

The XML Engine in Base/SAS® PNWSUG, 2007

Don Hurst, Zymogenetics

Sarmad Pirzada, Hybrid Data Systems

Disclaimer

The material contained in this presentation represents the opinions of the presenters and should not be construed as reflecting ZymoGenetics' position or opinion.

XML Questions

- How do I represent the hierarchical nature of XML in SAS datasets/tables?
- How do I manipulate XML file data within the SAS system in accordance to a set of business rules?
- What SAS and representative non-SAS tools are available to facilitate XML file handling requirements?
- If there is a need to create an XML output file, what does one need to consider?

XML Terms

- XML: Extensible Markup Language
 - New method of for web applications and data transfer
 - www.w3c.org standard for creating XML tags for transferring and interpreting business files
- TAG: `<subjnum>2008</subjnum>`
- DTD: Data type definition
 - collection of tags
 - parent child relationship

SGML – XML Relation

DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT S1CPOXMT (SPEC_HDR?,
S1CPOXMTcmd)>
<!ELEMENT S1CPOXMTcmd (CCD, CIC, SPL,
ICR, POC, BNO?, QTL?, EVT?,
S1CPOXMT_ORDER+)>
<!ELEMENT S1CPOXMT_ORDER (TNC, CPO,
PNR, QTO, UNT, UNP, SSD, SSM?, SHT?, QTN?,
SDC?, MFR?, PRI?, ACN?, DIS?, PKG?, LSE?,
PBR?, PDP?, VRN?, ACK?, POU?, CTN?, REM*)>
Include SPEC HDR
Include Common Support Data Dictionary
```

XML

```
<S1CPOXMTcmd>
  <CCD>S1BOOKED</CCD>
  <CIC>QF2</CIC>
  <SPL>81205</SPL>
  <ICR>USD</ICR>
  <POC>1</POC>
  <BNO>3</BNO>
  <S1CPOXMT_ORDER>
    <TNC>341</TNC>
    <CPO>EOLJ1234567</CPO>
    <PNR>65B2468</PNR>
    <QTO>1</QTO>
    <UNT>EA</UNT>
    <UNP>25.20</UNP>
    <SSD>20030715</SSD>
  </S1CPOXMT_ORDER>
</S1CPOXMTcmd>
</S1CPOXMT>
```

A DTD – Data Type Definition is used to identify the tags to be used in the XML message.

What Does XML Look Like?

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

```
<!--
```

```
  Edited with XML Spy v2007 (http://www.altova.com)  -->
```

```
<note>
```

```
<to> ove </to>
```

```
<from> Jani </from>
```

```
<heading> Reminder </heading>
```

```
<body> Don't forget me this weekend! </body>
```

```
</note>
```

Another XML Example

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

```
<breakfast menu>
```

```
  <food>
```

```
    <name>Belgian Waffles</name>
```

```
    <price>$5.95</price>
```

```
    <description>two of our famous Belgian Waffles with  
      plenty of real maple syrup</description>
```

```
    <calories>650</calories>
```

```
  </food >
```

```
</breakfast menu>
```

Every XML File Should Have

- There is a “beginning” and “end” container tag
- There is a hierarchy of items within the “food” segments
- Every segment and field begins and ends with a tag
- It should be well-formed

XML Error (Internet Explorer)

The XML page cannot be displayed

Cannot view XML input using XSL style sheet.

Please correct the error and then click the Refresh button, or try again later.

End tag '**Ffrom**' does not match the start tag 'from'. Error processing resource
'http://www.somewhereovertherainbow.com/xml/note_error.xml..

<from> Jani </**F**from> -----^

Correcting the XML Error

- Several methods:
 - Note pad and redisplaying in IE
 - W3C consortium tools
 - Sample 3d XML editing tools
 - XML Spy
 - Exchanger XML Editor
 - oXygen XML Editor

Representing a Hierarchy in SAS

- Visual perspective of the hierarchical XML file structure small part of XML
- SAS datasets are rows/columns, e.g. observations/variables
- Defaults that exist when an XML file is read into SAS
- 3 levels can be handled without extensive mapping

An Example

```
<?xml version="1.0" encoding="windows-1252" ?>
```

```
<TABLE>
```

```
<DEMOG>
```

```
<SUBJID> 200 </SUBJID>
```

```
<AGE> 58 </AGE>
```

```
<SEX> Female </SEX>
```

```
<RACE> Asian </RACE>
```

```
<IRTHDA> 1948-04-01 </IRTHDA>
```

```
</DEMOG>
```

```
</TABLE>
```

How do I get that XML file into SAS?

```
libname in1 xml 'Y:\subject.xml' ;  
proc copy indd=in1 outdd=work ;  
run;
```

What does it look like in SAS (Proc Print)?

Demographics “DEMOG” XML file represented in SAS

Obs	BIRTHDA	RACE	SEX	AGE	SUBJID
1	1948-04-01	Asian	Female	58	200
2	1965-12-18	Black	Female	37	300
3	1944-07-22	Other	Male	59	400

Why is the “Y” left off of the “BIRTHDAY” variable name?

Role of Schema/Mapping

- XML schema tells machines how to carry out rules defined in the schema, define the data structure, content, shared vocabularies and semantics of XML documents.
- Business rules often dictate complex hierarchical relationships that are beyond the default capabilities of the SAS/XML engine
- SAS® XML Mapper was created for such a purpose on mapping the XML document into SAS

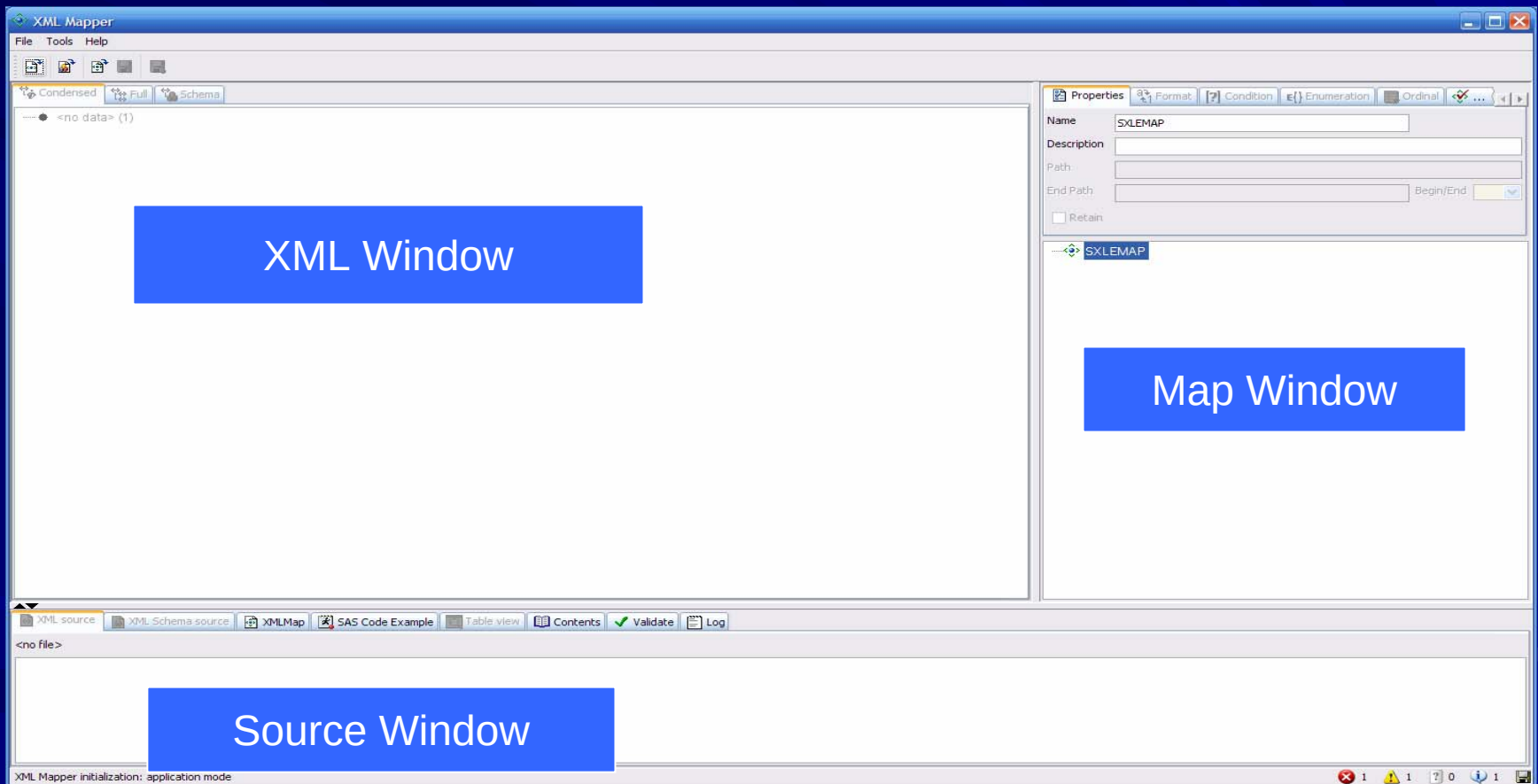
XML Mapper Overview

- XML Mapper is a tool provided by the SAS Institute for telling SAS how to read and represent an XML file into SAS
- XML Mapper has been widely written about and will be briefly covered here.

SAS ® XML Mapper

- Mapper is a Graphical User Interface (GUI) written in Java
- Allows the user to generate XML Map by drop and drag
- Also can show sample SAS code, table views, and contents.

What XML Mapper looks like



XML Mapper with ODM XML

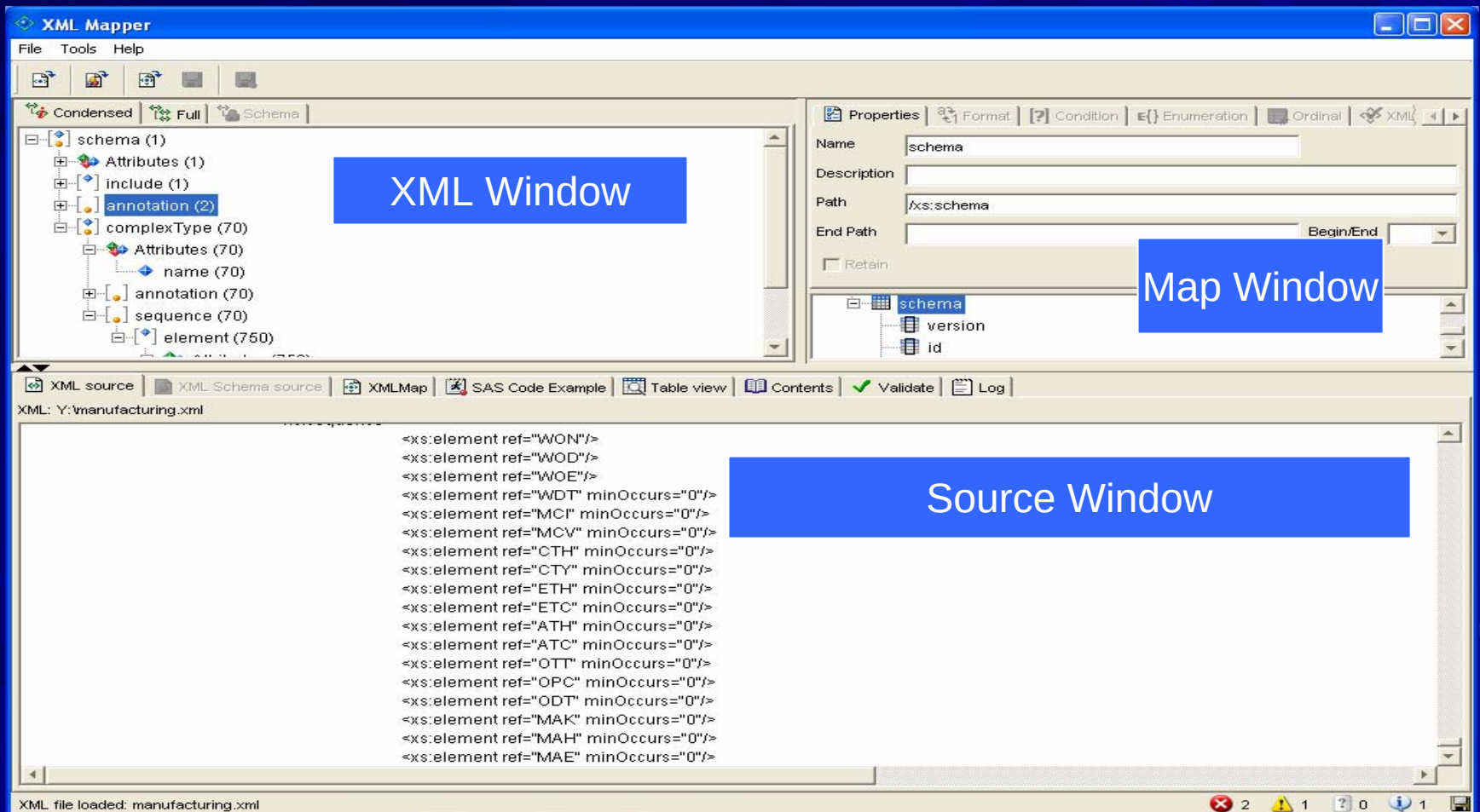
The screenshot displays the XML Mapper application interface, which is divided into three main sections:

- XML Window:** Located on the left, it shows a hierarchical tree structure of the XML document. The tree includes elements like ProtocolName, BasicDefinitions, MetadataVersion, AdminData, ClinicalData, Attributes, SubjectData, SubjectKey, StudyEventData, StudyEventOID, StudyEventRepeatKey, and FormData. A blue box labeled "XML Window" is overlaid on this section.
- Map Window:** Located on the right, it shows the mapping configuration for the selected element. The "Name" field is set to "StudyEventRepeatKey". The "Path" field is set to "/ODM/ClinicalData/SubjectData/StudyEventData/@StudyEventRepeatKey". A blue box labeled "Map Window" is overlaid on this section.
- Source Window:** Located at the bottom, it displays the raw XML source code. The code shows the structure of the XML document, including the StudyEventData, SubjectData, and FormData elements. A blue box labeled "Source Window" is overlaid on this section.

The XML source code in the Source Window is as follows:

```
</StudyEventData>
</SubjectData>
<SubjectData SubjectKey="802101005">
  <StudyEventData StudyEventOID="SE.MyEvent" StudyEventRepeatKey="SE.MyEvent">
    <FormData FormOID="FORM.FD.MyForm" FormRepeatKey="FD.MyForm">
      <ItemGroupData ItemGroupOID="IG.DM" ItemGroupRepeatKey="0000">
        <ItemData ItemOID="ID.studyid" Value="XT802" />
        <ItemData ItemOID="ID.usubjid" Value="802101005" />
        <ItemData ItemOID="ID.siteid" Value="101" />
        <ItemData ItemOID="ID.subjid" Value="802101005" />
        <ItemData ItemOID="ID.domain" Value="DM" />
        <ItemData ItemOID="ID.rfstdtc" Value="2003-02-24T12:51" />
        <ItemData ItemOID="ID.brthdtc" Value="1964-01-22" />
        <ItemData ItemOID="ID.dmdtc" Value="2003-02-24" />
        <ItemData ItemOID="ID.ageu" Value="YEARS" />
        <ItemData ItemOID="ID.country" Value="United States" />
        <ItemData ItemOID="ID.arm" Value="Drug B" />
        <ItemData ItemOID="ID.sex" Value="M" />
        <ItemData ItemOID="ID.rfendtc" Value="2003-06-04T14:47" />
        <ItemData ItemOID="ID.race" Value="CAUCASIAN" />
      </ItemGroupData>
    </FormData>
  </StudyEventData>
</SubjectData>
</StudyEventData>
```

XML Mapper with Manufacturer XML



XML Mapper Shortcomings

- It sometimes will abort with an error without any warning.
- Its validation of a well formed XML structure is not necessarily the same as other 3rd party tools, e.g. XMLspy, oXygen, etc.
- Map creation is very iterative and the resulting map when used in a SAS program can be frustrating.

How to Get SAS® Mapper

- Comes with the SAS Foundation cd's
- Or, download from support.sas.com under base software
- It can be run as a standalone after unzipping by clicking on “sas.sxle.atlas”, an executable jar.
- Meant, however, to be a plug-in into the SAS products of the programs.

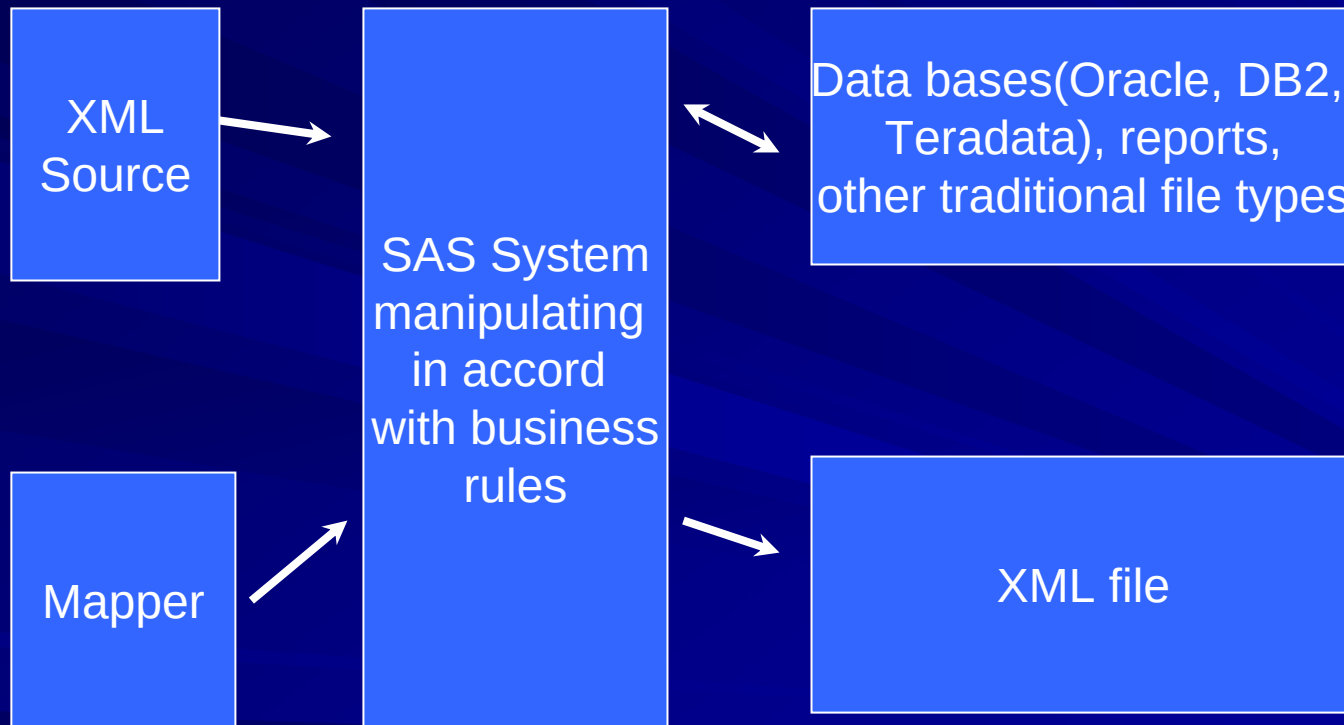
Summary of XML to SAS

- Use XML Libname engine
- Use XML Mapper for mapping and keys
- There are some very good articles on the details related to pharma and non-pharma on using XML within the SAS® System.
- XML Hierarchy is handled via keys within SAS datasets, either by structure or programmatically.

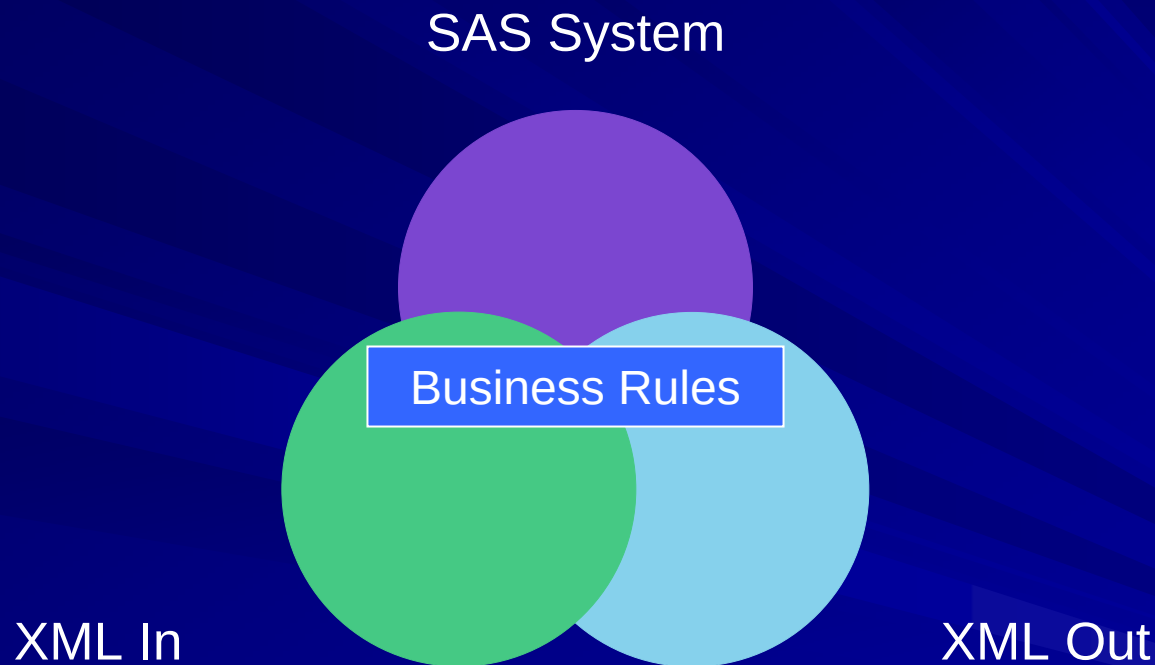
Manipulating an XML File once it is in SAS—Business Rules

- Business rules as reflected in a database or printed output governs the manipulation.
- Business rules are industry specific, e.g.
pharma: baseline ae's vs other ae's;
manu: a recycled apu vs. a new apu
- SAS programmers should use the same approach to manipulating the XML data as they would in any other source of data.

Model of XML/SAS in Many Application Areas



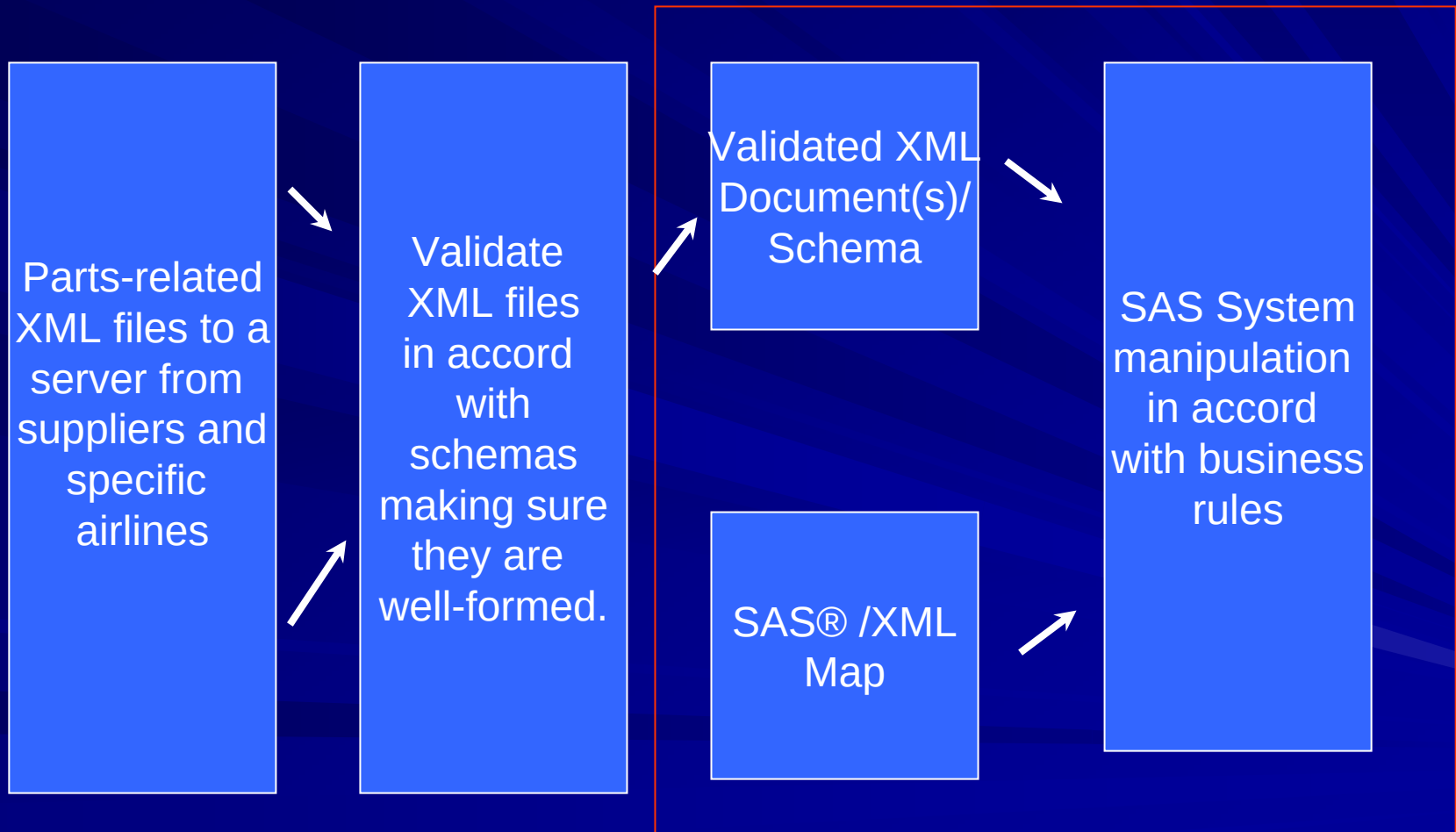
SAS System and XML



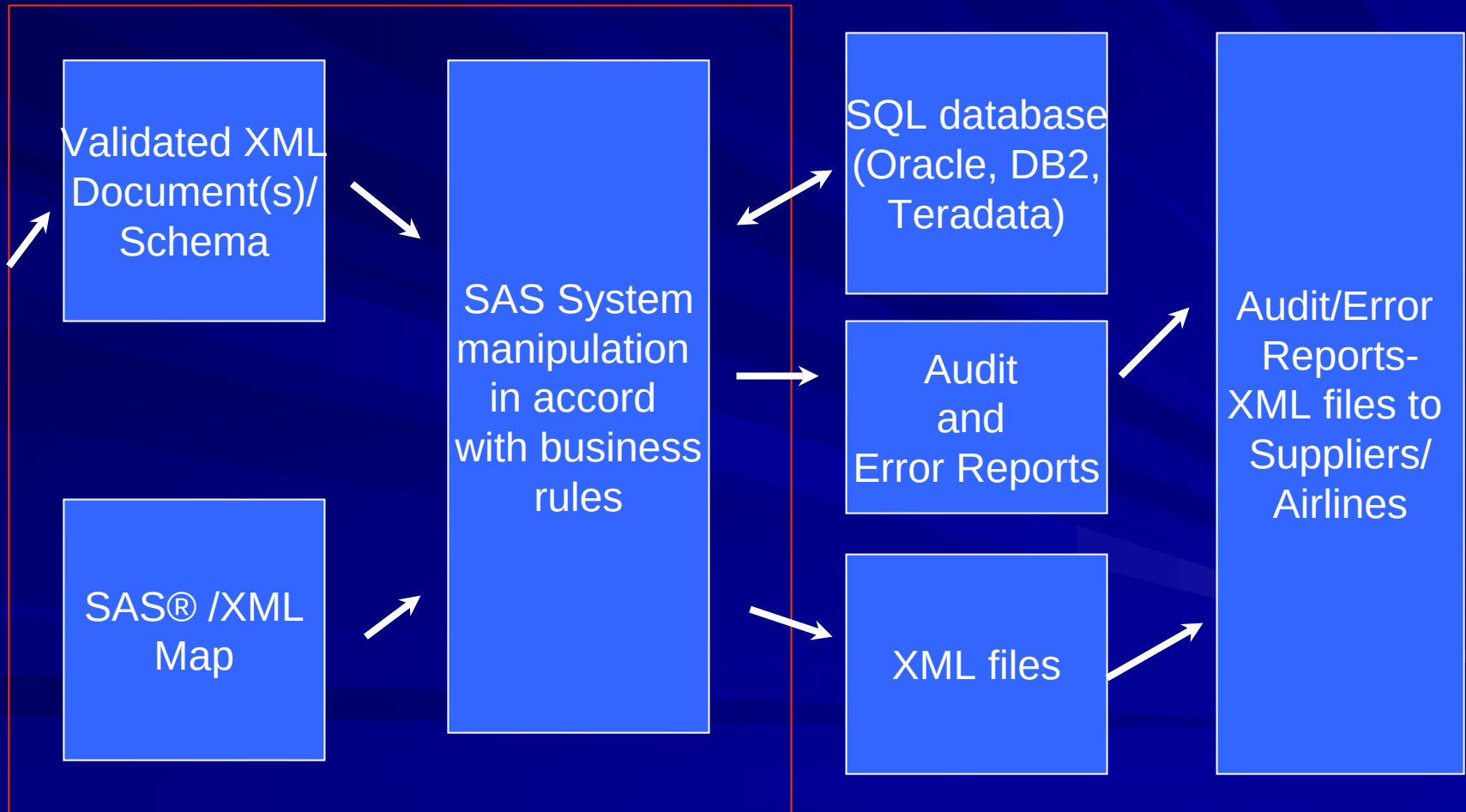
SAS XML Manufacturing Example- 1

- Hypothetical European Airbo airline manufacturer is tracking information related to airplane parts.
- Some of the information tracked:
 - How reliable are the parts
 - Replacement schedule, frequency of replacement
 - Type of aircraft, etc

SAS XML Manufacturing Example- 2



SAS XML Manufacturing Example- 3



SAS XML Pharma Example-1

- Main business driver for XML is the FDA and a FDA warehouse called **Janus**
- The metadata basis for XML and clinical studies is CDISC: www.cdisc.org
- Metadata models:
 - ADaM (Analysis Dataset Model)
 - ODM (Operational Data Model)
 - SDTM (Study Data Tabulation Model)

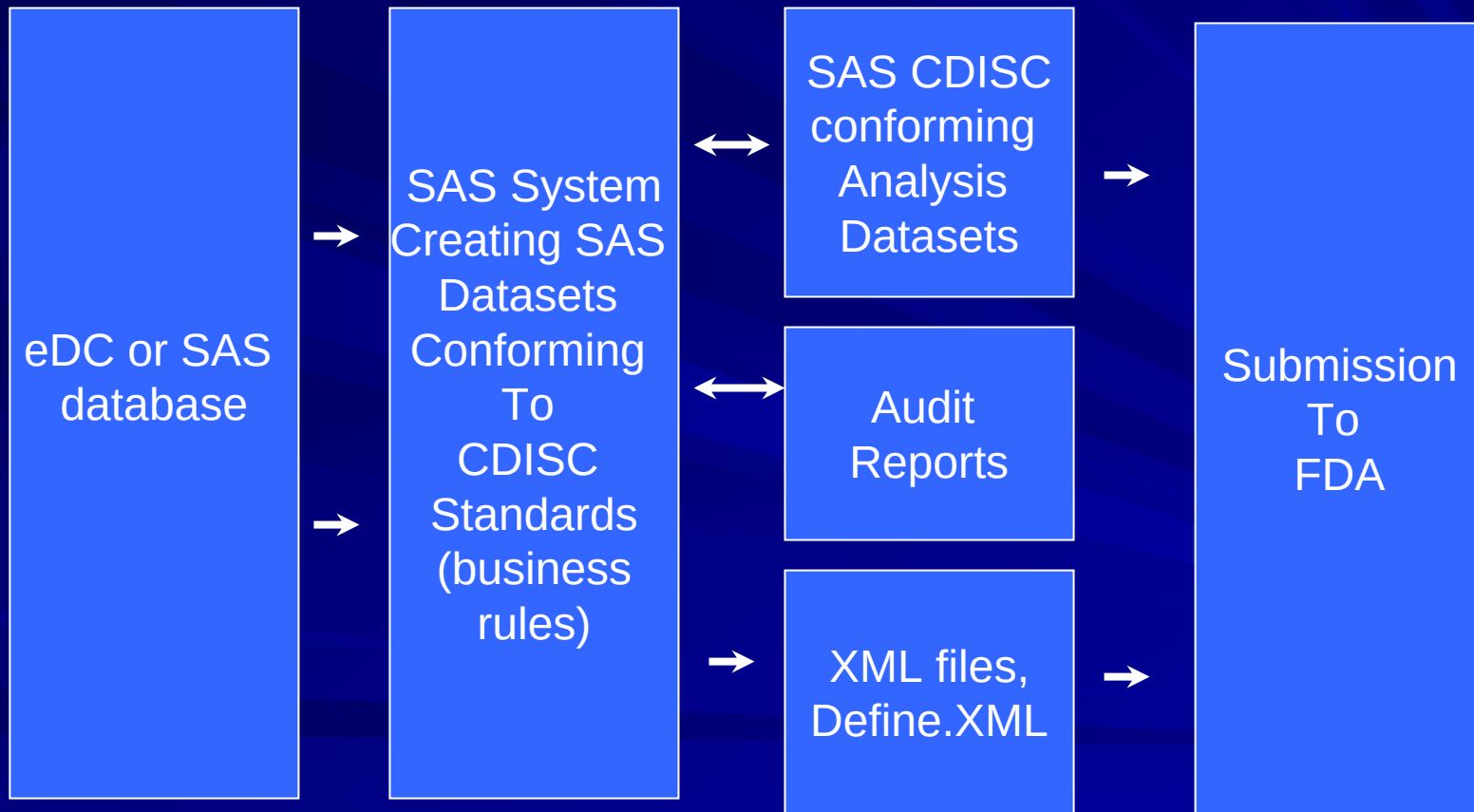
SAS XML Pharma Example-2

- Standardization requirements of CDISC
 - Domains, e.g. dm, lb, ae, mh, ex, etc.
 - Standard variable names within each domain
 - Date variables
 - Subject characteristics
 - AE's, baseline or started after first dose
 - Sub-qualified variables not part of CDISC
 - SAS datasets are vertical

SAS XML Pharma Example-3

- Preparation of the submission datasets follows a similar flow as presented for manufacturing.
- Main difference is that the XML files created are not continuously sent to the FDA, rather is done only for submission
- Future FDA req. of receiving XML files rather than transport V5 files.

Simplified Pharma Clinical Data Flow



Common SAS®/XML Flow

- Validated XML document (xml file + schema)
- Industry specific business rules
- Mapper file for bringing validated XML documents into SAS
- Industry specific XML validation with schema rules in accord with W3C

Creating an XML File from SAS

■ Sample techniques

- Data step
- Proc Template
- ODS
- JAVA
- PROC CDISC
- %DS2TREE
- %DS2CONST

XML Output requirements using these Programming techniques

- There must a beginning and end container tag.
- All fields must have a beginning and end tag.
- There needs to be an associated schema to interpret the XML file
- It must be well-formed, hierarchical, have a schema and in accord to the W3C and specific Industry group standards.

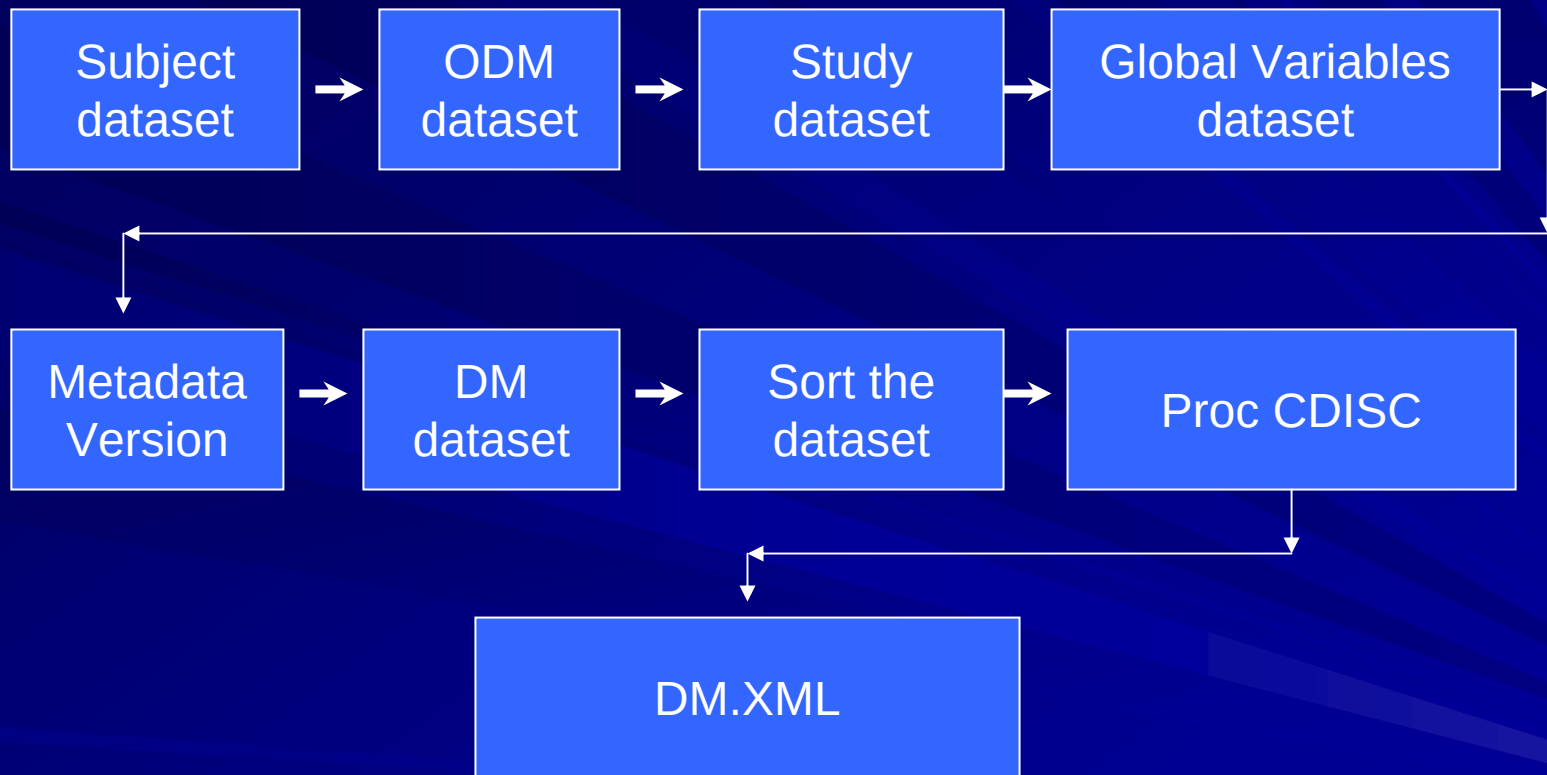
Getting information on how to write an XML file from SAS

- The data step, Proc Template, ODS, and JAVA are well-documented and can be found by “googling” for the relevant terms via the web.
- References at the end of this ppt presentation.

PROC CDISC

- Pharmaceutical specific **proc** developed for XML and handling related XML files
- Using the CDISC standard for schemas, the proc can both read and write conforming XML files.
- Since the CDISC standards are evolving and are being published with various versions, PROC CDISC is updated accordingly.

An Example of Proc CDISC -SAS steps



In SAS-(orig dm)

```
/* Data to create the needed DM data set */
```

```
data orig_dm;
```

```
  infile datalines dlm='|';
```

```
  length studyid $20 usubjid $40 siteid $10 subjid $25 domain $2 rfstdtc $16
```

```
    brthdte $16 dmdtc $16 ageu $6 country $13 arm $16 sex $1 rfendtc $16 race $9;
```

```
  input  studyid :$20. usubjid :$40. siteid :$10. subjid :$25. domain $2.
```

```
    rfstdtc:$16. brthdte :$16. dmdtc age ageu :$6. dmdy country :$13. arm :$16.
```

```
    armcd sex $1. rfendtc :$16. race :$9.;
```

```
  datalines;
```

```
  XT802|802101001|101|802101001|DM|2003-01-10T12:52|1975-02-26|2003-01-  
    10|27|YEARS|1|United States|Drug B|1|M|2003-04-12T10:00|CAUCASIAN
```

```
  .....
```

```
  ;
```

In SAS-(ODM)

```
/* PROC CDISC code to export */  
data odm;  
  ODMVersion = "1.2";  
  fileOID = "2004-09-15 Transfer";  
  FileType = "Snapshot";  
  Description = "DM SDTM v 3.1";  
run;
```

In SAS-(Study dataset)

```
data study;
```

```
  StudyOID = "TRIAL56505";
```

```
run;
```

In SAS-(GlobalVariables)

```
data globalvariables;
```

```
StudyName = "Trial 56505";
```

```
StudyDescription = "Trial 56505 - Study of  
Infectious Agent";
```

```
ProtocolName = "TRIAL56505";
```

```
run;
```

In SAS-(metadataversion)

```
data metadataversion;  
  MetadataVersionOID = "SDTMv3.1";  
  Name = "Some Metadata Version Name";  
run;
```

In SAS-(DM dataset)

data dm;

```
length __STUDYOID __METADATAVERSIONOID  
       __STUDYEVENTOID __STUDYEVENTREPEATKEY __SUBJECTKEY __FORMOID  
       __FORMREPEATKEY  
       __ITEMGROUPOID __ITEMGROUPREPEATKEY __TRANSACTIONTYPE $ 100.;
```

```
set orig_dm;
```

```
retain __STUDYOID      "TRIAL56505"  
       __METADATAVERSIONOID "SDTMv3.1"  
       __STUDYEVENTOID      "SE.MyEvent"  
       __STUDYEVENTREPEATKEY "SE.MyEvent"  
       __FORMOID           "FD.MyForm"  
       __FORMREPEATKEY      "FD.MyForm"  
       __ITEMGROUPOID       "IG.DM"  
       __ITEMGROUPREPEATKEY "0000"  
       __TRANSACTIONTYPE    "Insert";  
       __SUBJECTKEY = usubjid;
```

```
run;
```

In SAS-(Sort DM)

```
proc sort data=dm;  
  by __SUBJECTKEY;  
run;
```

In SAS-(Proc CDISC)

```
filename xmlout "C:\Documents and Settings\Don\Desktop\dm.xml";
```

```
proc cdisc  
  model=odm  
  odm  
  study  
  globalvariables  
  metadataversion  
  clinicaldata  
    domain = "DM"  
    origin = "My Pharma"  
    purpose = "To transfer my trial data"  
    name = "Demographics"  
    comment = "Patient Demographics";  
run;
```

```
write=xmlout;  
data = work.odm;  
data = work.study;  
data = work.globalvariables;  
data = work.metadataversion;  
data = work.DM
```

Datasets created
in prior steps

Final XML DM.XML-1

```
<?xml version="1.0" encoding="windows-1252" ?>
- <!-- Clinical Data Interchange Standards Consortium (CDISC) Operational Data Model
- (ODM) for clinical data interchange You can learn more about CDISC standards
- efforts at http://www.cdisc.org/standards/index.html -->
- <ODM xmlns="http://www.cdisc.org/ns/odm/v1.2"
- xmlns:ds="http://www.w3.org/2000/09/xmlsig#"
- xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
- xsi:schemaLocation="http://www.cdisc.org/ns/odm/v1.2 ODM1-2-0.xsd"
- ODMVersion="1.2" FileOID="2004-09-15 Transfer" FileType="Snapshot"
- Description="DM SDTM v 3.1" AsOfDateTime="2007-08-06T17:51:47"
- CreationDateTime="2007-08-06T17:51:47" SourceSystem="SAS 9.1 PROC CDISC"
- SourceSystemVersion="9.01.01M3P02022006 2.15.42">
- <Study OID="TRIAL56505">
- <!-- GlobalVariables is a REQUIRED section in ODM markup -->
- <GlobalVariables>
- <StudyName>Trial 56505</StudyName>
- <StudyDescription>Trial 56505 - Study of Infectious Agent</StudyDescription>
- <ProtocolName>TRIAL56505</ProtocolName>
- </GlobalVariables>
- <BasicDefinitions />
```

Final XML DM.XML-2, a look at one subject (some data removed)

```
- <SubjectData SubjectKey="802101001">
- <StudyEventData StudyEventOID="SE.MyEvent"
  StudyEventRepeatKey="SE.MyEvent">
- <FormData FormOID="FORM.FD.MyForm"
  FormRepeatKey="FD.MyForm">
- <ItemGroupData ItemGroupOID="IG.DM"
  ItemGroupRepeatKey="0000">
  <ItemData ItemOID="ID.studyid" Value="XT802" />
  <ItemData ItemOID="ID.usubjid" Value="802101001" />
  <ItemData ItemOID="ID.siteid" Value="101" />
  </ItemGroupData>
</FormData>
</StudyEventData>
</SubjectData>
```

Note that tags
have a start
and an end

SAS XML output:

%DS2TREE & %DS2CONST

- While the main objective of these two macros is to create GUI-based applets diagrams, there are parameters in each of these macros for creating XML files.
- For example, %DS2Tree XML parms include:
 - Xmltype
 - Makexml=Y
 - Xmlfile

Concluding Remarks

- Representing the hierarchical nature of XML in SAS datasets/tables
 - XML mapper
 - Data step
 - XML engine
 - 3rd party XML tools
 - W3C consortium

Concluding Remarks (continued)

- Manipulate XML file data within the SAS system in accordance to a set of business rules
 - Depends upon the Industry group and application
 - Desired output affects the manipulation of the SAS representation of the XML data.

Concluding Remarks (continued)

- SAS and representative non-SAS tools are available to facilitate XML file handling
 - XML Mapper
 - Altova xmlspy editor
 - oXygen XML editor
 - Exchanger XML editor

Concluding Remarks (continued)

- Creating an XML output file from SAS
 - Data step (put @'s)
 - ODS
 - Proc Template
 - Proc CDISC
 - %DS2Tree/%DSCONST

References and Further Reading

- SAS® XML Libname Engine 9.1.3, User's Guide, SAS Institute, Cary, NC, USA
- SAS® Implementing CDISC data models in the SAS Metadata Server
- Yeh, Shi-Tao. Using XML Mapper® to Create an XML Map. NESUG18 Proceedings.
- Cisternas, M., (2004): Reading and Writing XML files from SAS®, Proceedings of the 29th SAS Users Group International Conference, Paper 119.
- Pirzada, Sarmad. Moving XML Files within SAS® and Maintaining Segment Hierarchy. PNWSUG, 2005.
- Zender, Cymthia L. Creating a Tagset Template for the SAS® XML Libname Engine. TT-24, SAS Conference Proceedings: PharmaSUG 2007 June 3-6, 2007, Denver, Colorado.

References and Further Reading (continued)

- www.w3c.org
- support.sas.com
- www.cdisc.org
- www.altova.com
- www.oxygenxml.com
- www.freexmleditor.com

Acknowledgements

- SAS is a registered trademark of SAS Institute, Inc. in the USA and other countries. Other brand and product names are registered trademarks or trademarks in their respective companies.
- ® indicates USA registration.

Questions?

- Don Hurst, hurst_don@comcast.net
- Sarmad Pirzada,
sarmad@hybriddatasystems.com