=rocksoul;

DATA work.RocknSoul2;
   SET rs.entry;
RUN;

PROC PRINT DATA=work.RocknSoul2;
RUN;
* Exercise01-B_ReadXML.sas;

FILENAME rocksoul "&WorkShop\xml\heartofrocknsoul-2.xml";
LIBNAME rs xml XMLFILEREF=rocksoul;
NOTE: Libref RS was successfully assigned as follows:
    Engine:       XML
    Physical Name: 

DATA work.RocknSoul2;
SET rs.entry;
ERROR: There is an illegal character in the entity name.
    encountered during XMLMap parsing
    occurred at or near line 8, column 31
ERROR: XML describe error: Internal processing error.

RUN;

NOTE: The SAS System stopped processing this step because of err
WARNING: The data set WORK.ROCKNSOUL2 may be incomplete. When t
    observations and 0 variables.
NOTE: DATA statement used (Total process time):
    real time          0.03 seconds
    cpu time           0.01 seconds
Characters like the ampersand (&) and the left angle bracket (<) must be escaped: \&amp; and \&lt;

To import an XML document that contains non-escaped characters, you can specify the XMLPROCESS=RELAX option on the LIBNAME.

Note: **This is not recommended.** If an XML document consists of non-escaped characters, the content is not well-formed.

A conforming XML parser is not allowed to process an XML document that is not well-formed.
* Exercise01-C_ReadXML.sas ;

FILENAME rocksoul "&WorkShop\xml\heartofrocknsoul-2.xml";
LIBNAME rs xml XMLFILEREF=rocksoul XMLPROCESS=RELAX;

DATA work.RocknSoul2;
  SET rs.entry;
RUN;

PROC PRINT DATA=work.RocknSoul2;
RUN;
* Exercise01-C_ReadXML.sas;

FILENAME rocksoul "WorkShop\xml\heartofrocknSoul-2.xml";
LIBNAME rs xml XMLFILEREF=rocksoul XMLPROCESS=RELAX;

NOTE: Libref RS was successfully assigned as follows:
    Engine:      XML
    Physical Name:

DATA work.RocknSoul2;
SET rs.entry;
RUN;

NOTE: There were 16 observations read from the data set RS.ENTRY.
NOTE: The data set WORK.ROCKNSOUL2 has 16 observations and 8 variables.
NOTE: DATA statement used (Total process time):
    real time 0.09 seconds
    cpu time 0.07 seconds

PROC PRINT DATA=work.RocknSoul2;
RUN;

NOTE: There were 16 observations read from the data set WORK.ROCKNSOUL2.
NOTE: PROCEDURE PRINT used (Total process time):
    real time 0.00 seconds
    cpu time 0.00 seconds
Let’s make rank an attribute

```xml
<entry rank="556">
  <artist>The Sheppards</artist>
  <title>Tragic</title>
  <producer>Bunky Sheppard</producer>
  <writer>Kermit Chandler & O.C. Perkins</writer>
  <label>Apex 7762</label>
  <year>1961</year>
  <billboard>Did not make pop charts</billboard>
</entry>
```
* Exercise01-D_ReadXML.sas;

FILENAME rocksoul "&WorkShop\xml\heartofrocknsoul-3.xml";
LIBNAME rs xml XMLFILEREF=rocksoul;

PROC CONTENTS DATA=rs._ALL_ VARNUM;
RUN;

DATA work.RocknSoul3;
   SET rs.entry;
RUN;

PROC CONTENTS DATA=work.RocknSoul3 VARNUM;
RUN;

PROC PRINT DATA=work.RocknSoul3;
RUN;
The CONTENTS Procedure

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Set Name</td>
<td>RS.ENTRY</td>
<td>Observations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Member Type</td>
<td>DATA</td>
<td>Variables</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Engine</td>
<td>XML</td>
<td>Indexes</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Created</td>
<td></td>
<td>Observation</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Last Modified</td>
<td></td>
<td>Length</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protection</td>
<td></td>
<td>Deleted</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Data Set Type</td>
<td></td>
<td>Observations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Label</td>
<td></td>
<td>Compressed</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Data Representation</td>
<td>Default</td>
<td>Sorted</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Encoding</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Variables in Creation Order

<table>
<thead>
<tr>
<th>#</th>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Format</th>
<th>Informat</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BILLBOARD</td>
<td>Char</td>
<td>23</td>
<td>$23.</td>
<td>$23.</td>
<td>BILLBOARD</td>
</tr>
<tr>
<td>2</td>
<td>YEAR</td>
<td>Num</td>
<td>8</td>
<td>F8.</td>
<td>F8.</td>
<td>YEAR</td>
</tr>
<tr>
<td>3</td>
<td>LABEL</td>
<td>Char</td>
<td>14</td>
<td>$14.</td>
<td>$14.</td>
<td>LABEL</td>
</tr>
<tr>
<td>4</td>
<td>WRITER</td>
<td>Char</td>
<td>44</td>
<td>$44.</td>
<td>$44.</td>
<td>WRITER</td>
</tr>
<tr>
<td>5</td>
<td>PRODUCER</td>
<td>Char</td>
<td>29</td>
<td>$29.</td>
<td>$29.</td>
<td>PRODUCER</td>
</tr>
<tr>
<td>6</td>
<td>TITLE</td>
<td>Char</td>
<td>36</td>
<td>$36.</td>
<td>$36.</td>
<td>TITLE</td>
</tr>
<tr>
<td>7</td>
<td>ARTIST</td>
<td>Char</td>
<td>28</td>
<td>$28.</td>
<td>$28.</td>
<td>ARTIST</td>
</tr>
</tbody>
</table>

RANK ???
What is define.xml
Regulatory landscape
• July 2004 – FDA adds Study Data Specifications v1.0 to draft eCTD Guidance. This specification references the CDISC SDTM for data tabulation datasets

Electronic Common Technical Document (eCTD)

  - Federal Register Notice [TXT] [PDF]
  - The Draft Guidance

Specifications

- eCTD Backbone Files Specification for Module 1
- eCTD Backbone File Specification for Modules 2 through 5
- eCTD Backbone File Specification for Study Tagging Files
- FDA eCTD Table of Contents Headings and Hierarchy
- Study Data Specifications (Posted 7/21/2004)
• March 2005 – Study Data Specifications v1.1: Updates Specifications for Data Set Documentation
  - data definitions
  - annotated case report forms (CRFs)

• “The specification for the data definitions for datasets provided using the CDISC SDTM is included in the Case Report Tabulation Data DefinitionSpecification (define.xml) developed by the CDISC define.xml Team”

• Data Definition for other data sets follows: Providing Regulatory Submissions in Electronic Format – NDA (1999), which is the define.pdf
• 2006 – CDISC SDTM / ADaM Pilot Project: Collaborative Pilot project with FDA and industry to test how well the submission of CDISC compliant data sets and associated metadata meets the needs of both medical and statistical FDA reviewers

• Generation of ICH E3/eCTD clinical study report (CSR) using the CDISC data models

• Data Definition Tables were provided in XML format (CRT-DDS, define.xml)
Data Definition Tables in PDF
1999 Guidance: sponsor has to document submitted data by including data definition tables (`define.pdf`) and annotated case report forms (`blankcrf.pdf`)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Description of dataset</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEMO</td>
<td>Demographics</td>
<td><code>crt/datasets/1234/demo.xpt</code></td>
</tr>
<tr>
<td>INCLUDE</td>
<td>Inclusion criteria</td>
<td><code>crt/datasets/1234/include.xpt</code></td>
</tr>
</tbody>
</table>
1999 Guidance: sponsor has to document submitted data by including data definition tables (define.pdf) and annotated case report forms (blankcrf.pdf)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Label</th>
<th>Type</th>
<th>Codes</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATID</td>
<td>Patient identification</td>
<td>char</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STUDY</td>
<td>Study number</td>
<td>char</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CENTER</td>
<td>Study center</td>
<td>char</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRT</td>
<td>Assigned treatment group</td>
<td>num</td>
<td>0 = placebo&lt;br&gt;5 = 5mg/day</td>
<td></td>
</tr>
<tr>
<td>SEX</td>
<td>Sex of subject</td>
<td>char</td>
<td>f = female&lt;br&gt;m = male</td>
<td></td>
</tr>
<tr>
<td>BDATE</td>
<td>Birth date</td>
<td>date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WEIGHT</td>
<td>Weight in kg</td>
<td>num</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Data Definition Tables in XML
Data Definition Tables - XML

• As of January 1, 2008: follow the eCTD guidance and document submitted data by including data definition tables (define.xml) and annotated case report forms (blankcrf.pdf)

```xml
<?xml version="1.0" encoding="ISO-8859-1" ?>
<ODM xmlns="http://www.cdisc.org/ns/odm/v1.2"
     xmlns:xlink="http://www.w3.org/2001/XLink"
     xmlns:cdisc="http://www.cdisc.org/ns/cdisc/v1.0"
     xmlns:fmt="http://www.w3.org/1999/xlink"
     xsi:schemaLocation="http://www.cdisc.org/ns/odm/v1.2 define1-0-0.xsd">
  <Study OID="CDISC01">
    <GlobalVariables/>
    <MetaDataVersion OID="CDISC.SDM.3.1.1"
                     Name="Study CDISC01, Data Definitions"
                     Description="Study CDISC01, Data Definitions"
                     def:DefineVersion="1.0.0"
                     def:StandardName="CDISC SDTM"
                     def:StandardVersion="3.1.1"/>

    <ItemGroupDef GID="MH"
                  Name="MH" Repeating="Yes" IsReferenceData="No"
                  Purpose="Tabulation" def:Label="Medical History"
                  def:Structure="One record per medical history event per subject"
                  def:DomainKeys="STUDYID, USUBJID, MHCAT, MHTERM, MHSTDTC"
                  def:Class="EVENTS"
                  def:ArchiveLocationID="Location.MH">
      <ItemRef ItemOID="STUDYID" OrderNumber="1" Mandatory="Yes"
               Role="IDENTIFIER" RoleCodeListOID="RoleCodeList"/>
    </ItemGroupDef>

    <def:leaf ID="Location.MH" xlink:href="mh.xpt">
      <def:title>mh.xpt</def:title>
    </def:leaf>
  </Study>
</ODM>
```
Data Definition Tables - XML

• As of January 1, 2008: follow the eCTD guidance and document submitted data by including data definition tables (define.xml) and annotated case report forms (blankcrf.pdf)
• Case Report Tabulation Data Specification (CRT-DDS, define.xml)

• Production version: 1.0.0

• Based on version ODM version 1.2.1

• CRT-DDS version 1.0.0 is currently the only production version

• Maintained by CDISC’s XML Technologies Team (formerly known as the ODM team)

• New draft version of define.xml expected in 2010 with additional metadata support
Define.XML

FDA Adds CDISC ODM Define.xml to Study Data Specifications

The FDA has now included the CDISC Case Report Tabulation Data Definition Specification (define.xml), which is based on the CDISC ODM, as part of the eCTD Study Data Specifications for the eCTD for submissions using the SDTM. The revised specifications are available here.

Case Report Tabulation Data Definition Specification (CRT-DDS, also called define.xml)
Final Version 1.0

CRT-DDS Released for Implementation February 10, 2005.

The CDISC define.xml Team has published the Case Report Tabulation Data Definition Specification (define.xml) Version 1.0 for
## Case Report Tabulation Data Definition Specification (define.xml)
Prepared by the CDISC define.xml team

### Notes to Readers
This version of the Case Report Tabulation Data Definition Specification supersedes all prior versions. Version 1.0.0 reflects changes from a comment period through the Health Level 7 (HL7) Registered Clinical Research Information Management Technical Committee (RCRIM) in December 2003 (www.hl7.org) and CDISC website in September 2004 as well as work done by the define.xml team to add functionality, features, and additional documentation.

Version 1.0.0 incorporated the applicable comments, suggestions, and corrections received from the two comment periods specified above and is the official implementation version.

### Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Summary of Changes</th>
<th>Primary Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002-01-05</td>
<td>1.0.0</td>
<td>This is the official implementation version of the Case Report Tabulation Data Definition Specification.</td>
<td></td>
</tr>
<tr>
<td>2005-01-09</td>
<td>1.0.0</td>
<td>Administrative updates.</td>
<td>Anthony France, William Quack, Sally Casella</td>
</tr>
</tbody>
</table>
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="define-1-0-0.xsl"?>

<ODM xmlns="http://www.cdisc.org/ns/odm/v1.2"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:xlink="http://www.w3.org/1999/xlink"
     xsi:schemaLocation="http://www.cdisc.org/ns/odm/v1.2 define1-0-0.xsd"
     FileOID="StudyCDS1"
     ODMVersion="1.2" Filetype="Snapshot"
     CreationDateTime="2007-04-09T12:24:09">
  <Study OID="cdisc01">
    <GlobalVariables>
      <StudyName>CDISC01</StudyName>
      <StudyDescription>CDISC01 Test Study.</StudyDescription>
      <ProtocolName>CDISC01</ProtocolName>
    </GlobalVariables>
    <MetaDataVersion OID="CDISC.SDTM.3.1.1"
                      Name="Study CDISC01, Data Definitions"
                      Description="Study CDISC01, Data Definitions"
                      def:DefineVersion="1.0.0"
                      def:StandardName="CDISC SDTM"
                      def:StandardVersion="3.1.1">
    </MetaDataVersion>
  </Study>
</ODM>
Displaying the define.xml
- From the CDISC SDS Metadata Team (2007): 

\[ \text{define.xml + XSL style sheet = html} \]

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Description</th>
<th>Class</th>
<th>Structure</th>
<th>Purpose</th>
<th>Keys</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>SV</td>
<td>Subject Visits</td>
<td>TRIAL DESIGN</td>
<td>One record per subject per actual visit</td>
<td>Tabulation</td>
<td>STUDYID, USUBJID, SVSTDTIC, SVENDTIC, VISITNUM, SVUPDES</td>
<td>sv.xpt</td>
</tr>
<tr>
<td>TA</td>
<td>Trial Arms</td>
<td>TRIAL DESIGN</td>
<td>One record per element per arm</td>
<td>Tabulation</td>
<td>STUDYID, ARMCD, TAETORD</td>
<td>ta.xpt</td>
</tr>
<tr>
<td>TE</td>
<td>Trial Elements</td>
<td>TRIAL DESIGN</td>
<td>One record per element</td>
<td>Tabulation</td>
<td>STUDYID, ETCD</td>
<td>te.xpt</td>
</tr>
<tr>
<td>TI</td>
<td>Trial Inclusion</td>
<td>TRIAL DESIGN</td>
<td>One record per I/E criterion</td>
<td>Tabulation</td>
<td>STUDYID, IETESTCD</td>
<td>t.xpt</td>
</tr>
<tr>
<td>TS</td>
<td>Trial Summary</td>
<td>TRIAL DESIGN</td>
<td>One record per planned visit per arm</td>
<td>Tabulation</td>
<td>STUDYID, VISITNUM, ARMCD</td>
<td>ts.xpt</td>
</tr>
<tr>
<td>TV</td>
<td>Trial Visits</td>
<td>TRIAL DESIGN</td>
<td>One record per comment per subject</td>
<td>Tabulation</td>
<td>STUDYID, USUBJID, COSEQ, IDVAR, IDVARVAL</td>
<td>tv.xpt</td>
</tr>
<tr>
<td>CO</td>
<td>Demographics</td>
<td>SPECIAL PURPOSE</td>
<td>One record per subject</td>
<td>Tabulation</td>
<td>STUDYID, USUBJID</td>
<td>co.xpt</td>
</tr>
<tr>
<td>CM</td>
<td>Concomitant Medications</td>
<td>INTERVENTIONS</td>
<td>One record per medication intervention episode</td>
<td>Tabulation</td>
<td>STUDYID, USUBJID, CMCAT, CMITRT, CMSTDTIC, CMENDTIC.</td>
<td>cm.xpt</td>
</tr>
</tbody>
</table>
From the CDISC SDS Metadata Team (2007): 
*define.xml + XSL style sheet = html*

<table>
<thead>
<tr>
<th>Variable</th>
<th>Label</th>
<th>Type</th>
<th>Controlled Terminology</th>
<th>Origin</th>
<th>Role</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>STUDYID</td>
<td>Study Identifier</td>
<td>text</td>
<td></td>
<td>CRF Page 3</td>
<td>IDENTIFIER</td>
<td>Concatenation of STUDYID:SUBJID</td>
</tr>
<tr>
<td>DOMAIN</td>
<td>Domain Abbreviation</td>
<td>text</td>
<td></td>
<td>DERIVED</td>
<td>IDENTIFIER</td>
<td></td>
</tr>
<tr>
<td>SUBJID</td>
<td>Unique Subject Identifier</td>
<td>text</td>
<td></td>
<td>SPONSOR DEFINED</td>
<td>IDENTIFIER</td>
<td></td>
</tr>
<tr>
<td>CMSEQ</td>
<td>Sequence Number</td>
<td>integer</td>
<td></td>
<td>DERIVED</td>
<td>IDENTIFIER</td>
<td></td>
</tr>
<tr>
<td>CMSPID</td>
<td>Sponsor-Defined Identifier</td>
<td>text</td>
<td></td>
<td>SPONSOR DEFINED</td>
<td>IDENTIFIER</td>
<td></td>
</tr>
<tr>
<td>CMTRT</td>
<td>Reported Name of Drug, Med., or Therapy</td>
<td>text</td>
<td></td>
<td>CRF Pages 6, 35</td>
<td>TOPIC</td>
<td></td>
</tr>
<tr>
<td>CMMODIFY</td>
<td>Modified Reported Name</td>
<td>text</td>
<td></td>
<td>SPONSOR DEFINED</td>
<td>SYNONYM QUALIFIER</td>
<td>WHODRUG Version 2002/04</td>
</tr>
<tr>
<td>CMDECOD</td>
<td>Standardized Medication Name</td>
<td>text</td>
<td>DRUGDICT F</td>
<td>DERIVED</td>
<td>SYNONYM QUALIFIER</td>
<td></td>
</tr>
<tr>
<td>CMCAT</td>
<td>Category for Medication</td>
<td>text</td>
<td>CMCAT F</td>
<td>CRF Pages 6, 35</td>
<td>GROUPING QUALIFIER</td>
<td>Sponsor controlled terminology</td>
</tr>
</tbody>
</table>
Why convert the define .xml to SAS ??
Define.xml in the SDTM/ADaM CDISC FDA Pilot – Printing Issue
“A major issue identified by the regulatory review team was the difficulty in printing the Define file. The style sheet used in the pilot submission package was developed with the primary target of web browser rendering, which is not readily suited to printing. Reviewers who attempted to print the Define file found that the file did not fit on portrait pages, that page breaks were not clean, and that printing only a portion of the file was difficult. Opening the document in another application (e.g., Microsoft Word) provided a work-around, but was not an option that was user friendly or efficient.”
• CDISC SDTM / ADaM Pilot project report:

“This problem could be viewed as an implementation issue that sponsors will need to handle, after discussing the issue with their FDA reviewers. For example, a sponsor might choose to provide two versions of the style sheet – XML for viewing and PDF for printing. Ideally, a reminder of the issue would be included somewhere in the CRT-DDS guidance (e.g., a note that consideration be given to how the sponsor will respond to a request from reviewers for a print-friendly version of the style sheet). It should be noted that the regulatory review team for the pilot project emphasized that the ability to print the document would be essential for the future use of XML files.”
Printing the define.xml
• The **PDF** format is the *de facto standard* for printable documents on the web

• PDF is platform independent (no browser issues …)

• How can we create a PDF file from the define.xml ??
SAS Solution:

- Convert the XML hierarchy to a relational data model in the form of (2-dimensional) SAS data sets
- Once we have the define.xml content in SAS datasets, we can use SAS to create a PDF rendition (with ODS PDF) (which is an easy task for a SAS programmer)
Validating the define.xml
Some process used Metadata (data sets, variables, codelists) to create:
- define.xml
- SAS transport files

Validating the define.xml:
1. well-formedness
2. Against Schema
3. Against CRT-DDS Specification
4. Against SAS transport files
5. Against SDTM spec ("mandatory")
Validating the define.xml:

1. well-formedness
2. Against Schema
3. Against CRT-DDS Specification
4. Against SAS transport files
5. Against SDTM spec (“mandatory”)

Many XML based tools can do this.

XML schema (1.0) can not do this. Schema 1.1 ?? Schematron ??

XML based tools ????
Define.xml -> SAS datasets: Validation

Some process used Metadata (data sets, variables, codelists) to create:
- define.xml
- SAS transport files

Use SAS XML Mapper to convert define.xml to SAS data sets

define.xml as SAS datasets

VALIDATION

use SAS ODS to create define.pdf
Other examples of using the define.xml metadata
• Use codelist information (codes/decodes) to create a PROC FORMAT

```xml
<CodeList OID="SEX" Name="SEX" DataType="text" SASFormatName="$SEX">
  <CodeListItem CodedValue="F">
    <Decode>
      <TranslatedText xml:lang="en">FEMALE</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue="M">
    <Decode>
      <TranslatedText xml:lang="en">MALE</TranslatedText>
    </Decode>
  </CodeListItem>
  <CodeListItem CodedValue="U">
    <Decode>
      <TranslatedText xml:lang="en">UNKNOWN</TranslatedText>
    </Decode>
  </CodeListItem>
</CodeList>
```
• Use variable information (type, length, label) to create zero-observation datasets that can serve as data conversion targets

```
<ItemDef OID="DM.SEX"
  Name="SEX"
  DataType="text"
  Length="1"
  Origin="CRF Page 3"
  def:Label="Sex">
...
```

```
<ItemDef OID="DM.RACE"
  Name="RACE"
  DataType="text"
  Length="10"
  Origin="CRF Page 3"
  def:Label="Race">
...
```
define.xml as a relational model
• How to Convert the XML hierarchy to a relational data model in the form of (2-dimensional) SAS data sets

• Exercise02_ImportDefine_Naive.sas

FILENAME define 'WorkShop\tabulations\define.xml';
LIBNAME define xml XMLFILEREF=define;

PROC CONTENTS DATA=define._ALL_ VARNUM;
RUN;
FILENAME define "&WorkShop\tabulations\define.xml";
LIBNAME define xml XMLFILEREF=define;
NOTE: Libref DEFINE was successfully assigned as follows:
    Engine: XML
    Physical Name: DEFINE

PROC CONTENTS DATA=define..ALL..VARNUM;
RUN;

ERROR: XML describe error: XML data is not in a format supported natively by the XML libname engine. Files of this type usually require an XMLMap to be input properly.  
NOTE: Statements not processed because of errors noted above.
• **How to Convert** the XML hierarchy to a relational data model in the form of (2-dimensional) SAS data sets

• **Solution: SAS XML Mapper**

• **SAS XML Mapper:** free stand-alone Java client application available on the SAS product distribution disks

• Uses XPATH to create a MAP file that maps hierarchical XML to rows and columns in SAS
In SAS:

FILENAME DEFINE  "C:\HOW\jansen\tabulations\define.xml";
FILENAME SXLEMAP  "C:\HOW\jansen\maps\define.map";
LIBNAME DEFINE XML XMLMAP=SXLEMAP access=READONLY;

PROC COPY IN=define OUT=outlib;
RUN;
Solution: SAS XML Mapper + SAS ODS PDF

You will need to create a relational data model to convert the XML hierarchy to the 2-dimensional SAS data set

```xml
<ItemGroupDef OID="MH"
  Name="MH" Repeating="Yes" IsReferenceData="No"
  Purpose="Tabulation" def:Label="Medical History"
  def:Structure="One record per medical history event per subject"
  def:DomainKeys="STUDYID, USUBJID, MHCAT, MHTERM, MHSTDTC"
  def:Class="EVENTS" def:ArchiveLocationID="Location.AE">
  <ItemRef ItemOID="STUDYID" OrderNumber="1" Mandatory="Yes"
    Role="IDENTIFIER" RoleCodeListOID="RoleCodeList" />
</ItemGroupDef>

<def:leaf ID="Location.MH" xlink:href="mh.xpt">
  <def:title>mh.xpt</def:title>
</def:leaf>

<def:leaf/>
...
```

```xml
<ItemDef OID="STUDYID" Name="STUDYID" DataType="text" Length="7"
  Origin="CRF Page 3" def:Label="Study Identifier" />
```
SAS XML Mapper
### Define.xml -> SAS datasets

| Study_OID | MetaDataVersion_OID | ItemGroupDef_OID | ItemGroupDef_Name | ItemGroupDef_Repea | ItemGroupDef_Purpose | ItemGroupDef_Label | ItemGroupDef_St | ItemGroupDef_Doma | ItemGroupDef_Cl | ItemGroupDef_Def_Art |
|-----------|---------------------|------------------|-------------------|-------------------|----------------------|-------------------|----------------|-------------------|----------------|----------------|------------------|
| cdisc01   | CDISC.SDTM.3.1.1     | AE               | AE                | No                | Tabulation           | AE                | Yes             | No                | Tabulation       | AE              | EDG               |
| cdisc01   | CDISC.SDTM.3.1.1     | AE               | AE                | No                | Tabulation           | AE                | Yes             | No                | Tabulation       | AE              | EDG               |
| cdisc01   | CDISC.SDTM.3.1.1     | MH               | MH                | No                | Tabulation           | MH                | Yes             | No                | Tabulation       | MH              | Medical History   |
| cdisc01   | CDISC.SDTM.3.1.1     | DA               | DA                | Yes               | Tabulation           | DA                | No              | Yes               | Tabulation       | DA              | Drug Accountability |

### itemgroupdef.sas7bdat

<table>
<thead>
<tr>
<th>Study_OID</th>
<th>MetaDataVersion_OID</th>
<th>ItemDef_OID</th>
<th>ItemDef_Name</th>
<th>ItemDef_Type</th>
<th>ItemDef_Length</th>
<th>ItemDef_Origin</th>
<th>ItemDef_Comment</th>
<th>ItemDef_Label</th>
<th>CodeListRef_CodeListOID</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdisc01</td>
<td>CDISC.SDTM.3.1.1</td>
<td>AE.AEEM</td>
<td>AEEM</td>
<td>AEM</td>
<td>25</td>
<td>SPONSOR DEFINED</td>
<td>ID of entity</td>
<td>AEEM</td>
<td></td>
</tr>
<tr>
<td>cdisc01</td>
<td>CDISC.SDTM.3.1.1</td>
<td>AE.AEERM</td>
<td>AEERM</td>
<td>AER</td>
<td>10</td>
<td>DERIVED</td>
<td>Dictionary-Derived Element</td>
<td>AEERM</td>
<td></td>
</tr>
<tr>
<td>cdisc01</td>
<td>CDISC.SDTM.3.1.1</td>
<td>AE.AESEV</td>
<td>AESEV</td>
<td>AES</td>
<td>2</td>
<td>SPONSOR DEFINED</td>
<td>Modified Reported Term</td>
<td>AESEV</td>
<td></td>
</tr>
<tr>
<td>cdisc01</td>
<td>CDISC.SDTM.3.1.1</td>
<td>AE.AEGR</td>
<td>AEGR</td>
<td>AEG</td>
<td>1</td>
<td>SPONSOR DEFINED</td>
<td>Serious Event</td>
<td>AEGR</td>
<td></td>
</tr>
<tr>
<td>cdisc01</td>
<td>CDISC.SDTM.3.1.1</td>
<td>AE.AEAN</td>
<td>AEAN</td>
<td>AEA</td>
<td>30</td>
<td>CRF Page 34</td>
<td>Action Taken with Study</td>
<td>AEAN</td>
<td></td>
</tr>
</tbody>
</table>
<TABLE name="ItemGroupDef">
  <TABLE-DESCRIPTION>ItemGroupDef</TABLE-DESCRIPTION>
  <TABLE-PATH syntax="XPath">/ODM/Study/MetaDataVersion/ItemGroupDef</TABLE-PATH>

  <COLUMN name="MetaDataVersion_OID" retain="YES">
    <PATH syntax="XPath">/ODM/Study/MetaDataVersion/@OID</PATH>
    <DESCRIPTION>OID</DESCRIPTION>
    <TYPE>character</TYPE>
    <DATATYPE>string</DATATYPE>
    <LENGTH>100</LENGTH>
  </COLUMN>

  <COLUMN name="ItemGroupDef_OID">
    <PATH syntax="XPath">/ODM/Study/MetaDataVersion/ItemGroupDef/@OID</PATH>
    <DESCRIPTION>OID</DESCRIPTION>
    <TYPE>character</TYPE>
    <DATATYPE>string</DATATYPE>
    <LENGTH>100</LENGTH>
  </COLUMN>
</TABLE>
Define.xml -> SAS datasets

XMLMAP Selected Elements Tree View (version 1.2)

Legend:
+ 1 or more times
? 0 or 1 times

1 SXLEMAP

+ TABLE

+ COLUMN 1 TABLE-PATH

1 DATATYPE ? FORMAT ? INFORMAT 1 LENGTH 1 TYPE 1 PATH
PROC FORMAT FMTLIB;

VALUE $FRAME
   "S" = "SMALL"
   "M" = "MEDIUM"
   "L" = "LARGE"
   "XL" = "EXTRA LARGE"
;

VALUE $SEX
   "F" = "FEMALE"
   "M" = "MALE"
   "U" = "UNKNOWN"
;

VALUE SMKCLAS
   1 = "NEVER SMOKED"
   2 = "SMOKER"
   3 = "EX SMOKER"
;

RUN;
CREATE TABLE DM("LABEL"="Demographics") {
    STUDYID CHAR(7) "LABEL"="Study Identifier",
    DOMAIN CHAR(2) "LABEL"="Domain Abbreviation",
    USUBJID CHAR(14) "LABEL"="Unique Subject Identifier",
    SUBJID CHAR(6) "LABEL"="Subject Identifier for the Study",
    RFSTDTC CHAR(10) "LABEL"="Subject Reference Start Date/Time",
    RPENDTC CHAR(10) "LABEL"="Subject Reference End Date/Time",
    SITEID CHAR(3) "LABEL"="Study Site Identifier",
    BRTHDTC CHAR(10) "LABEL"="Date/Time of Birth",
    AGE NUMERIC "LABEL"="Age in AGEU at RFSTDTC",
    AGEU CHAR(5) "LABEL"="Age Units",
    ...
};