Developing Applications with XML DB

NYOUG Metro Oracle Day

September 23, 2003

Presented By:

Jeff Bernknopf

Director of Business Development

MFG Systems Corporation

AGENDA

- Evolution of XML at Oracle
- What is XML DB?
- A Sample XML DB Application
- XML Documents and an XML Schema
- FTPing XML Documents into XML DB
- Inserting Order Valid XML Documents
- Editing an Invalid Order XML Document
- Generating Supplier XML Documents

Evolution of XML at Oracle

- Oracle has been at the forefront of implementing XML technology
- Approach has been standards based
- XML parser developed based on W3C XML 1.0 standard (SAX/DOM 1.0)
- Parsers developed for Java, C/C++ and PL/SQL
- Parsers become a full function XDK
- Support for XSLT and early schema added

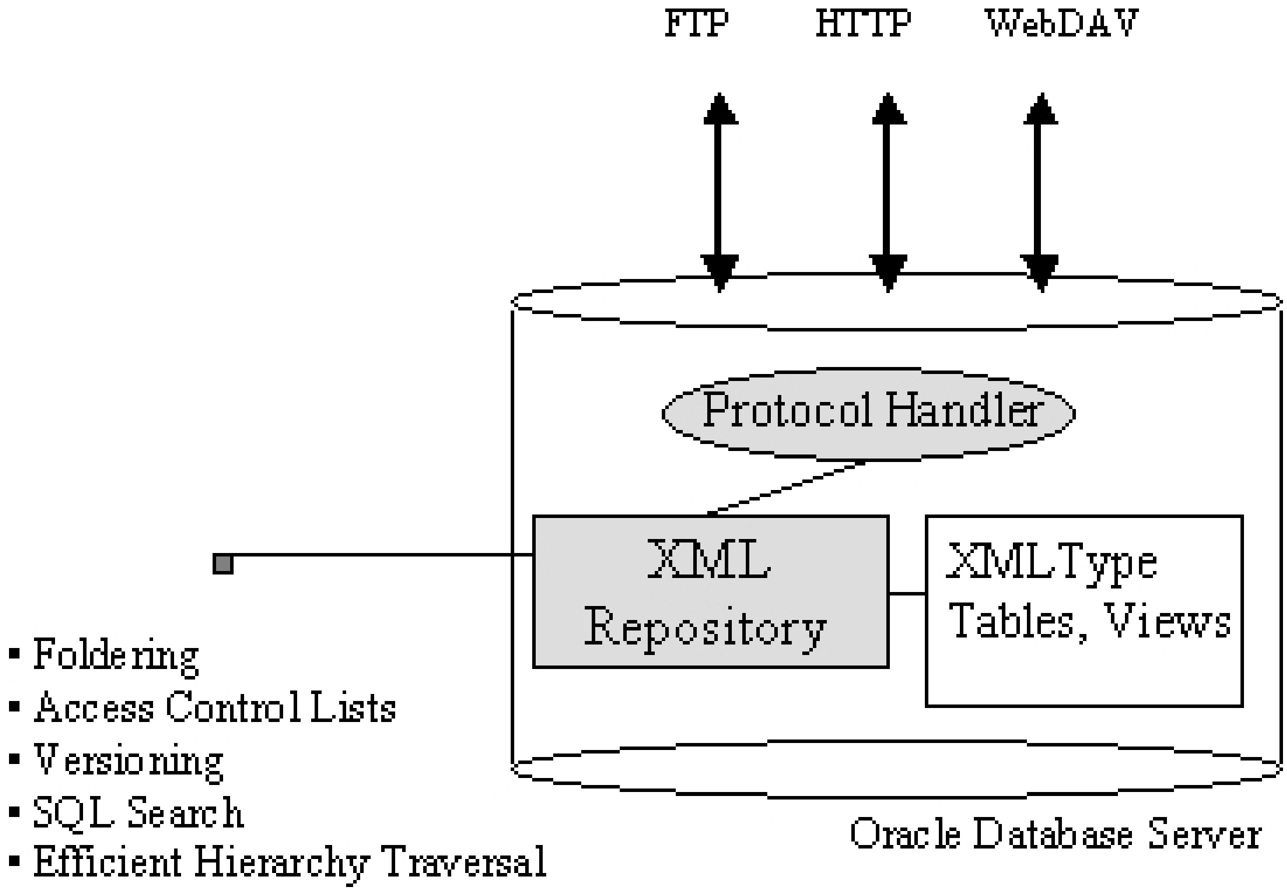
Evolution of XML at Oracle

- XML Parser and PDK enhanced to support SAX/DOM 2 APIs and XML Schema recommendation
- The XMLType datatype is introduced in Oracle 9i Release 1 (9.0.1)
- Oracle XML DB with enhanced XMLType functions and XML repository is included with Oracle 9i Release 2 (9.0.2)

What is XML DB?

- Available with Oracle 9i Release 2 or higher
- Oracle’s implementation of a newer XML standards like XQuery and SQL/XML
- A high performance XML solution
- Provides SQL access to XML data and XML access to SQL data
- Adds an XML Repository to the Oracle database with HTTP, FTP and WebDAV
- Introduces XMLType as a datatype

XML DB Architecture - Content View



FTP

HTTP

WebDAV

Protocol Handler

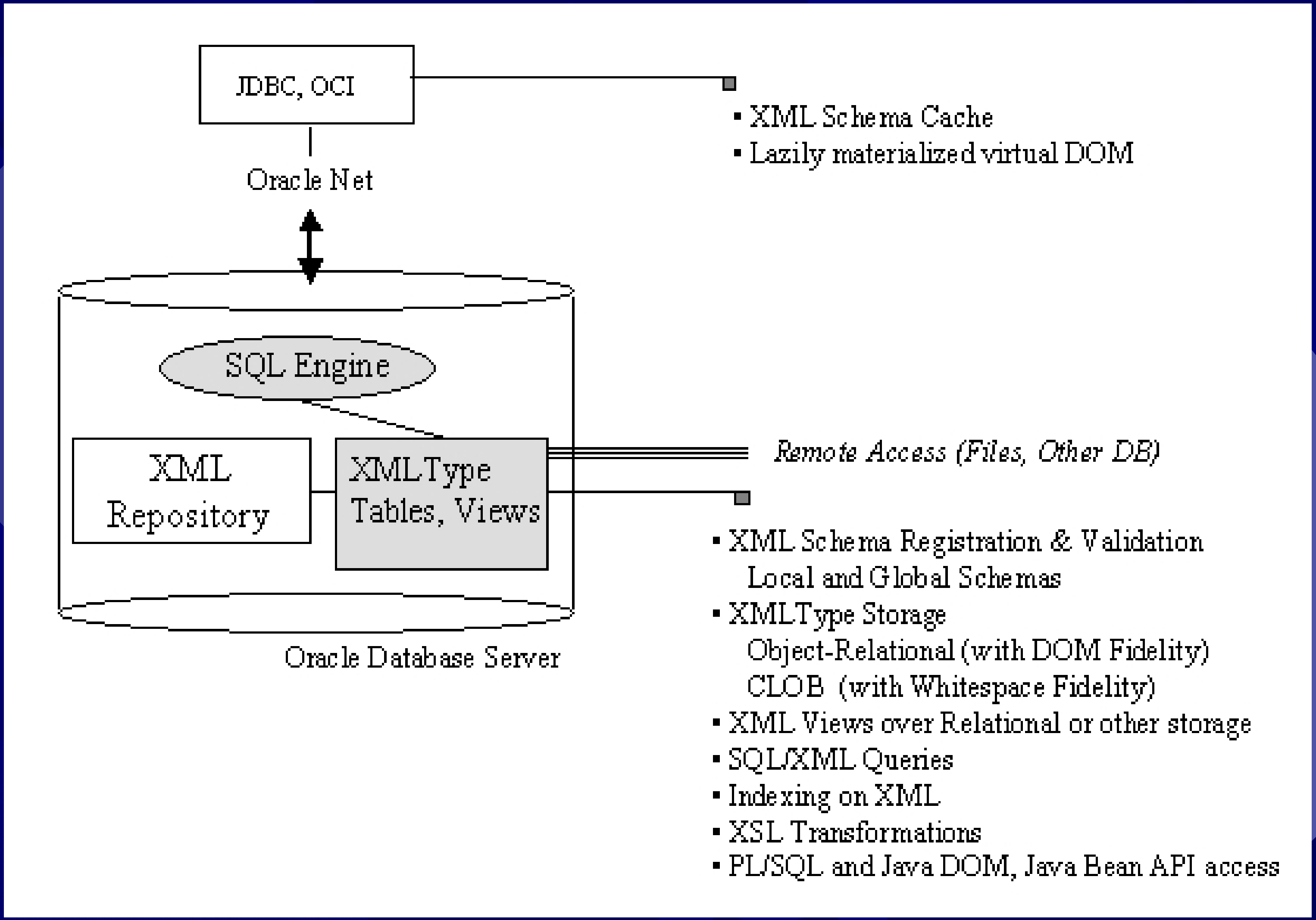
XML Repository

XMLType Tables, Views

- Foldering
- Access Control Lists
- Versioning
- SQL Search
- Efficient Hierarchy Traversal

Oracle Database Server

XML DB Architecture - Data View



- XML Schema Cache
- Lazily materialized virtual DOM

- XML Schema Registration & Validation
  Local and Global Schemas
- XMLType Storage
  Object-Relational (with DOM Fidelity)
  CLOB (with Whitespace Fidelity)
- XML Views over Relational or other storage
- SQL/XML Queries
- Indexing on XML
- XSL Transformations
- PL/SQL and Java DOM, Java Bean API access

An XML DB Application

- To gain an understanding of XML DB, we will develop a sample application
- Application is a portion of a simple order processing system for the Wholesale Chemical Supply (WCS) company
- Customer orders are FTPed to WCS in a specific XML format
- Nightly, WCS consolidates these orders and FTPs an XML document to suppliers

An XML DB Application

- The application requires Customer and Supplier XML documents, an XML schema and two Oracle tables
- Step #1: Orders are FTPed to WCS
- Step #2: Validate incoming orders
- Step #3: Insert valid orders
- Step #4: Store invalid orders separately
- Step #5: Edit invalid orders
- Step #6: Generate Supplier XML files
- Step #7: FTP XML files to Suppliers

The Customer Order XML File

```
<chemOrd xmlns="http://www.mfgsys.com/chemOrd.xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.mfgsys.com/chemOrd.xsd
      "http://www.mfgsys.com/chemOrd.xsd“
      custOrderNo="104" custId="216">
   <fillByDate>12/01/2002</fillByDate>
   <items>
      <item>
         <supplier>Ace Chemical Company</supplier>
         <itemNo>SL35-721</itemNo>
         <unitCost>17.97</unitCost>
         <qty>12</qty>
      </item>
      <item>
         <supplier>Ready Reagents</supplier>
         <itemNo>CP01-487</itemNo>
         <unitCost>162.75</unitCost>
         <qty>5</qty>
   </items>
</chemOrd>
```

The ChemOrder XML Schema

```
<schema
  targetNamespace="http://www.mfgsys.com/chemOrd.xsd"
   xmlns="http://www.w3.org/2001/XMLSchema">
   <element name="chemOrder" type="chemOrderType">
   <element name="items" type="itemType"
      minOccurs="1" maxOccurs="unbounded" />
   <complexType name="chemOrderType">
     <sequence>
       <element name="fillByDate" type="date" />
       <element name="items" type="itemsType" />
   <attribute name="custOrderNo" type="positive_integer"/>
   <attribute name="custId" type="positive_integer"/>
      </sequence></complexType>
   <complexType name="itemsType">
      <sequence>
         <element name="item" type="itemType"
            minOccurs="1" maxOccurs="unbounded" />
      </sequence></complexType>
```

The ChemOrder XML Schema

```
 <complexType name="itemType">
      <sequence>
         <element name=“supplier" type="string" />
         <element name="itemNo" type="char_8" />
         <element name="unitCost" type="decimal_6_2" />
         <element name="itemNo" type="positive_integer" />
      </sequence>
   </complexType>
   <simpleType name="decimal_6_2">
      <restriction base="decimal">
         <totalDigits>6</totalDigits>
         <fractionDigits>2</fractionDigits>
      </restriction>
   </simpleType>
   <simpleType name="char_8">
      <restriction base="string">
         <maxLength>8</maxLength>
      </restriction>
   </simpleType>
</schema>
```

The Supplier XML Document

```
<supplOrd>
   <wholesaler>Wholesale Chemical Supply</wholesaler>
   <supplier>Ready Reagents</supplier>
   <total>19872.43</total>
   <items>
      <item>
         <itemNo>SL35-014</itemNo>
         <qty>236</qty>
         <cost>17.97</unitCost>
      </item>
      <item>
         <itemNo>SL35-021</itemNo>
         <qty>35</qty>
         <cost>162.75</unitCost>
         .
         .
   </items>
</supplOrd>
```

Registering a Schema

- Before using an XML schema, it must be registered with the database using the DBMS_XMLSCHEMA package
- The two main functions of this package are the registerSchema and deleteSchema functions
- The XML schema source document can be defined as a VARCHAR, a CLOB, an XMLType or a URIType
- Documents can also be read directly from the file system

Registering an XML Schema

- RegisterSchema can be invoked from a PL/SQL procedure

```
BEGIN
   DBMS_XMLSCHEMA.registerSchema(
      'http://www.mfgsys.com/chemOrd.xsd',
      getDocument('SCHEMA_DIR', 'ChemOrd.xsd'),
      TRUE, TRUE, FALSE, FALSE) ;
END ;
```

- Registering an XML schema adds a resource to the XML DB repository
- Schema can be registered locally or globally

Registering an XML Schema

- By default, schemas are loaded locally and associated with the schema they are created under
- Local schemas are stored in the /sys/schemas/<username> directory

```
/sys/schemas/mfg/www.mfgsys.com/chemOrder.xsd
```

- Global schemas are stored in the /sys/schemas/PUBLIC directory

Creating a Valid Orders Table

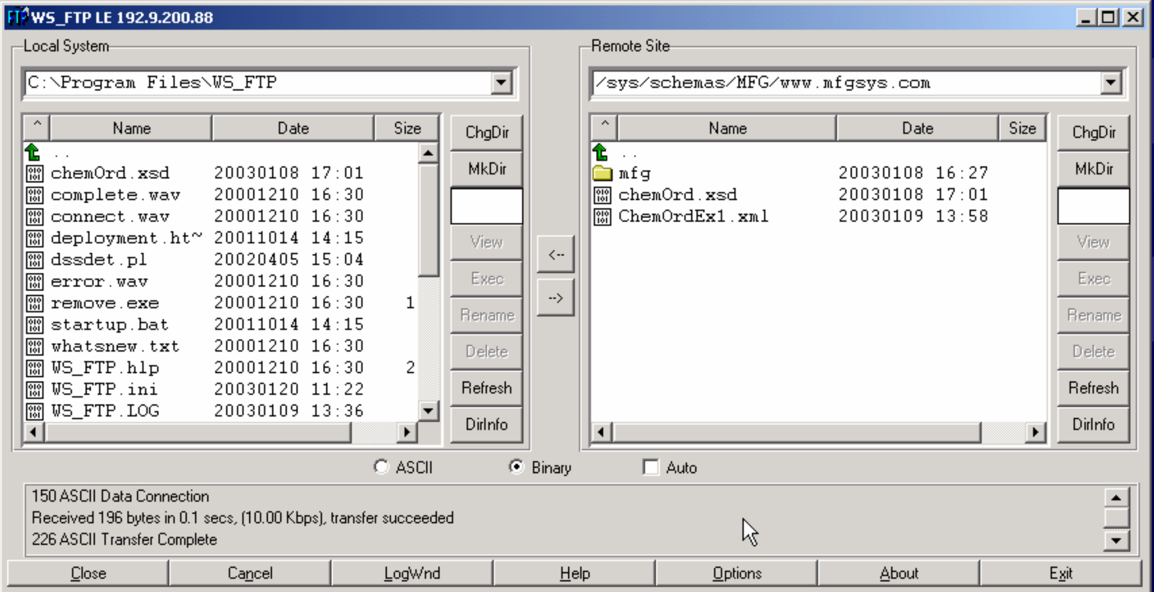
```
create table orders (
     orderNo          NUMBER(10),
     customerId       NUMBER(8),
     chemOrd          XMLTYPE,
     date_entered     DATE)

   XMLTYPE COLUMN chemOrd
     STORE AS CLOB (
        TABLESPACE ts_clob_ord
        STORAGE (INITIAL 4096  NEXT 4096)

   XMLTYPE COLUMN chemOrd
       XMLSCHEMA 'http://www.mfgsys.com/ChemOrd.xsd'
       ELEMENT 'chemOrd')
```

Step #1: FTPing the Orders

- XML DB includes an FTP server which listens on port 2100 by default
- The default can be changed by modifying the xdbconfig.xml file in the /sys folder



WS_FTP LE 192.9.200.88

Local System: C:\Program Files\WS_FTP

```
Name               Date            Size
..
chemOrd.xsd        20030108 17:01
complete.wav       20001210 16:30
connect.wav        20001210 16:30
deployment.ht~     20011014 14:15
dssdet.pl          20020405 15:04
error.wav          20001210 16:30
remove.exe         20001210 16:30    1
startup.bat        20011014 14:15
whatsnew.txt       20001210 16:30
WS_FTP.hlp         20001210 16:30    2
WS_FTP.ini         20030120 11:22
WS_FTP.LOG         20030109 13:36
```

Remote Site: /sys/schemas/MFG/www.mfgsys.com

```
Name               Date            Size
..
mfg                20030108 16:27
chemOrd.xsd        20030108 17:01
ChemOrdEx1.xml     20030109 13:58
```

○ ASCII ● Binary □ Auto

```
150 ASCII Data Connection
Received 196 bytes in 0.1 secs, (10.00 Kbps), transfer succeeded
226 ASCII Transfer Complete
```

Step #2: Validate incoming Orders

- The following procedure will insert a row into the Orders table:

```
DECLARE
   xmldoc  XMLType ;
BEGIN
   SELECT res INTO xmldoc
   FROM resource_view
   WHERE any_path =
   ‘/sys/schemas/MFG/www.mfgsys.com/ChemOrdEx1.xml’ ;

   INSERT INTO orders (orderNo, chemOrd, date_entered)
   VALUES (orderNo_seq.nextval, xmldoc, SYSDATE) ;
END ;
```

Step #3: Inserting valid orders

- The previous procedure will only work if the XML document is valid with respect to its XML schema
- If the XML document is not valid, Oracle throws an ORA-19007 exception

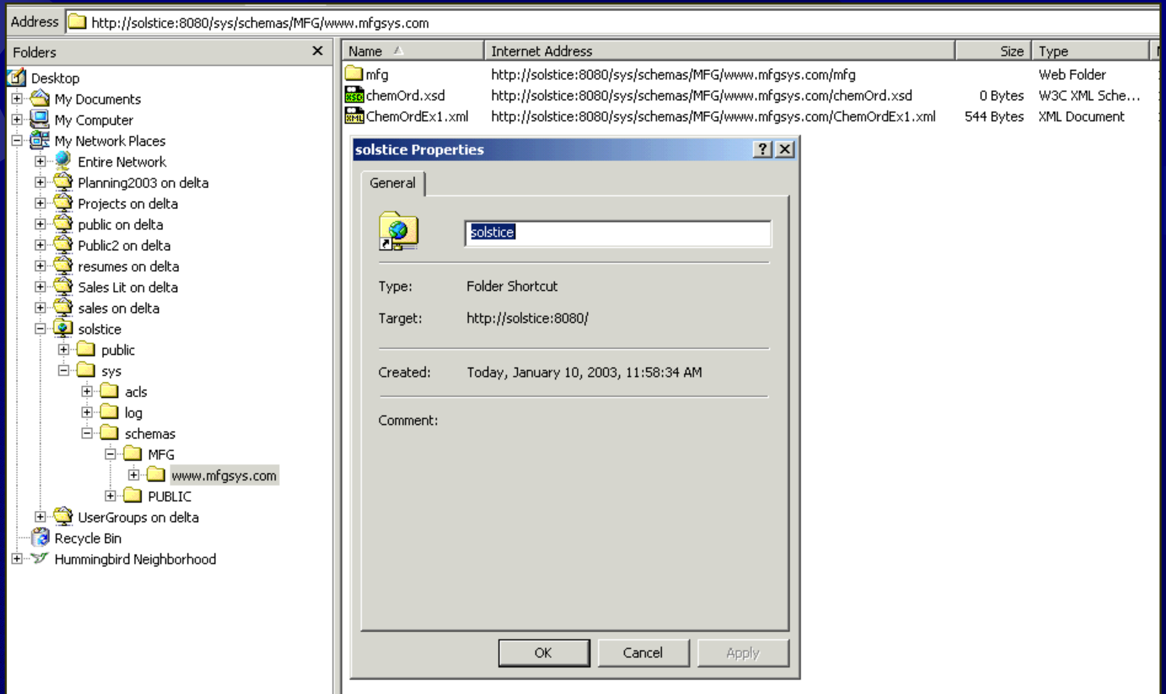
```
ORA-19007: Schema and element do not
           match
```

Step #4: Separating invalid orders

- This default mechanism is known as “full instance” validation
- Not very useful since it provides no control over handling errors
- Two other mechanisms for controlling invalid documents are a CHECK constraint or a BEFORE INSERT trigger
- A BEFORE INSERT triggers provides the most control

Step #5: Editing invalid orders

- The WebDAV protocol can be used to access the invalid XML file
- WebDAV is an HTTP extension and listens on port 8080 by default
- In Explorer, select “Map a Network Place”



Address: http://solstice:8080/sys/schemas/MFG/www.mfgsys.com

Name	Internet Address	Size	Type
mfg	http://solstice:8080/sys/schemas/MFG/www.mfgsys.com/mfg		Web Folder
chemOrd.xsd	http://solstice:8080/sys/schemas/MFG/www.mfgsys.com/chemOrd.xsd	0 Bytes	W3C XML Sche…
ChemOrdEx1.xml	http://solstice:8080/sys/schemas/MFG/www.mfgsys.com/ChemOrdEx1.xml	544 Bytes	XML Document

solstice Properties

General

solstice

Type: Folder Shortcut

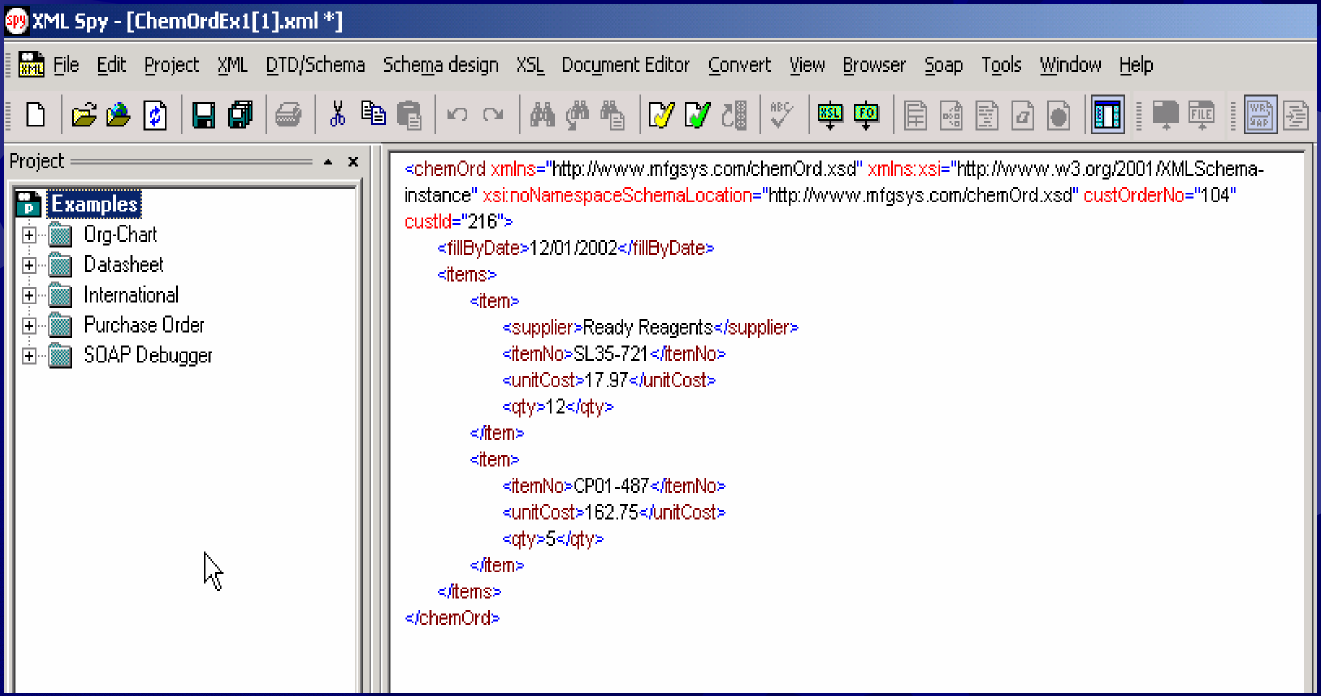
Target: http://solstice:8080/

Created: Today, January 10, 2003, 11:58:34 AM

Comment:

Step #5: Editing invalid orders

- Once the mapping has been established, the invalid XML document can be edited with XML Spy



```
<chemOrd xmlns="http://www.mfgsys.com/chemOrd.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="http://www.mfgsys.com/chemOrd.xsd" custOrderNo="104"
custId="216">
    <fillByDate>12/01/2002</fillByDate>
    <items>
        <item>
            <supplier>Ready Reagents</supplier>
            <itemNo>SL35-721</itemNo>
            <unitCost>17.97</unitCost>
            <qty>12</qty>
        </item>
        <item>
            <itemNo>CP01-487</itemNo>
            <unitCost>162.75</unitCost>
            <qty>5</qty>
        </item>
    </items>
</chemOrd>
```

Step #6: Generating an XML document

- The Supplier XML Order document summarizes all the orders for a given supplier on a specific date
- To extract XML document info, use the XMLType member functions like extract(), extractValue() and existsNode()
- The following query extracts all suppliers for a specific date:

```
SELECT extract(chemOrder,
   '/chemOrd/items//supplier').getStringVal()
FROM orders
WHERE date_entered = '12-JAN-2003'
```

Step #6: Generating an XML document

- To generate the supplOrd XML document SQL/XML functions are used
- These functions include:
  - XMLElement()
  - XMLForest()
  - XMLConcat
- The following function returns an XML document for a single supplier. It generated from an intermediate processing table

Step #6: Generating an XML document

```
create or replace function generateSupplOrdXML (
   iv_supplier VARCHAR2,
   iv_wholesaler VARCHAR2)
   RETURN VARCHAR2
IS
ln_total NUMBER := 0 ;
lv_xmldoc VARCHAR2(32000) ;
BEGIN
   /* First, generate the total */
   SELECT sum(qty*unitcost)
   INTO ln_total
   FROM suppTemp ;
```

Step #6: Generating an XML document

```
SELECT XMLELEMENT("supplOrd",
      XMLFOREST(iv_wholesaler AS "wholesaler",
                iv_supplier AS "supplier"),
      XMLELEMENT("total", to_char(ln_total)),
      XMLELEMENT("items",
         XMLAGG(XMLELEMENT("item",
               XMLCONCAT(XMLELEMENT("itemNo", itemNo),
                         XMLELEMENT("qty", sum(qty)),
                         XMLELEMENT("cost",
                                    sum(qty*unitcost))
               ))))).getStringVal()
   INTO lv_xmldoc
   FROM suppTemp
   WHERE supplier = iv_supplier
   GROUP BY supplier, itemNo ;
   RETURN lv_xmldoc ;
END ;
```

References

1. “Getting Started with Oracle 9i’s XML DB Facility”, Jeff Bernknopf, NYOUG Technical Journal, Quarter 2, 2003
2. The W3C XML 1.0 Recommendation (http://www.w3.org/TR/xml-rec)
3. The XSLT 1.0 recommendation (http://www.w3.org/TR/xslt)
4. The W3C XML Schema Recommendation (http://www.w3.org/TR/xmlschema-1/ and http://www.w3.org/TR/xmlschema-2/)
5. Oracle 9i XML Database Developer's Guide: Appendix G – Example Setup Scripts. (http://otn.oracle.com/tech/xml/doc.html)
6. Altova's XML Spy provides an integrated XML IDE (http://www.xmlspy.com) (MFG is an authorized reseller)
7. Latest working draft of the SQL/XML group (http://sqlx.org/5wd-14-xml-2002-08.pdf)

Additional Information

- For a copy of the “Getting Started with XML DB” and “Developing XML DB Applications” articles, please send an email to:

  jeff@mfgsys.com

- You can also email me with any additional questions