DB2 Data Management Software

IBM

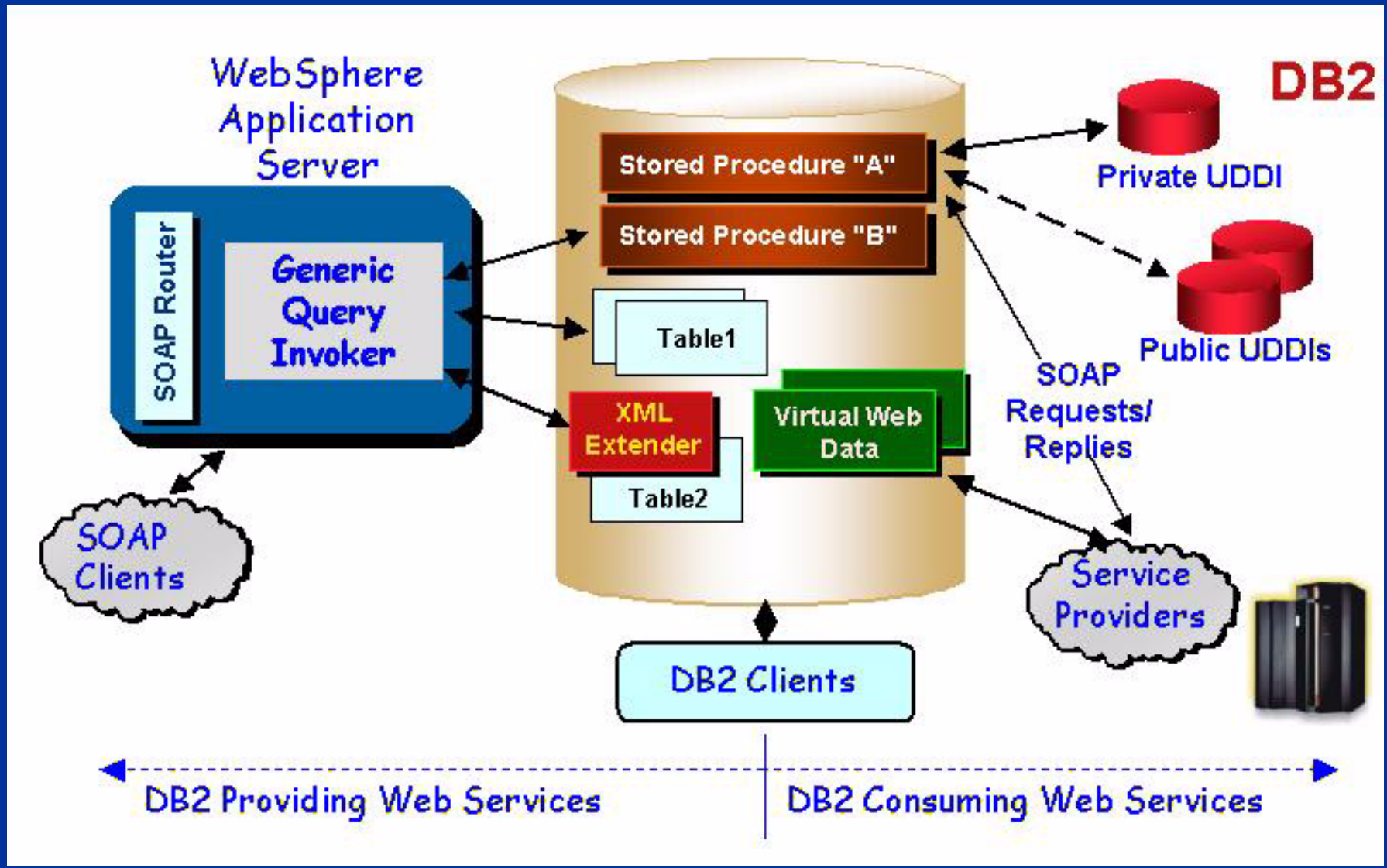
e business software

DB2 and Web Services

Susan Malaika

IBM Software Group

DB2's place within Web services



DB2 providing Web services

- DB2 Web Services enable Web Services client to invoke

[1] Any DB2 stored procedures (SPs)
- output parameters and result sets are returned

[2] SQL requests
- results are tagged using default tagging
- focus is data in and out of the database rather than the format

[3] The XML Extender DAD for
- Advanced tagging of XML results from SQL
- XML shredding into relational data
- XML extender functions for advanced manipulation of XML

Currently available on DB2 Unix, Windows - Will be available on OS/390 soon

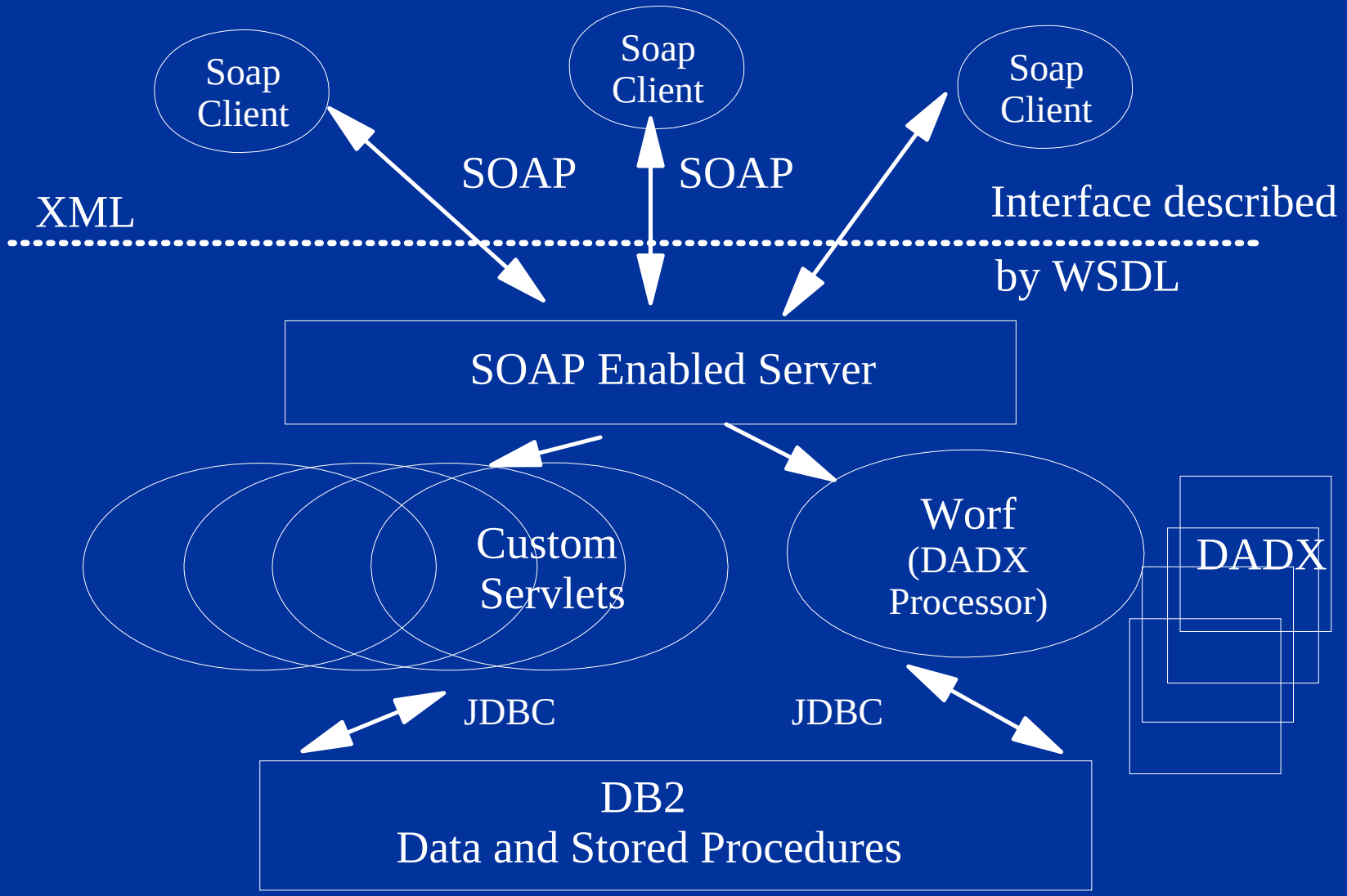
DB2 consuming Web services

- Future direction : User defined functions (UDFs) to invoke web services by issuing SOAP requests
- For example, using a stock quote within a SQL request

DB2 Web Service Support

- DADX: Document Access Definition Extension:
  - Produced in Websphere Studio or through a regular text editor
  - Generates WSDL and XML schema (xsd) to describe a DB2 Web Service
  - Used in conjunction with worf (Web services Object Runtime Framework) also known as the DADX processor to enable DB2 to be a Web Service Provider
  - DADX support is available through Websphere Studio or through Web download from the DB2 XML Extender Website
- DAD: Document Access Definition:
  - Provides advanced mapping between XML and relational data, e.g.,
    - Control where element repetition takes place
    - Control of the number of documents produced
  - DAD support is part of the DB2 XML Extender

Two options: Custom Servlets or DADX Files



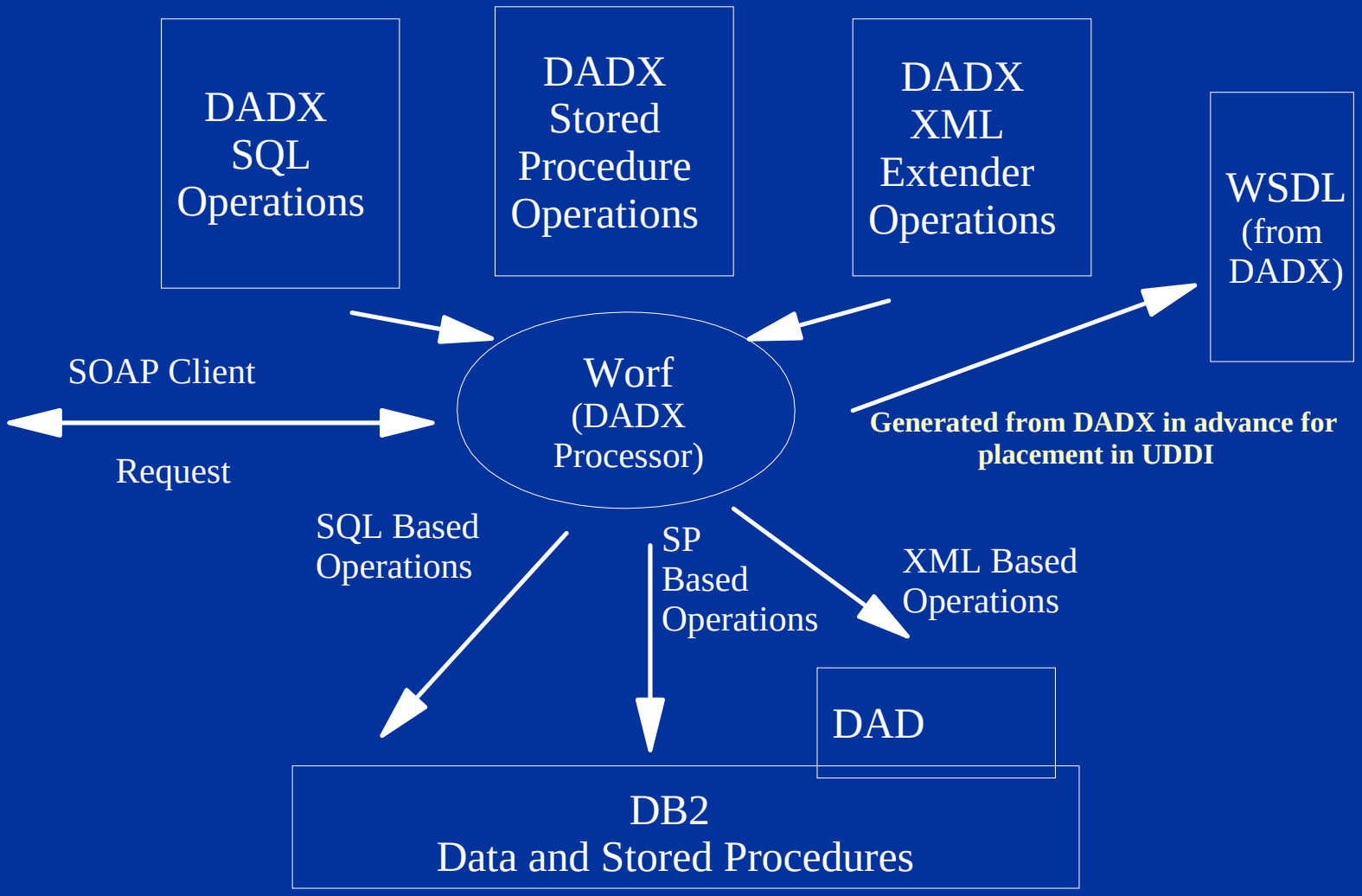
DADX Processor (WORF)

- Provides the run time environment for web services defined by DADX
  - Available through Websphere Studio
  - Or as web download from DB2 XML Extender website
    http://www.ibm.com/software/data/db2/extenders/xmlext/docs/v72wrk/WORF.html
- Can be used with WAS V4 or Apache Tomcat
- Requires JDBC 2.0 ie DB2 Version 7
- Overview paper available from:
  http://www.ibm.com/software/data/pubs/papers/db2webservices/db2webservices.pdf

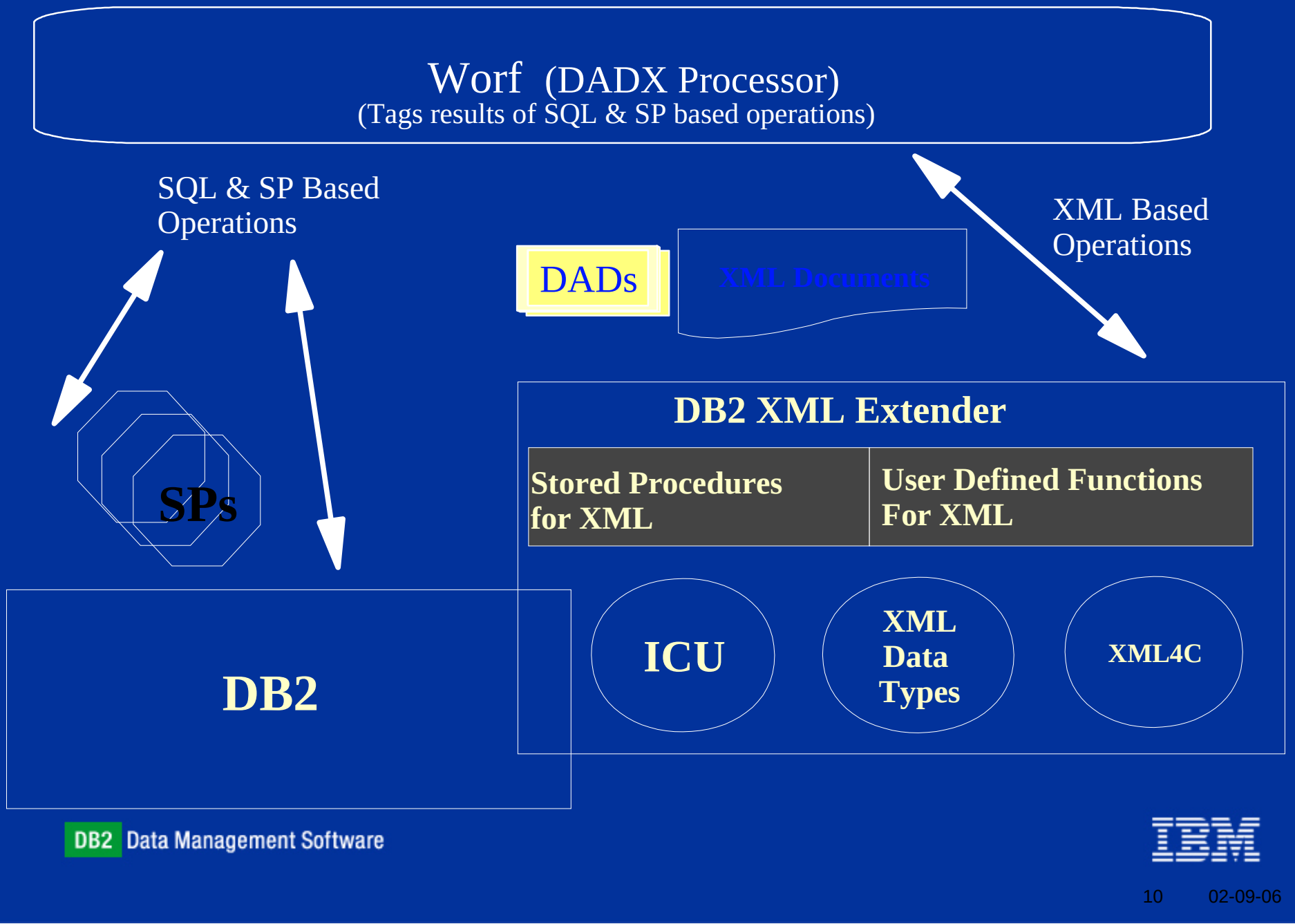
WORF provides support for

- Development
  - Web Service test facility over http (Test page generation from DADX)
- Web Service Generation from DADX
  - WSDL generation (includes comments supplied in DADX)
  - XML schema generation
- Execution time configuration (e.g., through group.properties file)
  - Connection pooling
  - Security specification

Three DADX Operation Types



DB2 Web Services and XML Extender Components



DADX (DAD eXtension)

- Defines the operations that can be performed by the web service
- Can be built using web service and query wizards in Websphere Studio or any text editor
- Specifies how to create a web service using a set of operations that are defined by SQL statements and optionally DAD files
- Provides two types of operations
  - XML collection operations
  - SQL operations (including calls to stored procedures)

Simple DADX file

```
<?xml version="1.0" encoding="UTF-8"?>
  <DADX xmlns="http://schemas.ibm.com/db2/dxx/dadx">

    <operation name="listDepartments">
      <query>
        <SQL_query>SELECT * FROM DEPARTMENT
                          WHERE LOCATION = :location
        </SQL_query>
        <parameter name ="location" type="xsd.string"/>
      </query>
    </operation>
  </DADX>
```

Specifies type of operation

Parameter passed to webservice when invoked

Element must be defined for each parameter

Operation types:

<retrievexml>	<query>
<storexml>	<update> (for Insert, Update, Delete statements)
	<call> (for Stored procedures - parameters must be defined as in, out or in/out NB result sets are supported)

DADX file that generates XML document from DB2

```
<?xml version="1.0"?>
 <DADX xmlns="http://schemas.ibm.com/db2/dxx/dadx"
       xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                      xmlns="http://www.w3.org/1999/xhtml">
    Provides queries for part order information at myco.com.
    See <a href="../documentation/PartOrders.html" target="_top">
        PartOrders.html</a> for more information.
    </wsdl:documentation>
   <operation name="findByExtendedPriceAndShipDate">
    <wsdl:documentation xmlns="http://schemas.xmlsoap.org/wsdl/">
     Returns all the orders with an extended price greater than $50.00
     and a ship date later than 1998-12-01.
    </wsdl:documentation>
    <retrieveXML>
      <DAD_ref>order_rdb.dad</DAD_ref>
       <XML_override>
         /Order/Part/ExtendedPrice > 50.00 AND
         Order/Part/Shipment/ShipDate > '1998-12-01'
       </XML_override>
    </retrieveXML>
   </operation>
</DADX>
```

Namespaces

Calls the dxxGenXML stored procedure as DAD is being passed

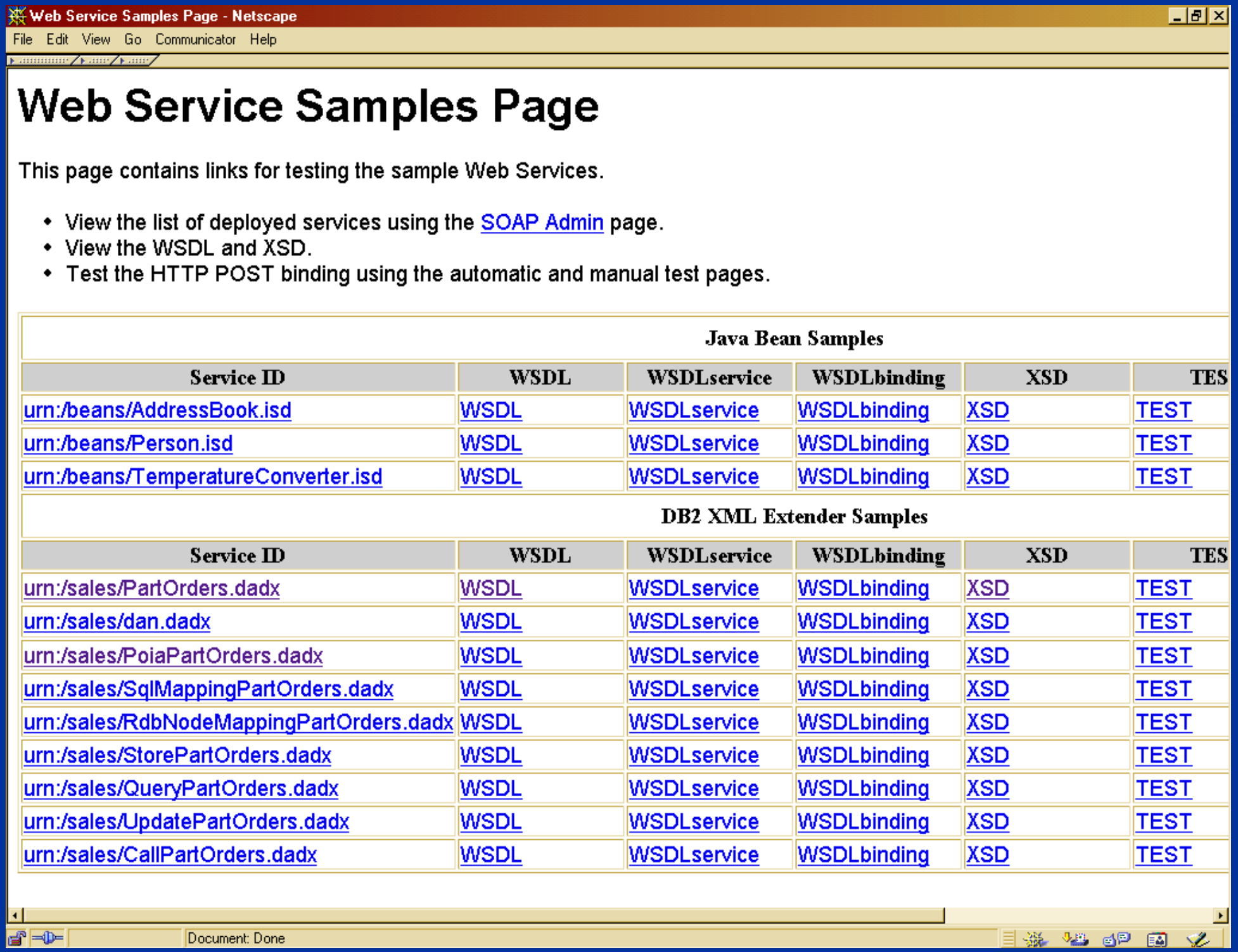
DAD

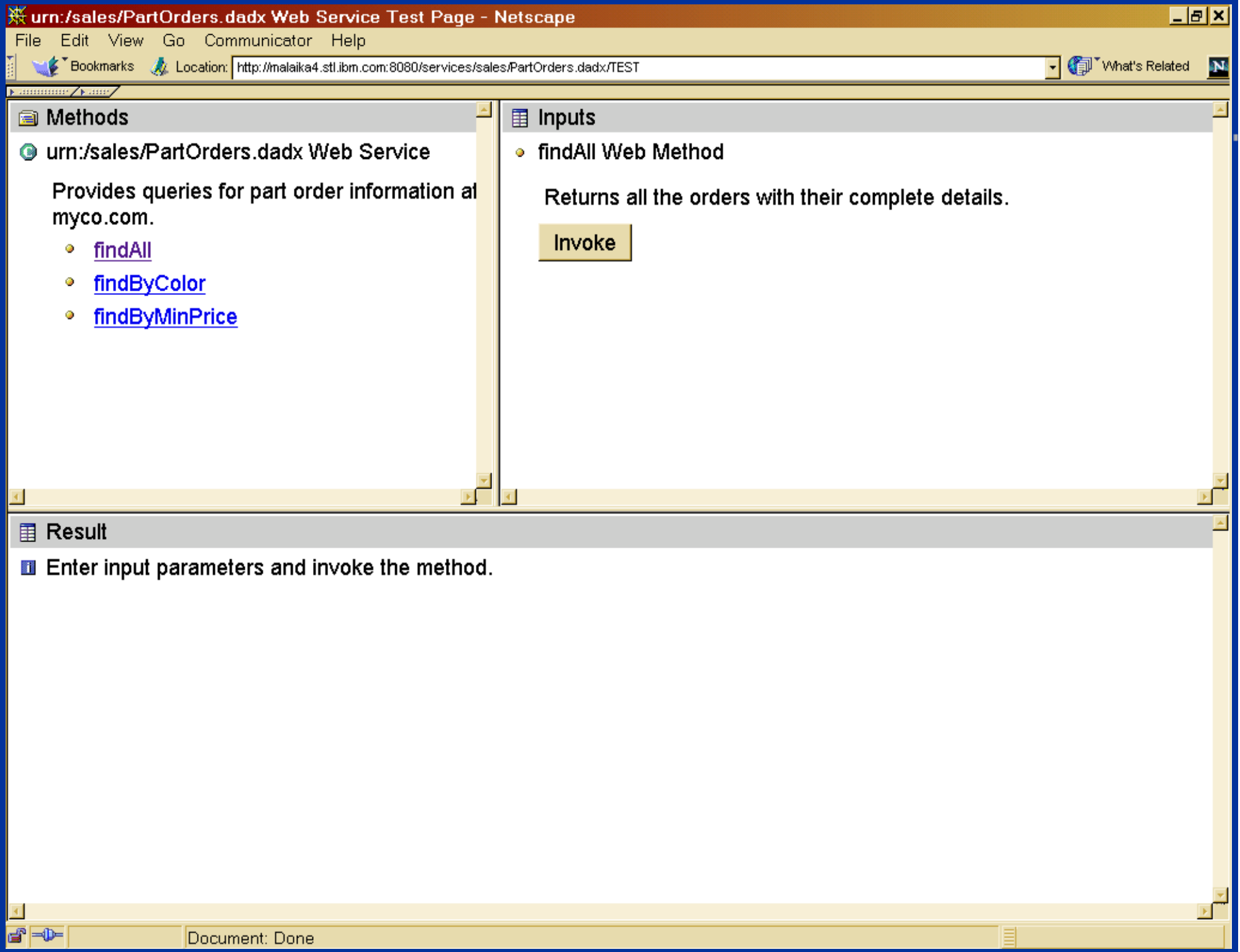
Alternatives are 'no override' or 'SQL override'

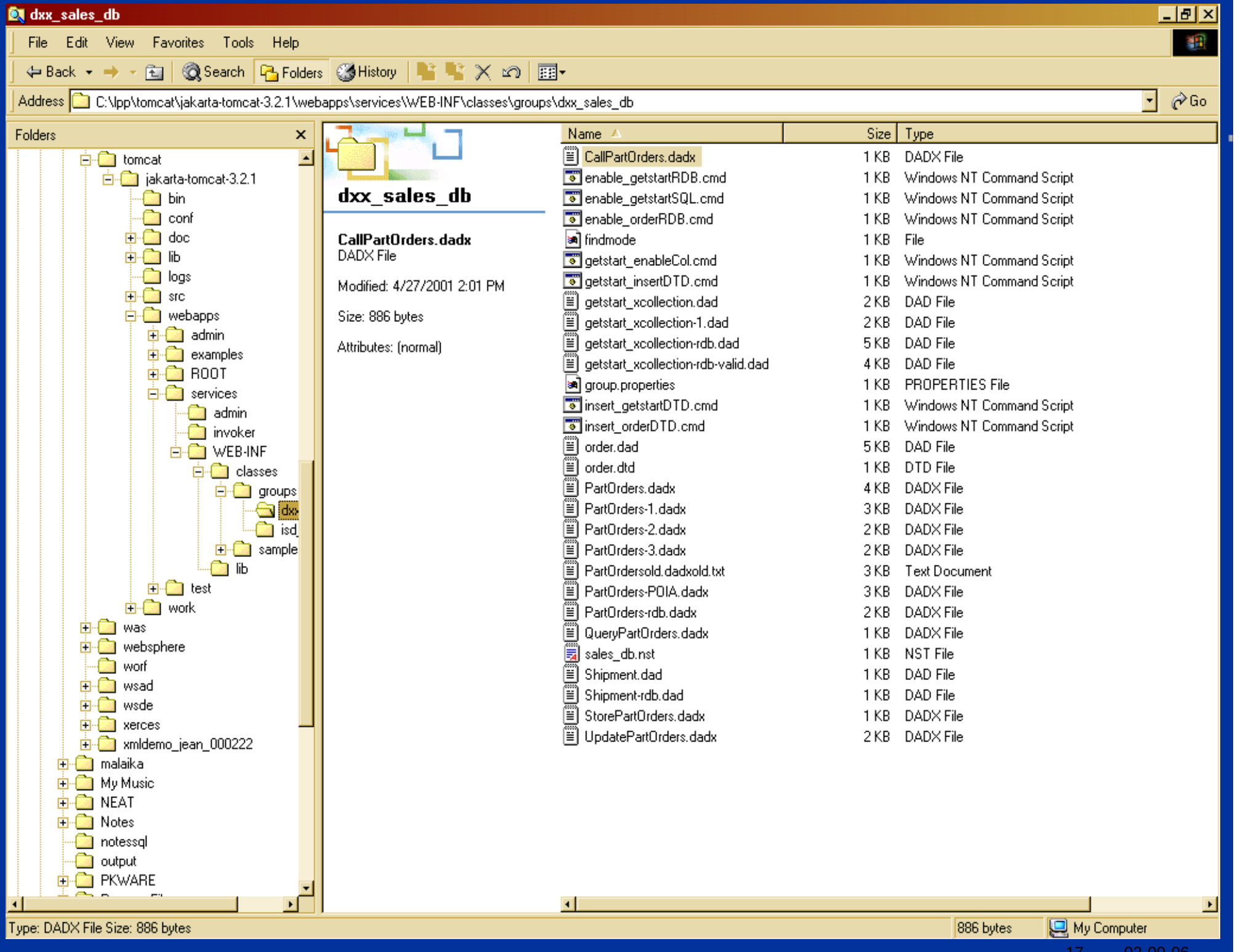
NB Would invoke dxxRetrieveXML if a collection was passed

Steps for using DB2 Web Services

- Show standard test page that is shipped as part of worf
- Show directories in Apache Tomcat (similar in WebSphere)
- Show DADX sample PartOrders.dadx
  - Three operations (transactions) in the sample
    - findall, findByColor, findByMinPrice
- Show DAD
  - To describe the shape of the generated document
- Show worf generating from the DADX:
  - WSDL for a Web Service (SOAP and HTTP bindings)
  - XSD for the interface (XML Schema)
- Show Web Service execution and XML results
- Show adding findByMode operation to the PartOrder.dadx
- Show directions for DB2 Web Services
- Show Resources (URLs) for further information


Web Service Samples Page - Netscape
File Edit View Go Communicator Help
Web Service Samples Page
This page contains links for testing the sample Web Services.
View the list of deployed services using the SOAP Admin page.
View the WSDL and XSD.
Test the HTTP POST binding using the automatic and manual test pages.
Java Bean Samples
Service ID WSDL WSDLservice WSDLbinding XSD TES
urn:/beans/AddressBook.isd WSDL WSDLservice WSDLbinding XSD TEST
urn:/beans/Person.isd WSDL WSDLservice WSDLbinding XSD TEST
urn:/beans/TemperatureConverter.isd WSDL WSDLservice WSDLbinding XSD TEST
DB2 XML Extender Samples
Service ID WSDL WSDLservice WSDLbinding XSD TES
urn:/sales/PartOrders.dadx WSDL WSDLservice WSDLbinding XSD TEST
urn:/sales/dan.dadx WSDL WSDLservice WSDLbinding XSD TEST
urn:/sales/PoiaPartOrders.dadx WSDL WSDLservice WSDLbinding XSD TEST
urn:/sales/SqlMappingPartOrders.dadx WSDL WSDLservice WSDLbinding XSD TEST
urn:/sales/RdbNodeMappingPartOrders.dadx WSDL WSDLservice WSDLbinding XSD TEST
urn:/sales/StorePartOrders.dadx WSDL WSDLservice WSDLbinding XSD TEST
urn:/sales/QueryPartOrders.dadx WSDL WSDLservice WSDLbinding XSD TEST
urn:/sales/UpdatePartOrders.dadx WSDL WSDLservice WSDLbinding XSD TEST
urn:/sales/CallPartOrders.dadx WSDL WSDLservice WSDLbinding XSD TEST
Document: Done


urn:/sales/PartOrders.dadx Web Service Test Page - Netscape
File Edit View Go Communicator Help
Bookmarks Location: http://malaika4.stl.ibm.com:8080/services/sales/PartOrders.dadx/TEST
What's Related
Methods
urn:/sales/PartOrders.dadx Web Service
Provides queries for part order information at myco.com.
findAll
findByColor
findByMinPrice
Inputs
findAll Web Method
Returns all the orders with their complete details.
Invoke
Result
Enter input parameters and invoke the method.
Document: Done


dxx_sales_db
File Edit View Favorites Tools Help
Back Search Folders History
Address C:\lpp\tomcat\jakarta-tomcat-3.2.1\webapps\services\WEB-INF\classes\groups\dxx_sales_db
Go
Folders
tomcat
jakarta-tomcat-3.2.1
bin
conf
doc
lib
logs
src
webapps
admin
examples
ROOT
services
admin
invoker
WEB-INF
classes
groups
dxx
isd_
sample
lib
test
work
was
websphere
worf
wsad
wsde
xerces
xmldemo_jean_000222
malaika
My Music
NEAT
Notes
notessql
output
PKWARE
dxx_sales_db
CallPartOrders.dadx
DADX File
Modified: 4/27/2001 2:01 PM
Size: 886 bytes
Attributes: (normal)
Name
Size
Type
CallPartOrders.dadx 1 KB DADX File
enable_getstartRDB.cmd 1 KB Windows NT Command Script
enable_getstartSQL.cmd 1 KB Windows NT Command Script
enable_orderRDB.cmd 1 KB Windows NT Command Script
findmode 1 KB File
getstart_enableCol.cmd 1 KB Windows NT Command Script
getstart_insertDTD.cmd 1 KB Windows NT Command Script
getstart_xcollection.dad 2 KB DAD File
getstart_xcollection-1.dad 2 KB DAD File
getstart_xcollection-rdb.dad 5 KB DAD File
getstart_xcollection-rdb-valid.dad 4 KB DAD File
group.properties 1 KB PROPERTIES File
insert_getstartDTD.cmd 1 KB Windows NT Command Script
insert_orderDTD.cmd 1 KB Windows NT Command Script
order.dad 5 KB DAD File
order.dtd 1 KB DTD File
PartOrders.dadx 4 KB DADX File
PartOrders-1.dadx 3 KB DADX File
PartOrders-2.dadx 2 KB DADX File
PartOrders-3.dadx 2 KB DADX File
PartOrdersold.dadxold.txt 3 KB Text Document
PartOrders-POIA.dadx 3 KB DADX File
PartOrders-rdb.dadx 2 KB DADX File
QueryPartOrders.dadx 1 KB DADX File
sales_db.nst 1 KB NST File
Shipment.dad 1 KB DAD File
Shipment-rdb.dad 1 KB DAD File
StorePartOrders.dadx 1 KB DADX File
UpdatePartOrders.dadx 2 KB DADX File
Type: DADX File Size: 886 bytes
886 bytes
My Computer

DADX Example PartOrders.dadx – Part 1

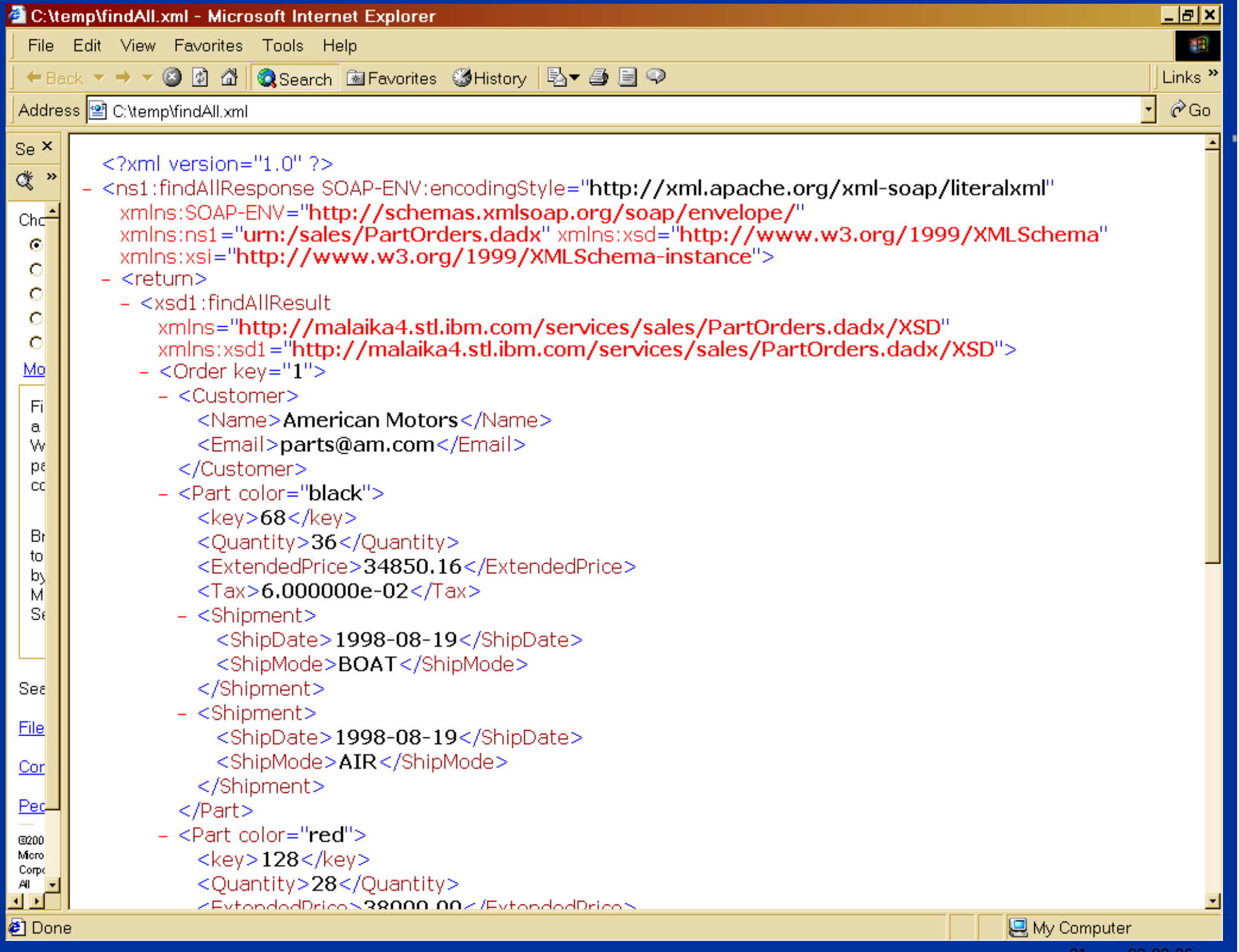
```
<?xml version="1.0"?>
<DADX xmlns="urn:ibm.com:dxx:dadx"
    xmlns:xsd="http://www.w3.org/1999/XMLSchema">
 <documentation <documentation xmlns="http://schemas.xmlsoap.org/wsdl/">
 Provides queries for part order information at myco.com.</documentation>

 <operation name="findAll">
  <documentation xmlns="http://schemas.xmlsoap.org/wsdl/">
      Returns all the orders with their complete details.</documentation>
  <retrieveXML>
    <DAD_ref>getstart_xcollection.dad</DAD_ref>
    <SQL_override>select o.order_key, customer_name, customer_email,
     p.part_key, color, quantity, price, tax, ship_id, date, mode  from order_tab o,
                                                          part_tab p,
     table(select substr(char(timestamp(generate_unique())),16) as ship_id,
     date, mode, part_key from ship_tab) s
     where p.order_key = o.order_key and s.part_key = p.part_key
     order by order_key, part_key, ship_id
    </SQL_override>
  </retrieveXML>
 </operation>
```

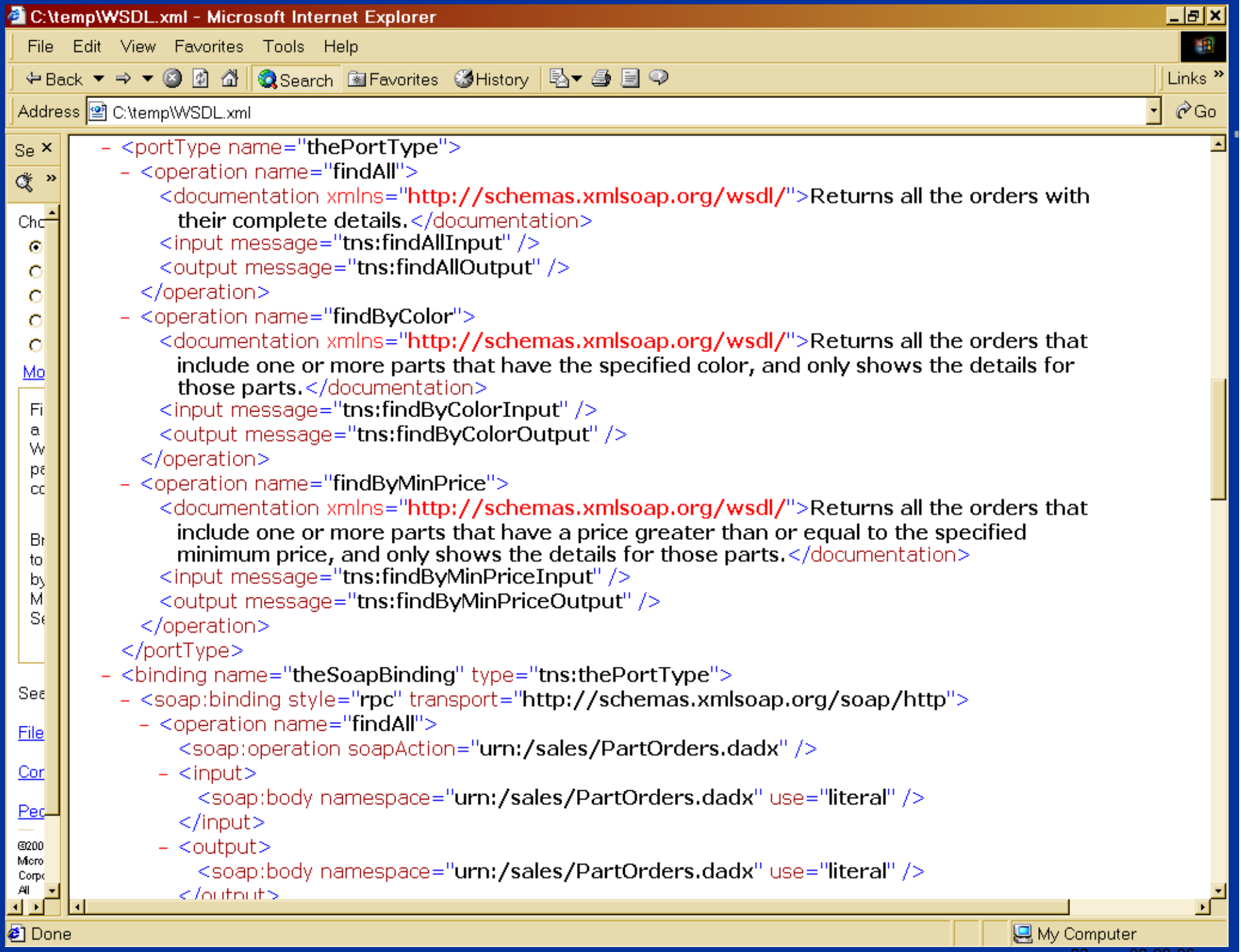
DADX Example PartOrders.dadx – Part 2

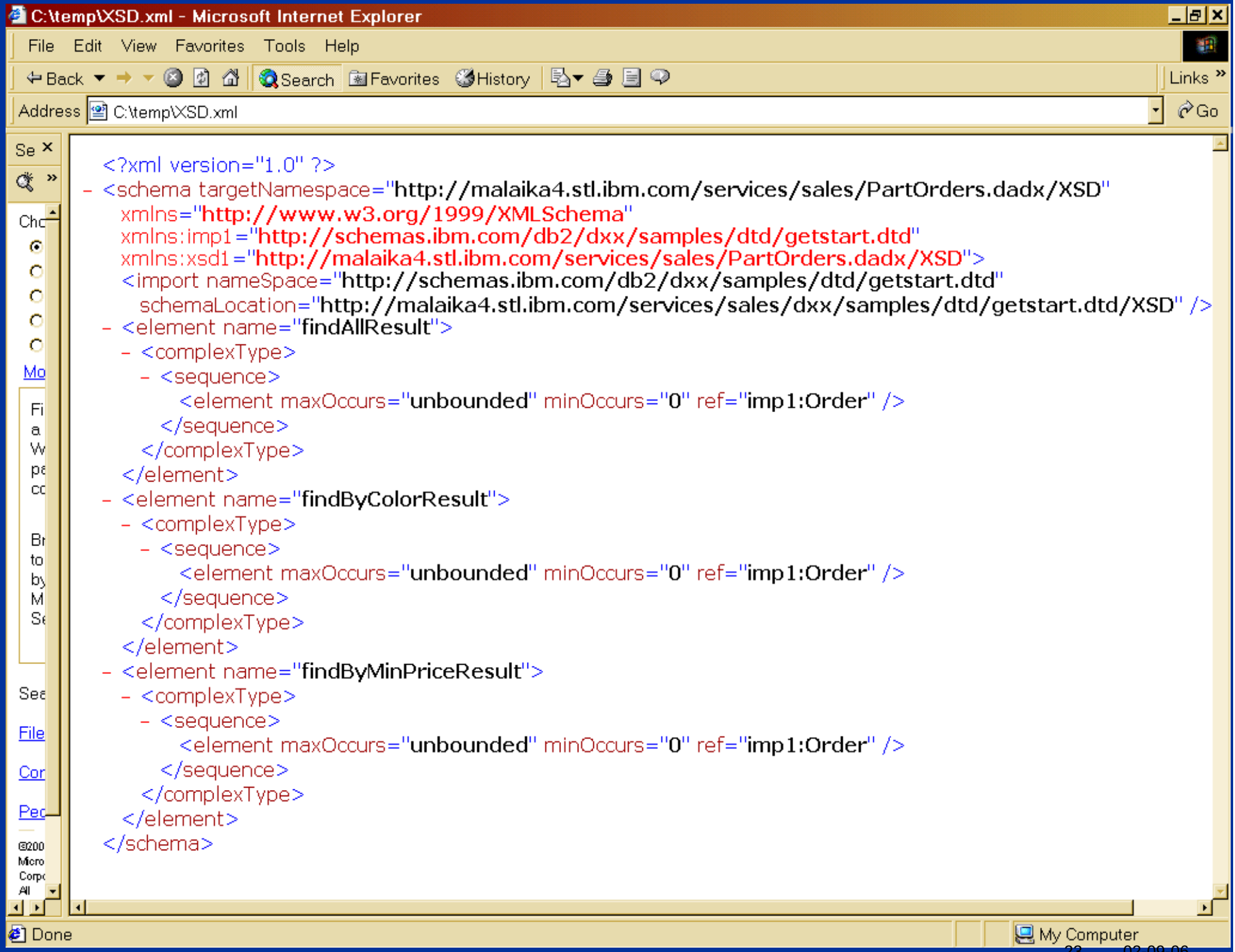
```
<operation name="findByColor">
  <documentation xmlns="http://schemas.xmlsoap.org/wsdl/">Returns all the orders
   that include one or more parts that have the specified color, and only shows
   the details for those parts.</documentation>
  <retrieveXML>
   <DAD_ref>getstart_xcollection.dad</DAD_ref>
   <SQL_override>
    select o.order_key, customer_name, customer_email,
      p.part_key, color, quantity, price, tax, ship_id, date, mode
    from order_tab o, part_tab p,
      table(select substr(char(timestamp(generate_unique())),16) as ship_id,
        date, mode, part_key from ship_tab) s
    where p.order_key = o.order_key and s.part_key = p.part_key
      and color = :color
    order by order_key, part_key, ship_id
   </SQL_override>
   <parameter name="color" type="xsd:string"/>
  </retrieveXML>
</operation>
```

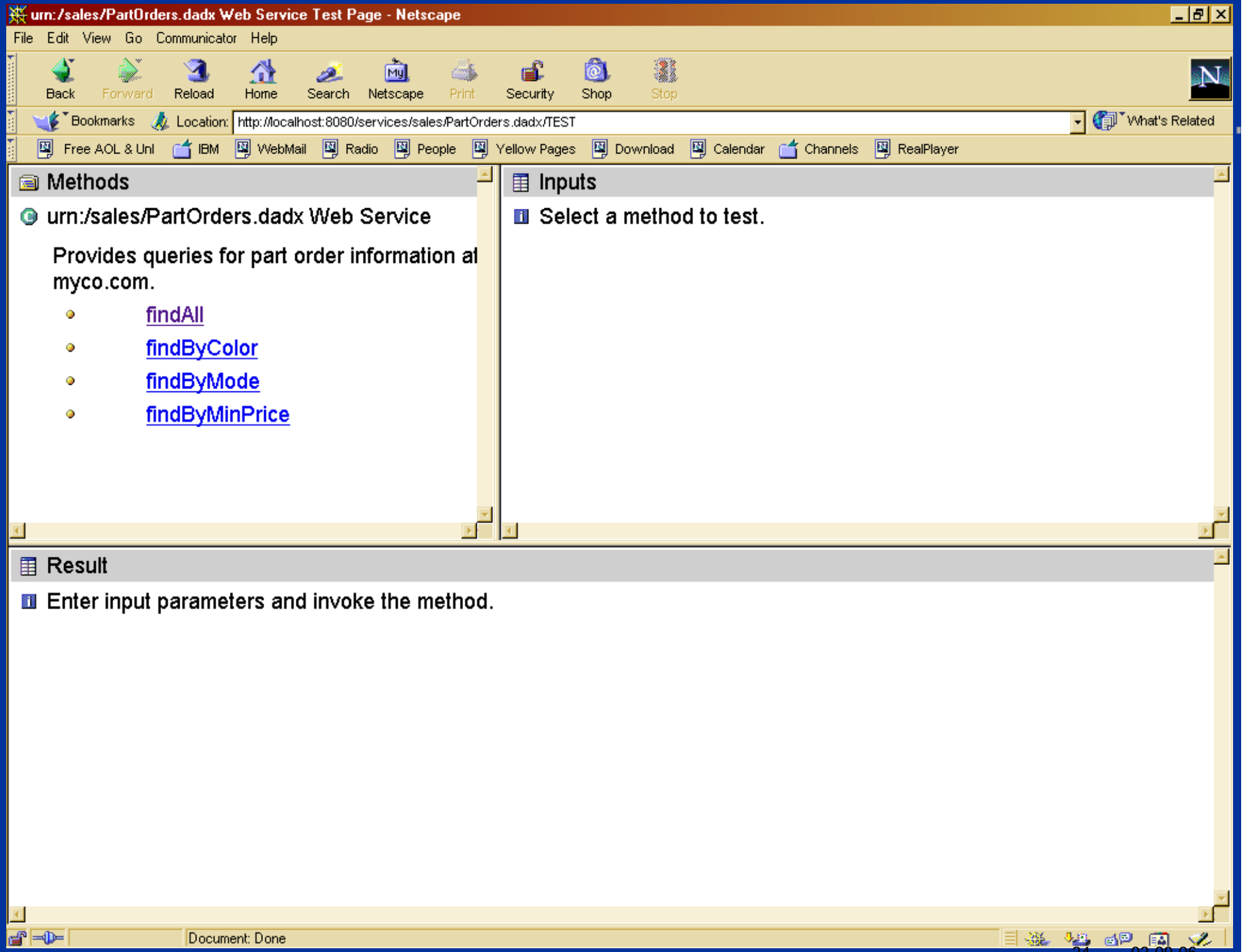
DADX Example PartOrders.dadx – Part 3

```
<operation name="findByMinPrice">
  <documentation xmlns="http://schemas.xmlsoap.org/wsdl/">
   Returns all the orders that include one or more parts that have a
  price greater than or equal to the specified minimum price, and only shows the details for
   those parts.</documentation>
  <retrieveXML>
    <DAD_ref>getstart_xcollection.dad</DAD_ref>
    <SQL_override>
     select o.order_key, customer_name, customer_email,
       p.part_key, color, quantity, price, tax, ship_id, date, mode
     from order_tab o, part_tab p,
       table(select substr(char(timestamp(generate_unique())),16) as ship_id,
         date, mode, part_key from ship_tab) s
     where p.order_key = o.order_key and s.part_key = p.part_key
       and p.price >= :minprice
     order by order_key, part_key, ship_id
    </SQL_override>
    <parameter name="minprice" type="xsd:decimal"/>
  </retrieveXML>
 </operation>
</DADX>
```


C:\temp\findAll.xml - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History Links
Address C:\temp\findAll.xml
Go
<?xml version="1.0" ?>
<ns1:findAllResponse SOAP-ENV:encodingStyle="http://xml.apache.org/xml-soap/literalxml"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:/sales/PartOrders.dadx" xmlns:xsd="http://www.w3.org/1999/XMLSchema"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance">
<return>
<xsd1:findAllResult
xmlns="http://malaika4.stl.ibm.com/services/sales/PartOrders.dadx/XSD"
xmlns:xsd1="http://malaika4.stl.ibm.com/services/sales/PartOrders.dadx/XSD">
<Order key="1">
<Customer>
<Name>American Motors</Name>
<Email>parts@am.com</Email>
</Customer>
<Part color="black">
<key>68</key>
<Quantity>36</Quantity>
<ExtendedPrice>34850.16</ExtendedPrice>
<Tax>6.000000e-02</Tax>
<Shipment>
<ShipDate>1998-08-19</ShipDate>
<ShipMode>BOAT</ShipMode>
</Shipment>
<Shipment>
<ShipDate>1998-08-19</ShipDate>
<ShipMode>AIR</ShipMode>
</Shipment>
</Part>
<Part color="red">
<key>128</key>
<Quantity>28</Quantity>
Done
My Computer


C:\temp\WSDL.xml - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History Links
Address C:\temp\WSDL.xml
Go
- <portType name="thePortType">
- <operation name="findAll">
<documentation xmlns="http://schemas.xmlsoap.org/wsdl/">Returns all the orders with their complete details.</documentation>
<input message="tns:findAllInput" />
<output message="tns:findAllOutput" />
</operation>
- <operation name="findByColor">
<documentation xmlns="http://schemas.xmlsoap.org/wsdl/">Returns all the orders that include one or more parts that have the specified color, and only shows the details for those parts.</documentation>
<input message="tns:findByColorInput" />
<output message="tns:findByColorOutput" />
</operation>
- <operation name="findByMinPrice">
<documentation xmlns="http://schemas.xmlsoap.org/wsdl/">Returns all the orders that include one or more parts that have a price greater than or equal to the specified minimum price, and only shows the details for those parts.</documentation>
<input message="tns:findByMinPriceInput" />
<output message="tns:findByMinPriceOutput" />
</operation>
</portType>
- <binding name="theSoapBinding" type="tns:thePortType">
- <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http">
- <operation name="findAll">
<soap:operation soapAction="urn:/sales/PartOrders.dadx" />
- <input>
<soap:body namespace="urn:/sales/PartOrders.dadx" use="literal" />
</input>
- <output>
<soap:body namespace="urn:/sales/PartOrders.dadx" use="literal" />
Done
My Computer


C:\temp\XSD.xml - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History Links
Address C:\temp\XSD.xml
Go
<?xml version="1.0" ?>
<schema targetNamespace="http://malaika4.stl.ibm.com/services/sales/PartOrders.dadx/XSD"
xmlns="http://www.w3.org/1999/XMLSchema"
xmlns:imp1="http://schemas.ibm.com/db2/dxx/samples/dtd/getstart.dtd"
xmlns:xsd1="http://malaika4.stl.ibm.com/services/sales/PartOrders.dadx/XSD">
<import nameSpace="http://schemas.ibm.com/db2/dxx/samples/dtd/getstart.dtd"
schemaLocation="http://malaika4.stl.ibm.com/services/sales/dxx/samples/dtd/getstart.dtd/XSD" />
<element name="findAllResult">
<complexType>
<sequence>
<element maxOccurs="unbounded" minOccurs="0" ref="imp1:Order" />
</sequence>
</complexType>
</element>
<element name="findByColorResult">
<complexType>
<sequence>
<element maxOccurs="unbounded" minOccurs="0" ref="imp1:Order" />
</sequence>
</complexType>
</element>
<element name="findByMinPriceResult">
<complexType>
<sequence>
<element maxOccurs="unbounded" minOccurs="0" ref="imp1:Order" />
</sequence>
</complexType>
</element>
</schema>
Done
My Computer


urn:/sales/PartOrders.dadx Web Service Test Page - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://localhost:8080/services/sales/PartOrders.dadx/TEST
What's Related
Free AOL & Unl IBM WebMail Radio People Yellow Pages Download Calendar Channels RealPlayer
Methods
urn:/sales/PartOrders.dadx Web Service
Provides queries for part order information at myco.com.
findAll
findByColor
findByMode
findByMinPrice
Inputs
Select a method to test.
Result
Enter input parameters and invoke the method.
Document: Done

Add findbyMode operation in PartOrders.dadx

Modifying a DB2 Web Service is simple!

```
<operation name="findByMode">
  <documentation xmlns="http://schemas.xmlsoap.org/wsdl/">Returns  orders by
       ode</documentation>
  <retrieveXML>
   <DAD_ref>getstart_xcollection.dad</DAD_ref>
   <SQL_override>   select o.order_key, customer_name, customer_email,  p.part_key, color,
     quantity, price, tax, ship_id, date, mode  from order_tab o, part_tab p,
     table(select substr(char(timestamp(generate_unique())),16) as ship_id,
      date, mode, part_key from ship_tab) s
     where p.order_key = o.order_key and s.part_key = p.part_key and mode = :mode
    order by order_key, part_key, ship_id </SQL_override>
   <parameter name="mode" type="xsd:string"/>
  </retrieveXML>
 </operation>
```

Building a DB2 Web Service - Steps to follow

- The WORF readme describes in detail how to manually
  - 1 : Define a group of webservices by editing files
    - web.XML in WEB-INF directory and group.properties in a group directory and a group.properties file
  - 2: Install WORF on WAS V4 (or Tomcat)
  - 3: Create and deploy a web service
    - Create a DADX file using a text editor
    - Generate a deployment descriptor using java command and copy isd file
    - Restart web application and use sample page to test
    - Generate schema file (XSD) to define data types used in the web services interface as WSDL uses schema rather than DTDs
    - Generate WSDL for UDDI registration

Demos

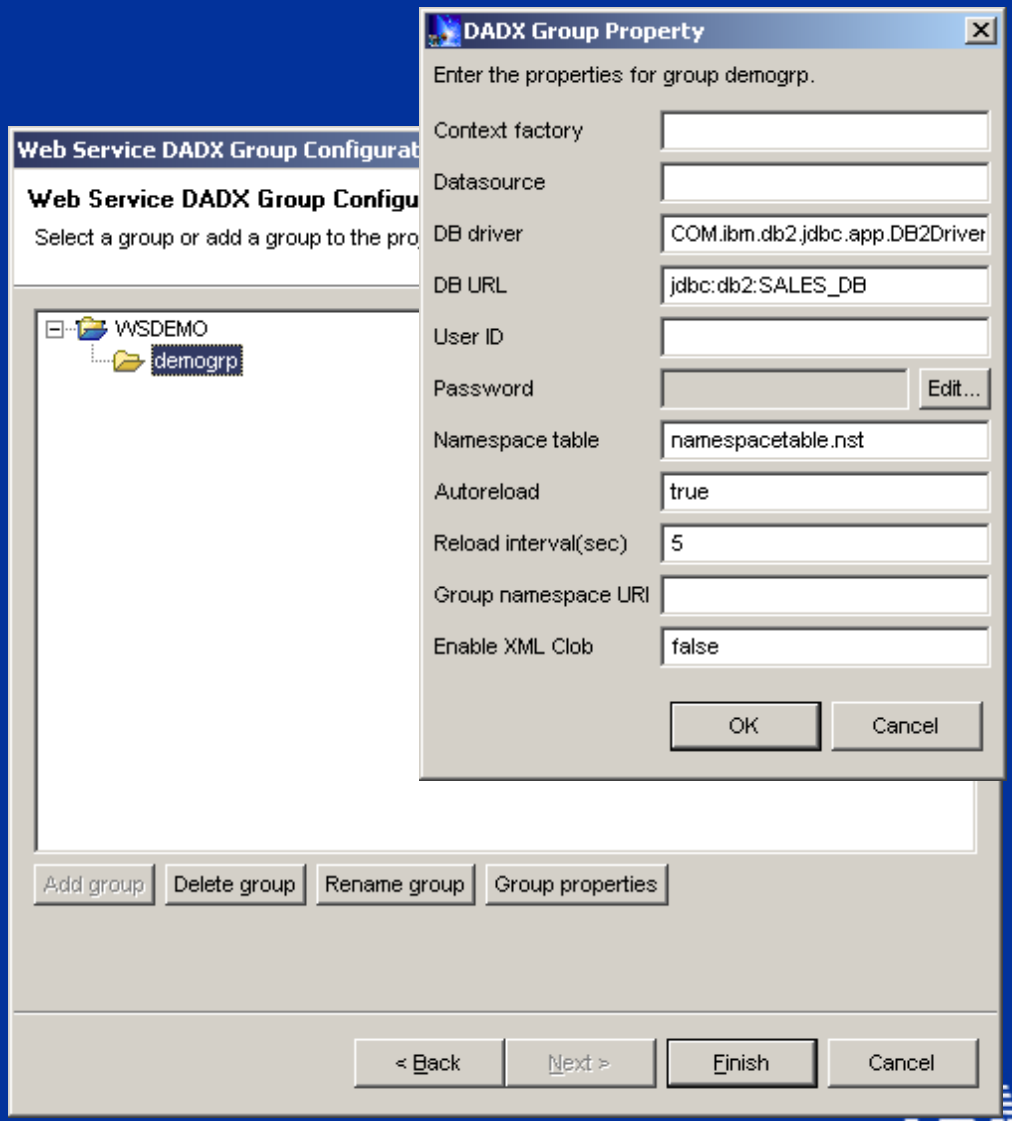
Using Websphere Studio
Video Central

Building... Using Websphere Studio

- Websphere Studio : IDE for building and testing J2EE applications
- Wizards and tools include
  - Relational Schema Center : imports the database schema into the project
  - SQL to XML Wizard : assembles SQL statements into DADX documents
  - DADX Group Configuration Wizard: for setting up the DADX runtime
  - Web Service Wizard : deploys DADX documents to the runtime
  - WORF Runtime : executes Web Services defined by DADX documents, generates a test page, WSDL, and converts DTD to XML Schema
  - RDB to XML Mapper : defines the mapping from relational to XML data, and generates DAD documents that can be used in a DADX

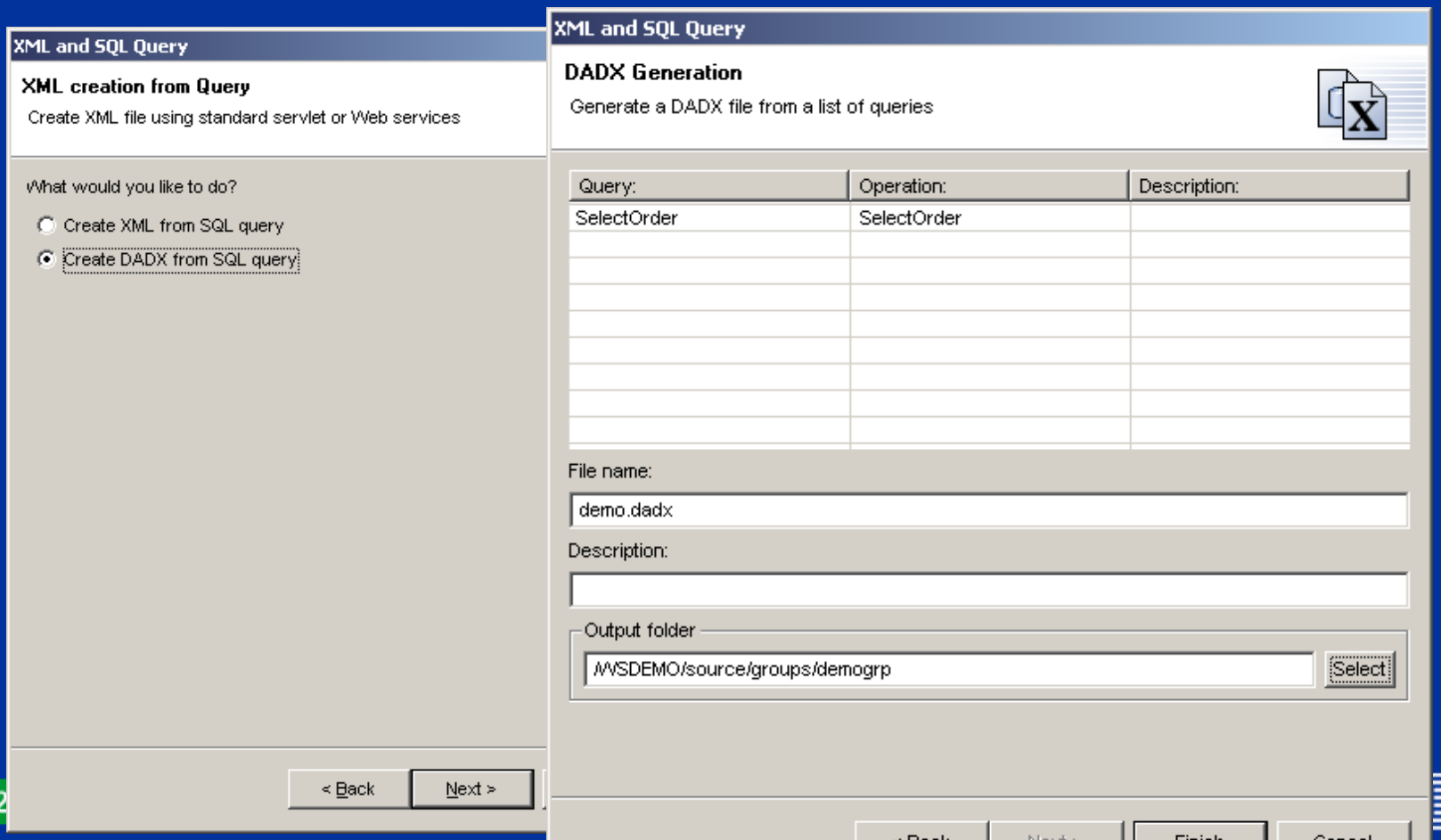
Step 1 - Create a web services group

- Pre-req : Check you are using JDBC 2.0
  - check sqllib\java12\inuse
  - add db2java.zip to server configuration in WSAD
- Step 1 : Create group
  - New - Other - Web services - Web service DADX configuration
  - Add group under your chosen project
  - Click group properties and select required database

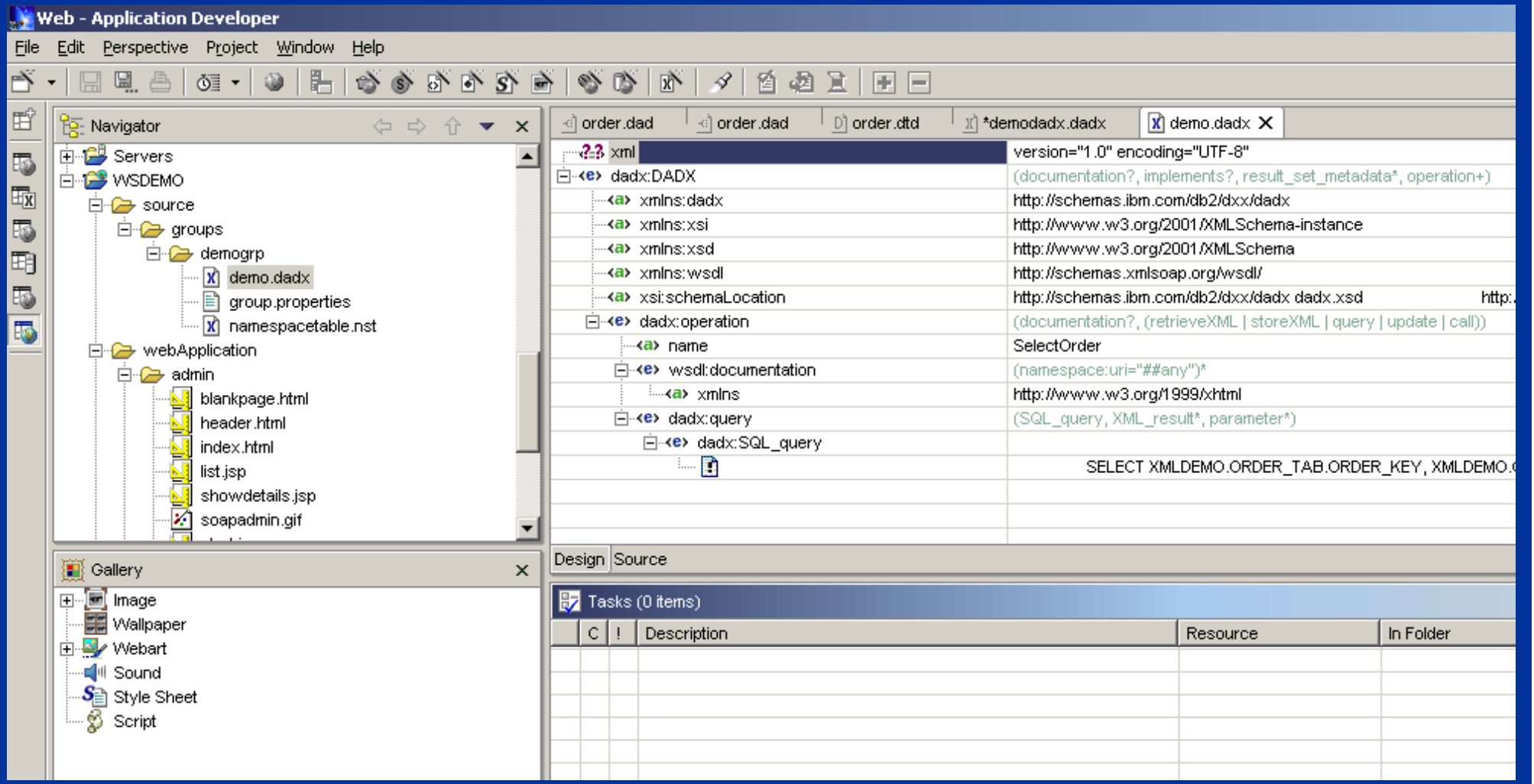


Step 2 - Create DADX

- Click on XML Perspective : New - XML from SQL query - DADX from SQL query
- Next 2 panels ask you to select required statements and/or DADs
- Choose an output folder of /project/source/groups/group name

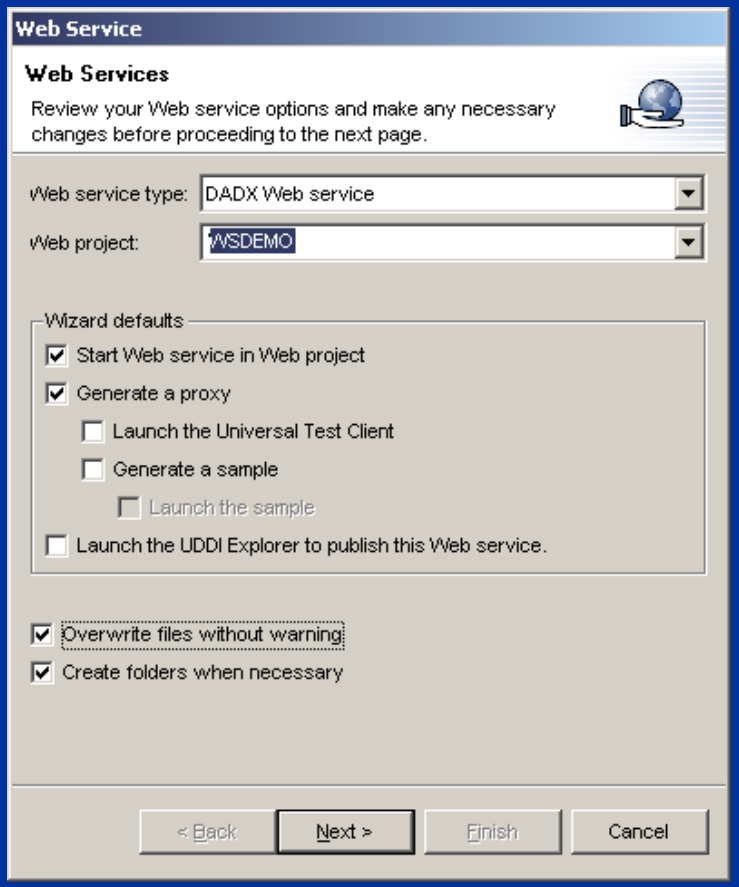


Generated DADX



Step 3 : Create the web service

- New - Web service
- Select required DADX
- Generates a SOAP deployment descriptor, adds it to the Websphere configuration file, runs the server and generates the WSDL files for the service
- You can then test it using test page
  - http://servername:port/project/group/dadx name/TEST
  - Right click on DADX - Run on server
- My First Java and DB2 Web service demo scripts can be found at http://w3.torolab.ibm.com/~ryman/



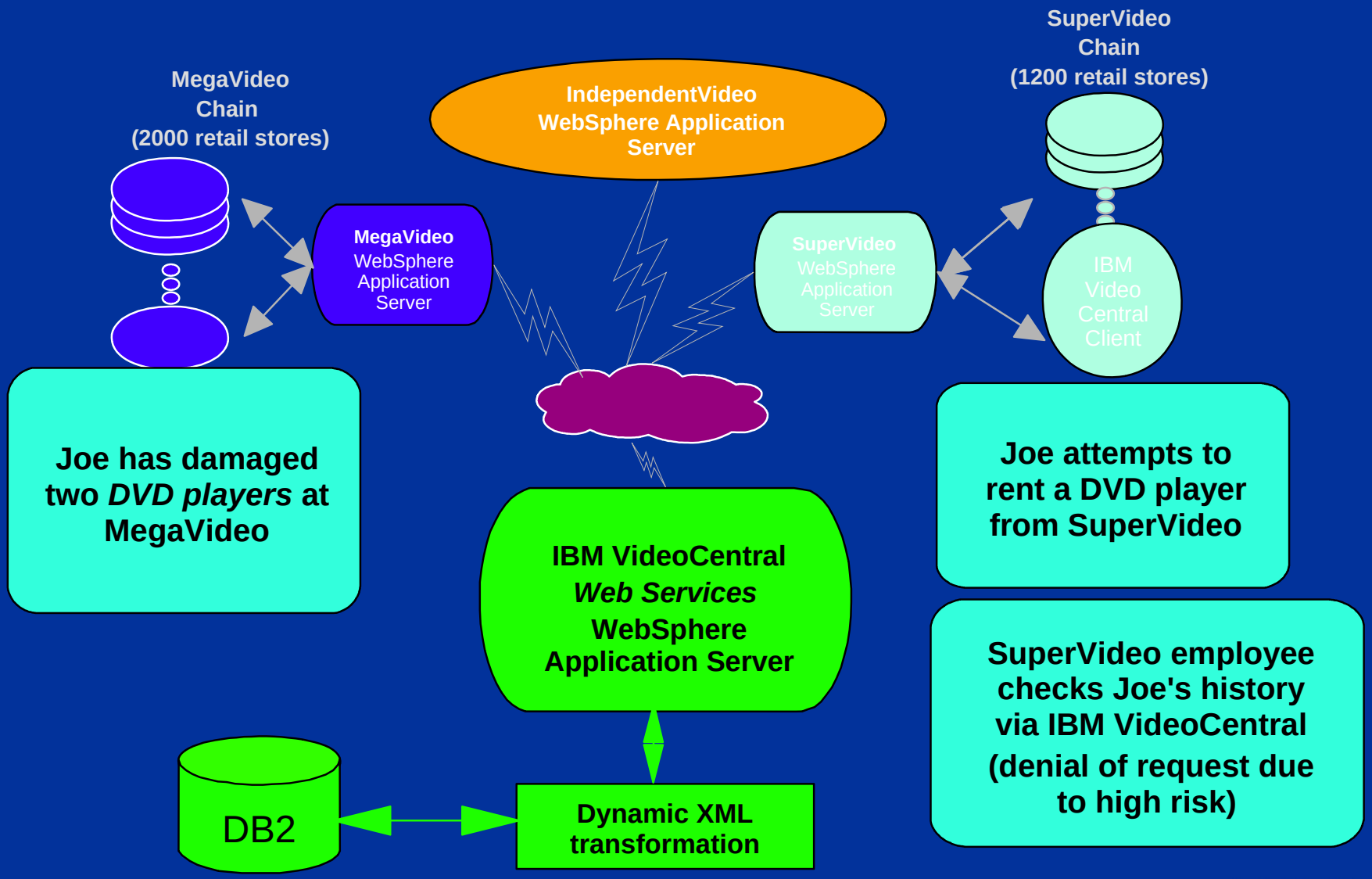
Some points to note

- Web services groups
  - Share common properties eg database connection parameters
  - Group.properties file under webApplication/WEB-INF/classes/groups
- DADX files are like JAVA classes
  - Both contain the implementation of web services
  - Therefore your DADX libraries can be loaded by the Java class loader and they can execute directly from the WAR files
- Use Netscape Communicator rather than Internet Explorer when testing as it gives more diagnostics
  - Netscape Communicator and IE Version 5 or later support XML

Video Central Demo

- Hypothetical web service provider for video rental applications
- Business services
  - validates customer credentials (checks history across all the video rental companies
- Customer services
  - rental history
  - wish-list registry
  - recommended video list
- XML application tutorial
  - http://www.ibm.com/software/data/developer/samples/video/

IBM Video Central - DB2 Web services in Action



DB2 Data Management Software

Video Central Design

The IBM Video Central application is a three tier application:

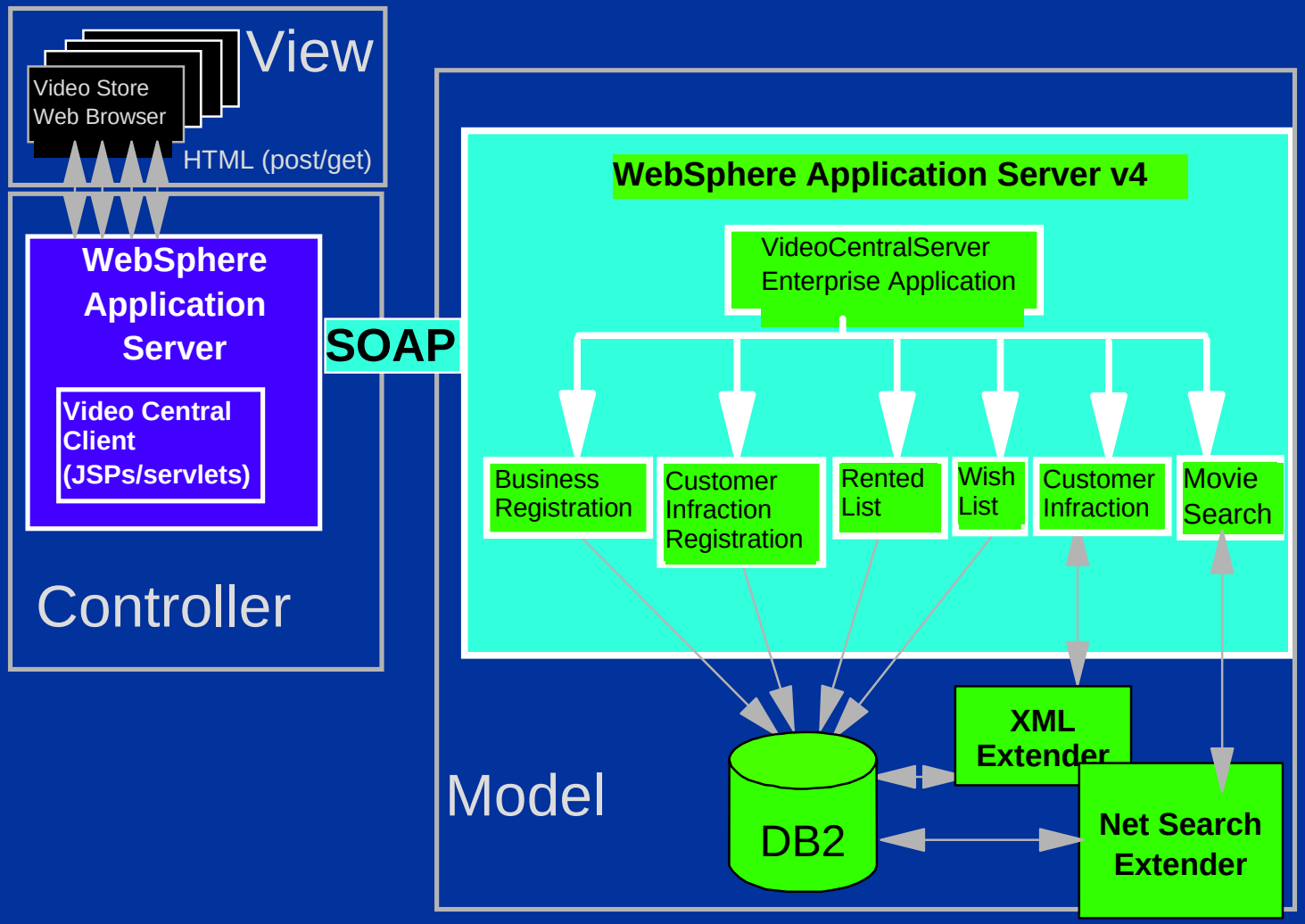
1. Web Interface Layer
2. Business Logic Layer
3. Data Access Layer

The **Web Interface Layer** for IBM Video Central utilizes JavaServer Pages (JSPs) for presentation and servlets for Web service invocation.

The **Business Logic Layer** is implemented using JavaBeans and the interface is described using WSDL. The WSDL interface can be published to a public or private UDDI (Universal Description, Discovery, and Integration) registry.

The **Data Access Layer** is implemented using IBM's Data Access Bean technology (based on JDBC). Standard features of WebSphere Application Server including database connection pooling are used to provide efficient resource usage.

Video Central Architecture



Videocentral demo

- Accessible via the following website
  - http://edata1.dfw.ibm.com/EVCClient_App/evc/requestJSP/EVCClientRequest.jsp
- Some sample tasks that can be carried out:
  - Register your own customer, add and infraction and query it
  - Alternatively, query an existing infraction using the name "Grant Hutchison" and phone number 999 999 9999 and a date range that includes Feb. 10, 2001

Futures and directions

Including XQuery and Web services

DB2 and Web Services Directions

- DB2 as a Web Service (Provider)
  - Available now through Websphere Studio and DADX:
    - The ability for DB2 data, functions and stored procedures to beinvoked as Web Services without writing any custom servlets
  - Directions: Support for more varieties of SQL
  - Directions: The ability for Web Services clients to invoke XQuery requests very easily
- DB2 as a Web Service Client
  - Directions: the ability for DB2 applications to invoke Web services very easily

XQuery Web services

- A future DB2 release will ship with XML query over relational data implementation
  - exact details and dates are not yet finalised
- Approach
  - user defines XML views over relational data
  - views are defined by XML query statements with extensions to access relational tables
  - user can query these XML views with XML query statements
- There will be extensions to the DADX file format to allow web service implementations by XML queries
- Benefits:
  - standardized query language
  - more relational to XML mapping capabilties
  - XML query optimized before execution of selects
  - Disadvantage : No update capability

XML View Example 1

Create view statement

```
create view customer_orders as (
  for [$KEY=ORDER_KEY
       $NAME=CUSTOMER_NAME
       $EMAIL=CUSTOMER_EMAIL
       $PHONE=CUSTOMER_PHONE ]
       in WOLLSCH.ORDER_TAB
  return
    <Order key={$KEY} >
       <Customer>
          <Name>{$NAME}</Name>
          <Email>{$EMAIL}</Email>
          <Phone>{$PHONE}</Phone>
       </Customer>
  </Order>
  )
```

Result of view("customer_orders")

```
<RESULT>
   <Order key="1">
      <Customer>
         <Name>American Motors</Name>
         <Email>parts@am.com</Email>
         <Phone>800-AM-PARTS</Phone>
      </Customer>
   </Order>
  [...]
</RESULT>
```

XML View Example 2

Create view

```
create view full_order as (
for $i in view("customer_orders")
return <Order key={$i/@key} >
        {$i/Customer}
        { for $pi in view("part_orders")
                where $i/@key=$pi/@orderkey
        return $pi }
      { for $si in view("part_shipments")
        where $i/@key=$si/@order
        return $si }
  </Order>  )
```

Result of view ("full_order")

```
<RESULT>
   <Order key="1">
      <Customer>
         <Name>American Motors</Name>
         <Email>parts@am.com</Email>
         <Phone>800-AM-PARTS</Phone>
      </Customer>
      <Part color="red    " orderkey="1">
         <key>156</key>
         <Quantity>17</Quantity>
         <ExtendedPrice>17954.55</ExtendedPrice>
         <Tax>0.02</Tax>
      </Part>
```

XML Queries

get order number 1:

```
for $i in view ("full_order")
  where $i/@key=1
  return $i
view("full_order")[@key=1]
```

get the total sum of part prices shipped for order 1

```
for $i in view ("full_order")
  where $i/@key=1
  return sum ($i/Part/ExtendedPrice)
sum(view("full_order")[@key=1]/Part/ExtendedPrice)
```

get the parts in order one sorted by price

```
for $i in view ("full_order")
where $i/@key=1
return $i/Part sortby (ExtendedPrice descending)
```

DADX with XML Query

Example

```
<operation name="myXMLQuery">
  <query>
    <XML_query>
      view("full_order")[@key=1]
    </XML_query>
  </query>
</operation>
```

WSDL generation uses:

- anyType as return type of operation
- reference to existing external XMLSchema describing the query result
- definition of XML schema types in the DADX file

Binding parameters to XMLQuery as in SQL case

DB2, XML and MQSeries

What does it do?

- DB2 7.2 (June 2001) provide basic MQSeries Integration
  - SQL Functions to support datagrams, request/reply, pub/sub
  - MQAssist Wizard to facilitate construction of table functions and views
  - Support for XML messaging through the DB2 XML Extender

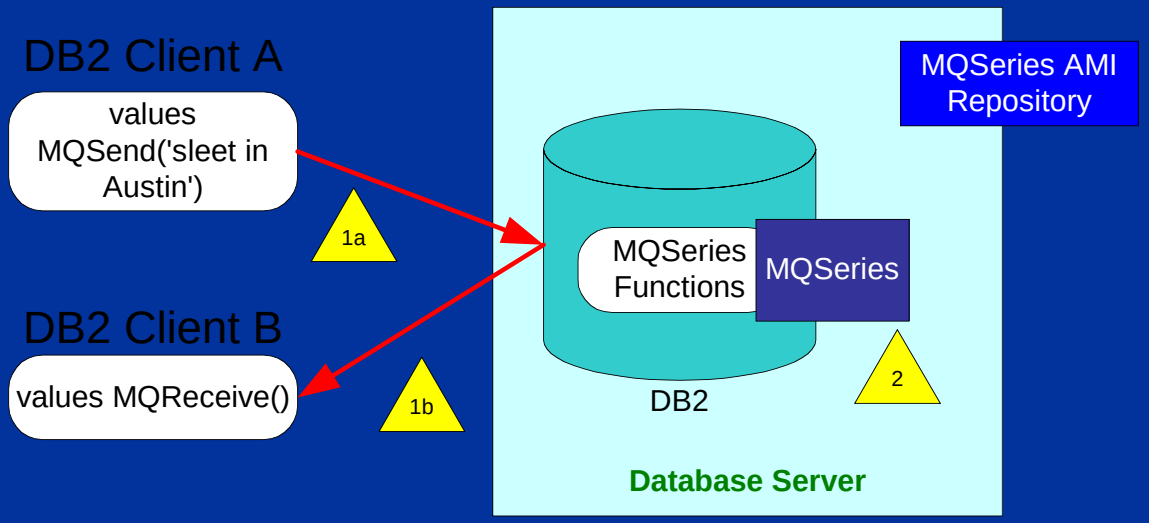
- These features will be available for application development and will also be incorporated into the Data Warehouse Center
  - Allows MQSeries as a source of information for a DB2 Data Warehouse
  - Supports XML structured messages and simple structured messages

Simplified Application Interfaces

- Database programmers can invoke messaging functions in a familiar environment.
  - MQ integration functions can be used as part of SQL statements
  - Incorporate messaging operations into triggers
- Available from any language supporting SQL
  - C, Java, C++, SQL, ...
- Optionally specificy of location and policy
  - Location is the logical name describing the destination or source of a message
  - Policy is the logical name describing message handling
  - Use AMI Administration tool to configure
- Support for multiple message contents models
  - Simple structures (positional or delimited)
  - Unstructured strings
  - Self-describing XML
  - Typed messages defined within the MQSeries Message Repository (future)

Basic Local Messaging

- MQSeries server executes on the same machine as the DB2 Server
- DB2 clients may be local or remote.
  - standalone applications, WebSphere servlets or EJBs
  - local stored procedures



Basic Local Messaging

Basic Messaging

- MQSend
  - MQSend(msg)
  - MQSend(service, msg)
  - MQSend(service, policy, msg)
  - MQSend(service,policy, msg, correlid)
- Examples
  - values MQSend('a test message')
  - values MQSend('DB2.DEFAULT.SERVICE','DB2.DEFAULT.POLICY', 'a message')
  - values MQSend('DB2.DEFAULT.SERVICE','DB2.DEFAULT.POLICY', 'a message','correlid1')
  - select MQSend(lastname) from employee
  - select MQSend(lastname || ' ' || firstnme) from employee
  - select MQSend(lastname) from employee where deptno=20
  - select MQSend(e.lastname || ' ' || d.manager) from employee e, dept d where e.deptno = d.deptno

Basic Messaging

- getting messages - scalar functions
  - MQRead - non-destructive read
  - MQReceive - destructive read
  - msg = MQRead()
  - msg = MQReceive()
  - msg = MQRead(receive-service)
  - msg = MQRead(receive-service, service-policy)
  - msg = MQReceive(receive-service, service-policy, correl-id)

- Examples
  - values MQRead()
  - values MQReceive('myService','myPolicy', 'correlid1')
  - insert into MESSAGE_ARCHIVE(time, msg) values (current time, MQRead())

Basic Messaging

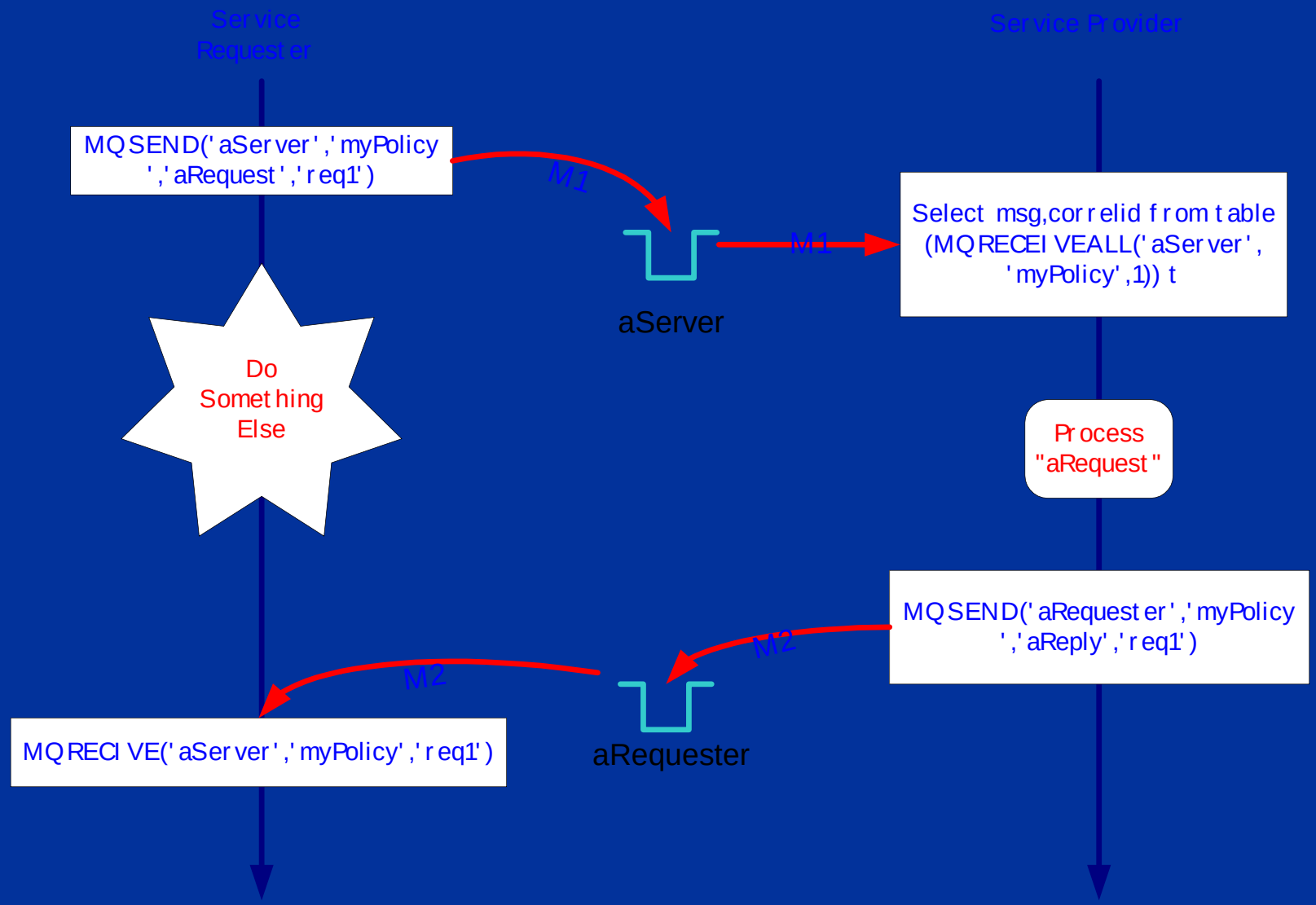
- getting messages - table functions
  - MQReadAll - non-destructive read
  - MQReceiveAll - destructive read
  - MQReadAll()
  - MQReadAll(receive-service)
  - MQReadAll(receive-service, policy)
  - MQReceiveAll(num-rows)
  - MQReceiveAll(receive-service, policy, correlid, num-rows)

- Returns
  - msg - Varchar(4000)
  - Correlid - Varchar(24)
  - Topic - Varchar(40)
  - QNAME - Varchar(48)
  - MSGID - Char(24)
  - MSGFORMAT - Varchar(8)

Basic Messaging

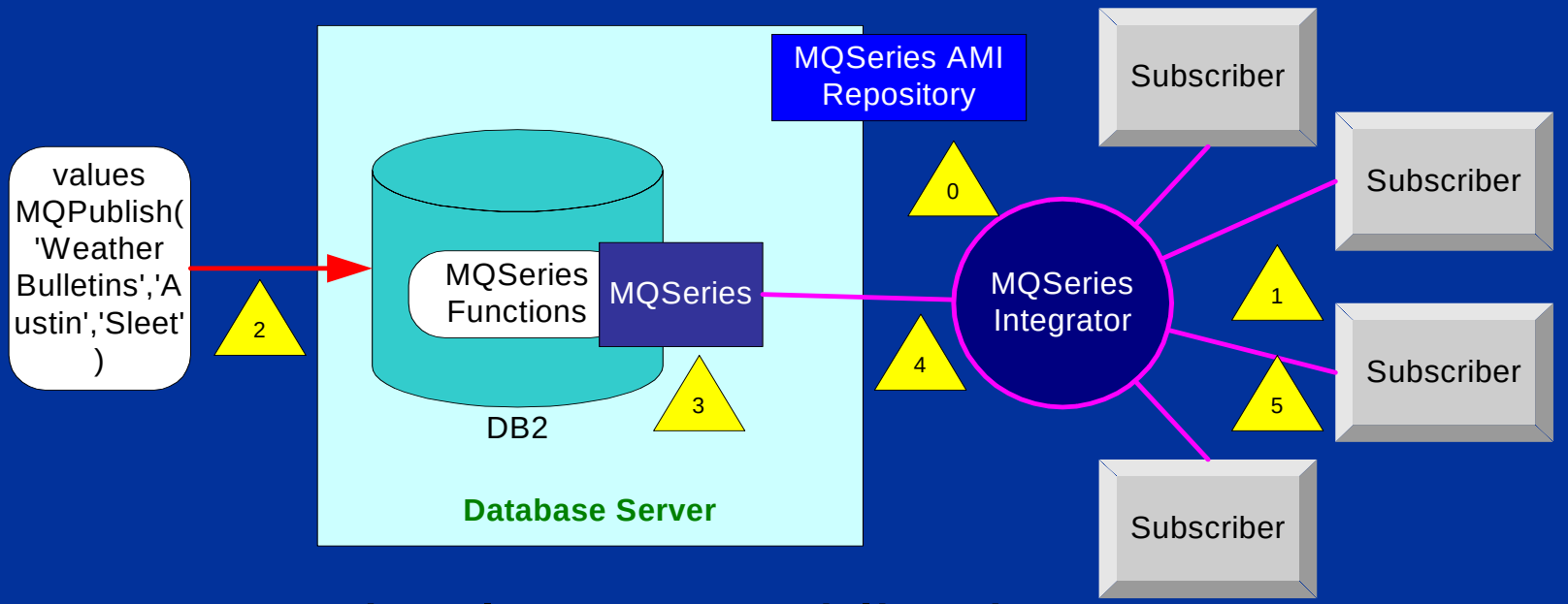
- Table Function Examples
  - select * from table (MQReadAll()) t
  - select msg from table (MQReadAll()) t
  - select varchar(msg,20) from table (MQReceiveAll(10)) t
  - select count(*) from table (MQReadAll()) t
  - select count(*) from table (MQReceiveAll()) t
  - select * from table (MQReadAll(10)) t where t.CorrelID='AB'
  - select * from table (MQReceiveAll('myService','myPolicy','AB',10)) t
  - select t.msg, e.age from employee e, table(MQReadAll()) t where t.msg = e.lastname
  - insert into msg_history values (select * from table (MQReadAll()) t)
  - create view M as (select msg from table (MQReadAll()) t)

Request/Reply Messaging



Publish/Subscribe

- Supports both MQSeries publish/subscribe and MQSI V2
- Publication may be on-demand or automated through triggers
- Subscribers can register one or more topics



Simple Data Publication

Publish / Subscribe

- MQPublish
  - MQPublish (msg)
  - MQPublish (publisher-service, msg)
  - MQPublish (msg, topic)
  - MQPublish (publisher-service, service-policy, msg, topic, correlid)

- Examples
  - values MQPublish('my favorite stock quote')
  - values MQPublish('200','IBM:MyPortfolio')
  - CREATE TRIGGER new_employee AFTER INSERT ON employee REFERENCING NEW AS n FOR EACH ROW MODE DB2SQL VALUES MQPublish('HR_INFO_PUB', current date || ' ' || lastname || ' ' || DEPARTMENT,'NEW_EMP')

Publish/Subscribe

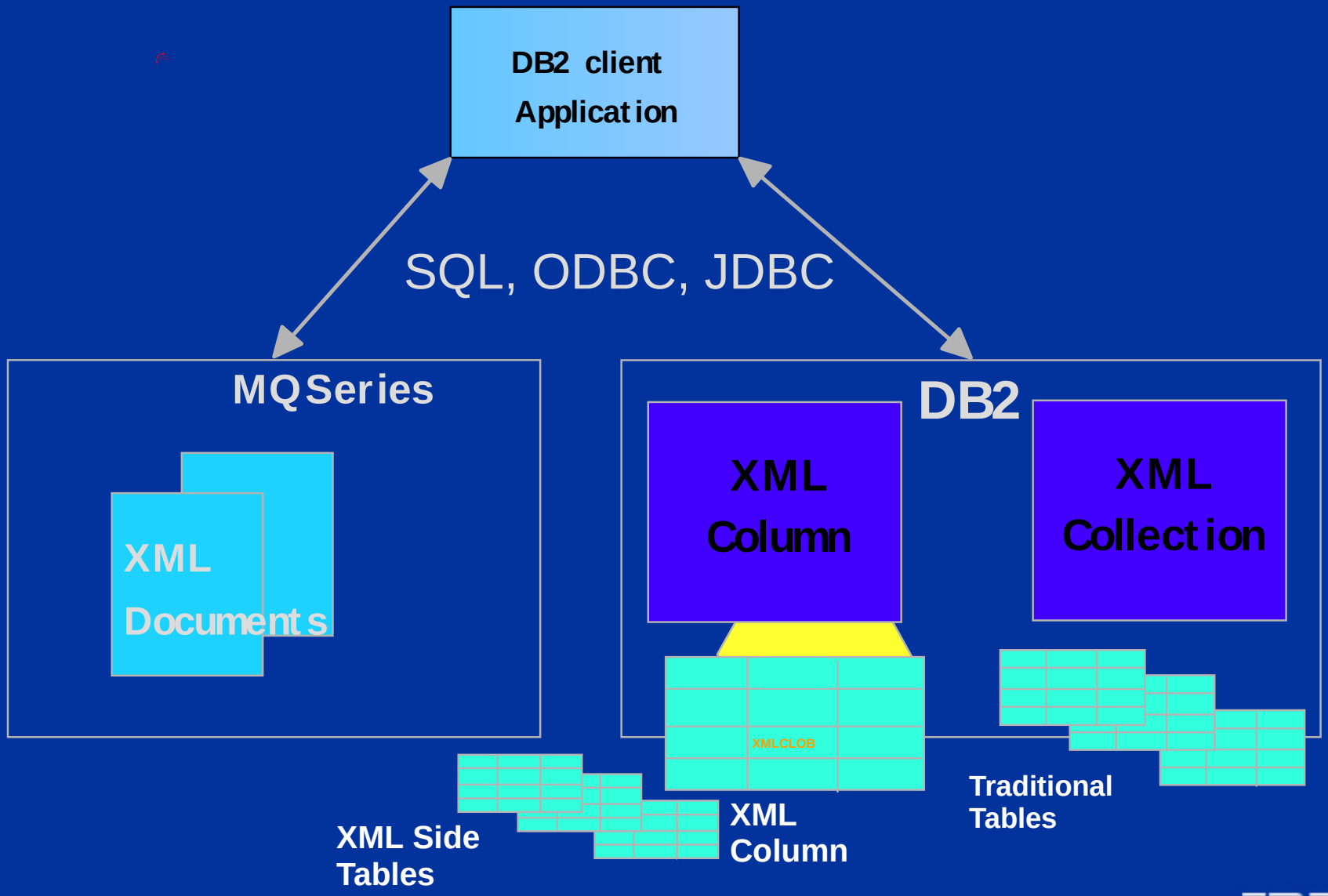
- MQSubsubscribe
  - MQSubscribe(topic)
  - MQSubscribe(subscriber-service, policy, topic)
- MQUnSubscribe
  - MQUnSubscribe(topic)
  - MQUnSubscribe(subcriber-service, policy, topic)
- Examples
  - values MQSubscribe('myPortfolio')
  - values MQSubscribe('myPortfolio:IBM')
  - values MQSubscribe('emp_subscription','NEW_EMP')
    select * from table (MQReceiveAll('emp_subscription')) t
  - values MQUnsubscribe('myPortfolio')

Using the DB2 XML Extender and MQSeries through SQL and XML

- A series of SQL User Defined Functions and Stored Procedures are supplied in DB2 7.2 enabling a DB2 client application to do the following from a single request:
  - To read from MQSeries queues into DB2 XML columns
  - To send and publish messages to MQseries queues from DB2 XML columns
  - To send XML messages composed from relational data to MQSeries queues
  - To decompose (shred) XML messages held in MQSeies queues into relational data

- It is possible to integrate MQSeries operations with DB2 table operations in a single SQL request such as a SELECT

DB2 and MQSeries: Integration through XML



DB2 Data Management Software

DB2 XML Column for MQSeries

- MQReadXML (or MQReadXMLCLOB)
  - Places an XML message at the head of an MQSeries queue, without removing it from the queue, in an XMLVarchar (or an XMLCLOB) row
- MQReadXMLAll (or MQReadAllXMLCLOB)
  - Returns a DB2 table containing XML message data without removing messages from the queue. The messages are placed in an XMLVarchar column (or an XMLCLOB column)
- MQReceiveXML (or MQReceiveXMLCLOB)
  - Removes an XML message at the head of an MQseries queue, and places it in an XMLVarchar (or an XMLCLOB) row
- MQReceiveAllXML (or MQReadAllXMLCLOB)
  - Removes XML messages from an MQseries queue, and places them in an XMLVarchar column (or an XMLCLOB column)

DB2 XML Column for MQSeries

- MQSendXML (and MQSendXMLCLOB)
  - Send an XML message to a queue from a DB2 XML column
- MQSendXMLFILE (and MQSendXMLFILECLOB)
  - Send an XML message to an MQSeries queue from an XML file
- MQPublishXML (and MQPublishXMLCLOB)
  - Publish XML messages to an MQSeries queue from a DB2 XML column
- Example: SELECT MQSend(orderdoc) from order_tab
  - This statement will cause all orders in the order table to be sent to an MQSeries queue

DB2 XML Collection for MQSeries

- dxxmqGen and dxxmqRetrieve
  - Compose XML documents from relational data and place documents in MQSeries queues.
- dxxmqShred and dxxmqInsert
  - Decompose an XML document held in message queues and store in new or existing relational database tables.
- dxxmqInsertAll and dxxmqShredAll
  - Decompose XML documents held in message queues and store in new or existing relational database tables.
- Parameters include MQSeries Service Names and Policy Names

In Closing...

- Open Standards (like XML) are key enablers of e-business
- Open Standards and Open Source are not the same thing but they complement each other
- We are moving towards a Web of services
  - SOAP provides service-oriented architecture for server-to-server and server-to-device communication
  - UDDI provides directory for services
- IBM is here to work with you for e-business solutions

And finally don't forget ...

- Please complete and return the feedback form
- The second workshop 'Using the DB2 XML Extender' will run next Tuesday and Wednesday and will show the end to end process including tooling

Information sources

- DB2 Web Services
  - www.ibm.com/software/data/webservices
  - www.ibm.com/developerworks/webservices
- **Download DB2 Web Services (also available through WSAD)**
  - **http://www.ibm.com/software/data/db2/extenders/xmlext/docs/v72wrk/WORF.html**
- **DB2 and XML Web Services Papers**
  - Running DB2® Web Services on WebSphere® Application Server AE 4.0 by Reto Preisig
    - http://www7b.boulder.ibm.com/dmdd/library/techarticle/preisig/0108preisig.html
  - http://www.ibm.com/software/data/webservices
- IBM Developerworks for Web Services
  - http://www.ibm.com/developerworks/webservices/
- WSAD: http://www.ibm.com/software/ad/studioappdev/

Websphere Information Sources

- WAS Supported databases and JDBC Resources can be found at:
  - http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html
- IBM WebSphere
  - http://www.ibm.com/software/webservers/
- IBM WebSphere Certification for Administrators and Application Developers
  - http://www.ibm.com/Education/certify
- WebSphere Magazine
  - http://www.websphereadvisor.com
- WebSphere Developer Domain
  - http://www7b.boulder.ibm.com/wsdd/
- WebSphere Users' Group
  - http://www.websphere.org
  - http://www.websphere-users.ca/ - Canada

XML Resources: IBM Papers

- DB2 MQSeries and XML Papers
  - http://www7b.boulder.ibm.com/dmdd/library/techarticle/wolfson/0108wolfson.html
  - http://www7b.boulder.ibm.com/dmdd/library/techarticle/wolfson/0201wolfson.html
  - http://www.ibm.com/software/data/db2/extenders/xmlext/docs/v72wrk/dxxmq.htm
  - http://www.ibm.com/software/data/db2/extenders/xmlext/docs/v72wrk/dxxrnfp4.htm#Header_10