Session B06

Web Services (SOAP) for DB2

S. Malaika, db2xml@us.ibm.com

IBM Data Management Technical Conference, CA **September 2002**

Agenda

- Web Services Overview
- DB2 as a Web Service Provider
  - DADX
- What's Next
  - DB2 as a Web Service Requestor
- Resources
- Summary

Web Services Overview

- Web Services: Programmatic interfaces for providing and invoking URL addressable software across the Web (using http)
- A Web Service: URL addressable software that can be invoked over the Web without the requestor knowing the Web Service implementation
  - A self-contained, self describing modular service that can be
    - provided and published on the Web
    - discovered and invoked across the Web by other applications and Web Services
  - The service can perform various functions from simple requests to complex business processes

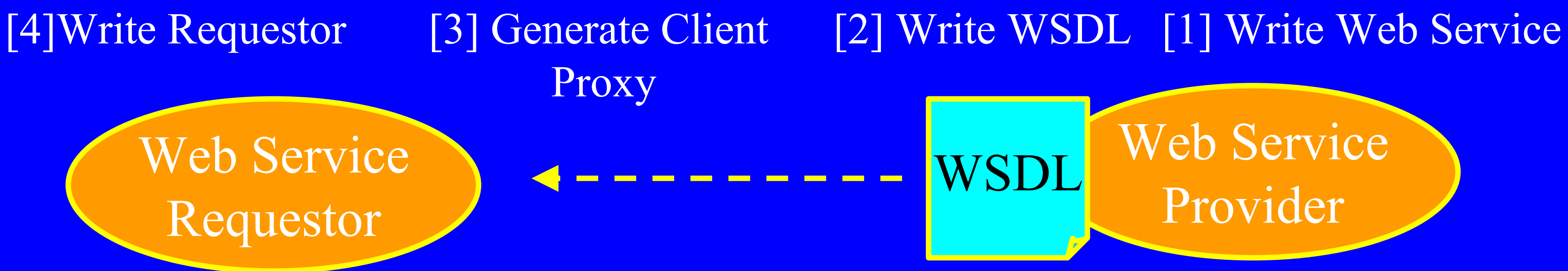
Why Web Services

- Enable customized or general purpose programs to be Web requestors
  - Prior to Web Services, the main Web requestors were humans using general purpose browsers
- Provide flexibility and platform independence once deployed
  - Based on XML standard data formats and exchange mechanisms
- Web Services Invocations Examples
  - Get the current temperature in Sorrento
  - Get the current value of IBM stock
  - Get my last 5 savings account transactions
  - Get the hotel receipts for these expense claims
  - Buy two movie tickets for tonight's 7pm show of Possession at the nearest cinema
  - Register all my employees for the XML class in San Francisco next week

Web Services Standards Activities

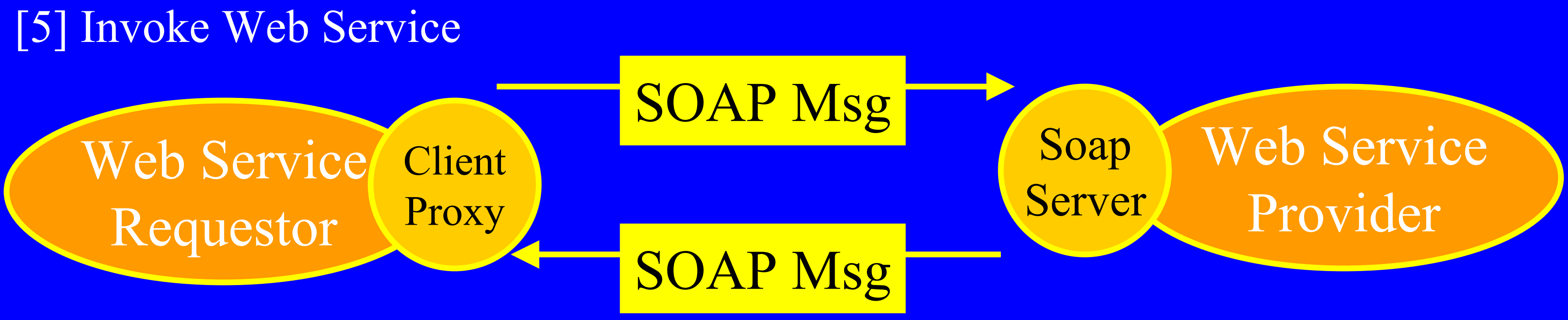
- W3C : http://www.w3.org/
  - Three Web Services groups
    - Web Services Architecture: Provide usage scenarios
    - SOAP (Simple Object Access Protocol):
      - Defines the precise content of messages transmitted as XML when Web services are invoked across http
    - WSDL (Web Services Description Language) Working Group
      - Defines a Web Service interface, using an XML notation
  - A coordination group
- WS-I: WS Interoperability http://www.ws-i.org/
- Oasis: http://www.oasis-open.org/
  - UDDI : Universal Description, Discovery, and Integration

SOAP and WSDL: Preparing and Invoking



Development Time

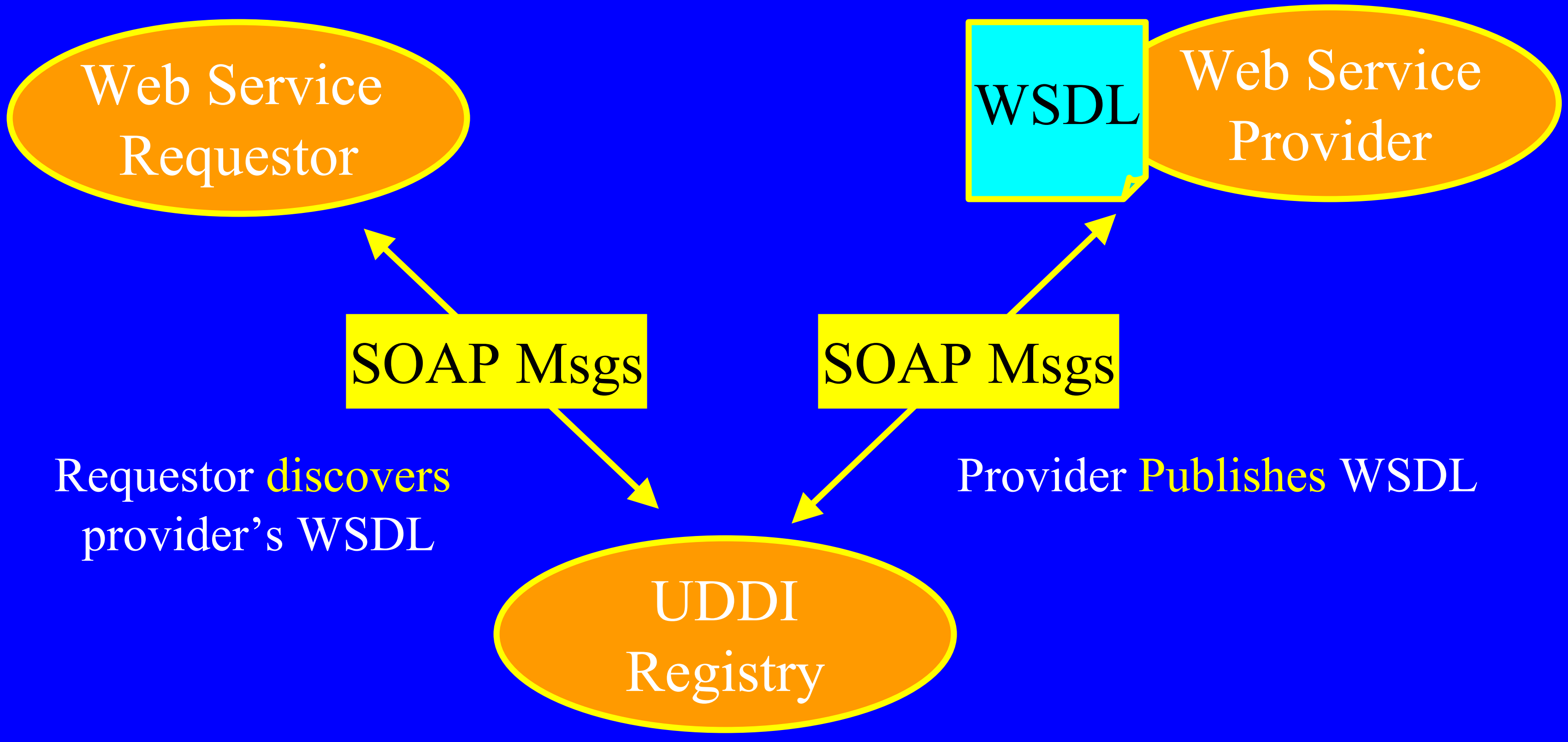
Preparing a Web Service and a Web Service Requestor



Run Time

Requestor invokes the Provider through the client proxy

SOAP, WSDL & UDDI: Publishing & Discovering



UDDI: Similar to telephone yellow pages

Publication and Discovery may take place prior to a SOAP request being issued

WSDL

The following components can appear in a piece of WSDL:

Apply to a collection of similar Web services in various locations:

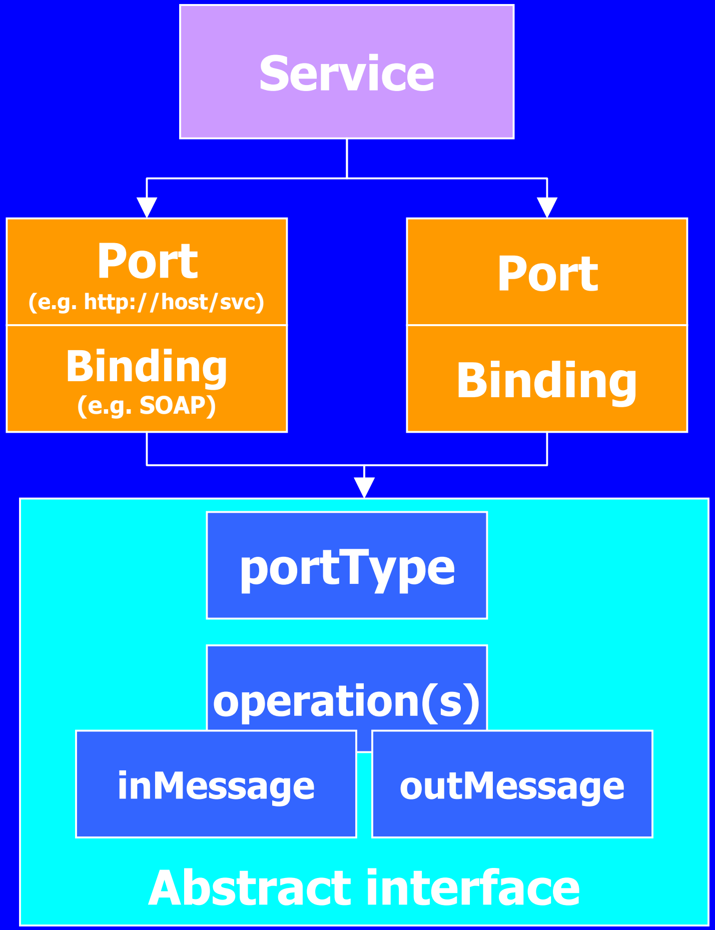
- type: defines data types used in the WSDL
- message: defines types of messages to be generated for the operations
- portType: defines the operations offered by the service
- serviceType: defines a grouping of port types

Apply to a particular Web Service at a particular location:

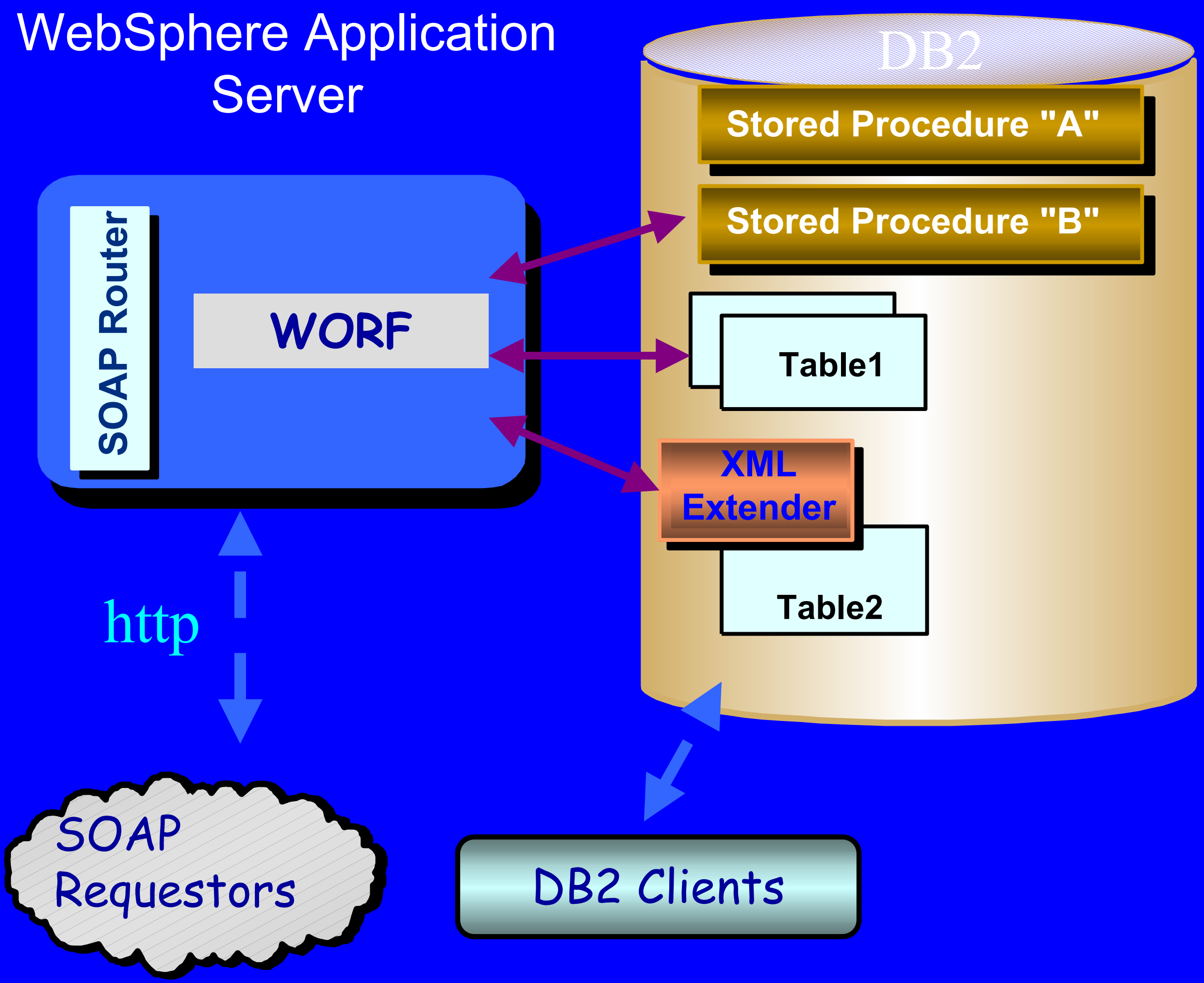
- binding: defines the protocol used to send the messages.
- service: defines a collection of ports (bound port types) and where they can be located (an end-point)

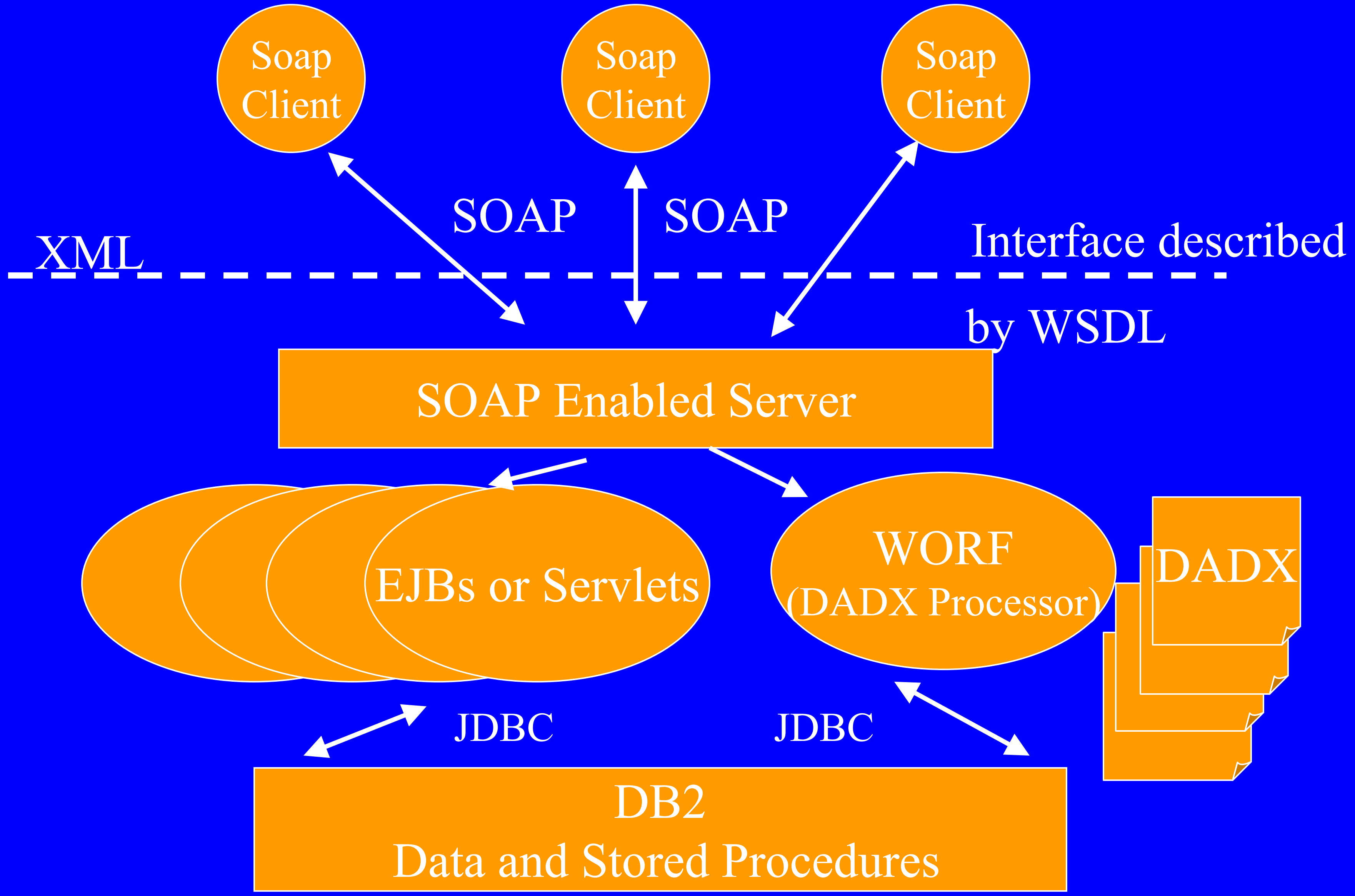
WSDL Model

- A WSDL description has three basic parts:
- **Abstract interface:**
  - Abstract definition of a service
    - A set of operations within a porttype
- **Protocol bindings**
  - How to access the service
  - Multiple per portType, e.g.,:
    - SOAP, JMS, direct call.
- **Deployed service access ports**
  - Where to access the service.



DB2 as a Web Service Provider





Two Options: 1. Custom EJB components / servlets or
2. DADX files

DADX Processor (WORF)

- Available through WebSphere Studio or as Web download from the DB2 XML Extenders Website:

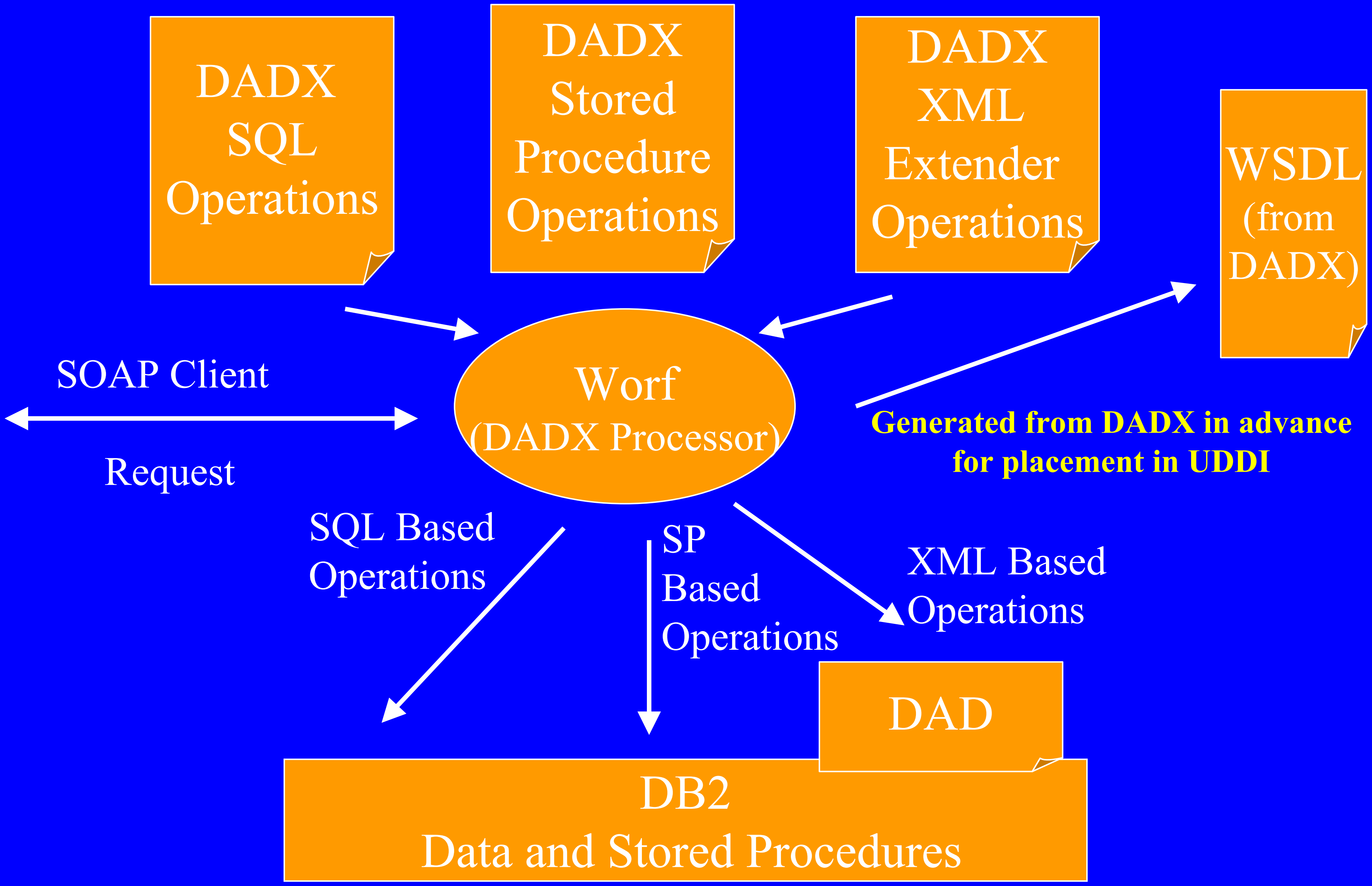
http://www.ibm.com/software/data/db2/extenders/xmlext/docs/v72wrk/WORF.html

Part of DB2 V8

- Overview paper available from:

http://www.ibm.com/software/data/pubs/papers/db2webservices/db2webservices.pdf

- The DADX processor can be used with:
  - WebSphere
  - Apache Tomcat
- 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


DADX SQL Operations
DADX Stored Procedure Operations
DADX XML Extender Operations
WSDL (from DADX)
SOAP Client
Request
Worf (DADX Processor)
Generated from DADX in advance for placement in UDDI
SQL Based Operations
SP Based Operations
XML Based Operations
DAD
DB2 Data and Stored Procedures

Three DADX Operation Types

Sample database tables that generated the XML document

order_tab

order_key	customer_name	customer_phone	customer_email
1	American Motors	Parts@am.com	800-am-parts

part_tab

part_key	color	quantity	price	tax	order_key
156	red	17	17954.55	2.0	1
68	black	36	34850.16	6.0	1
128	red	28	38000.00	7.0	1

ship_tab

date	mode	comment	part_key	order_key
2002-03-13	truck	Comment 1	156	1
2002-01-16	fedex	Comment 2	156	1
2002-08-19	boat	Comment 3	68	1
2002-08-19	air	Comment 4	68	1
2002-12-30	truck	Comment 5	128	1

Sample SQL Composition DAD

GetstartxCollection.dad

```
<?xml version="1.0"?>

<!DOCTYPE DAD SYSTEM "c:\dxx\dtd\dad.dtd">

<DAD>

<validation>NO</validation>

<Xcollection>
```

1. Scoping the Document

```
<SQL_stmt>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 o.order_key = 1 and p.price > 20000 and p.order_key = o.order_key and
s.part_key = p.part_key ORDER BY order_key, part_key, ship_id</SQL_stmt>

<prolog>?xml version="1.0"?</prolog>
```

2. Shaping the Document

```
<doctype>!DOCTYPE Order SYSTEM "c:\dxx\samples\dtd\getstart.dtd"</doctype

<root_node><element_node name="Order">  <attribute_node name="key">

    <column name="order_key"/>  </attribute_node>
```

Sample SQL Composition DAD (cont)

```
<element_node name="Customer">  <element_node name="Name">
                <text_node><column  name="customer_name"/></text_node> </element_node>
               <element_node name="Email">   <text_node><column name="customer_email"/></text_node>
       </element_node>  </element_node>
        <element_node name="Part">  <attribute_node name="color"> <column name="color"/>  </attribute_node>
        <element_node name="key"> <text_node><column name="part_key"/></text_node>    </element_node>
         <element_node name="Quantity"> <text_node><column name="quantity"/></text_node> </element_node>
         <element_node name="ExtendedPrice"> <text_node><column name="price"/></text_node> </element_node>
        <element_node name="Tax">      <text_node><column name="tax"/></text_node> </element_node>
       <element_node name="Shipment" multi_occurrence="YES">
                  <element_node name="ShipDate"> <text_node><column name="date"/></text_node> </element_node>
                <element_node name="ShipMode">  <text_node><column name="mode"/></text_node> </element_node>
</element_node>  </element_node></element_node>
</root_node></Xcollection>
</DAD>
```

Generated XML Document

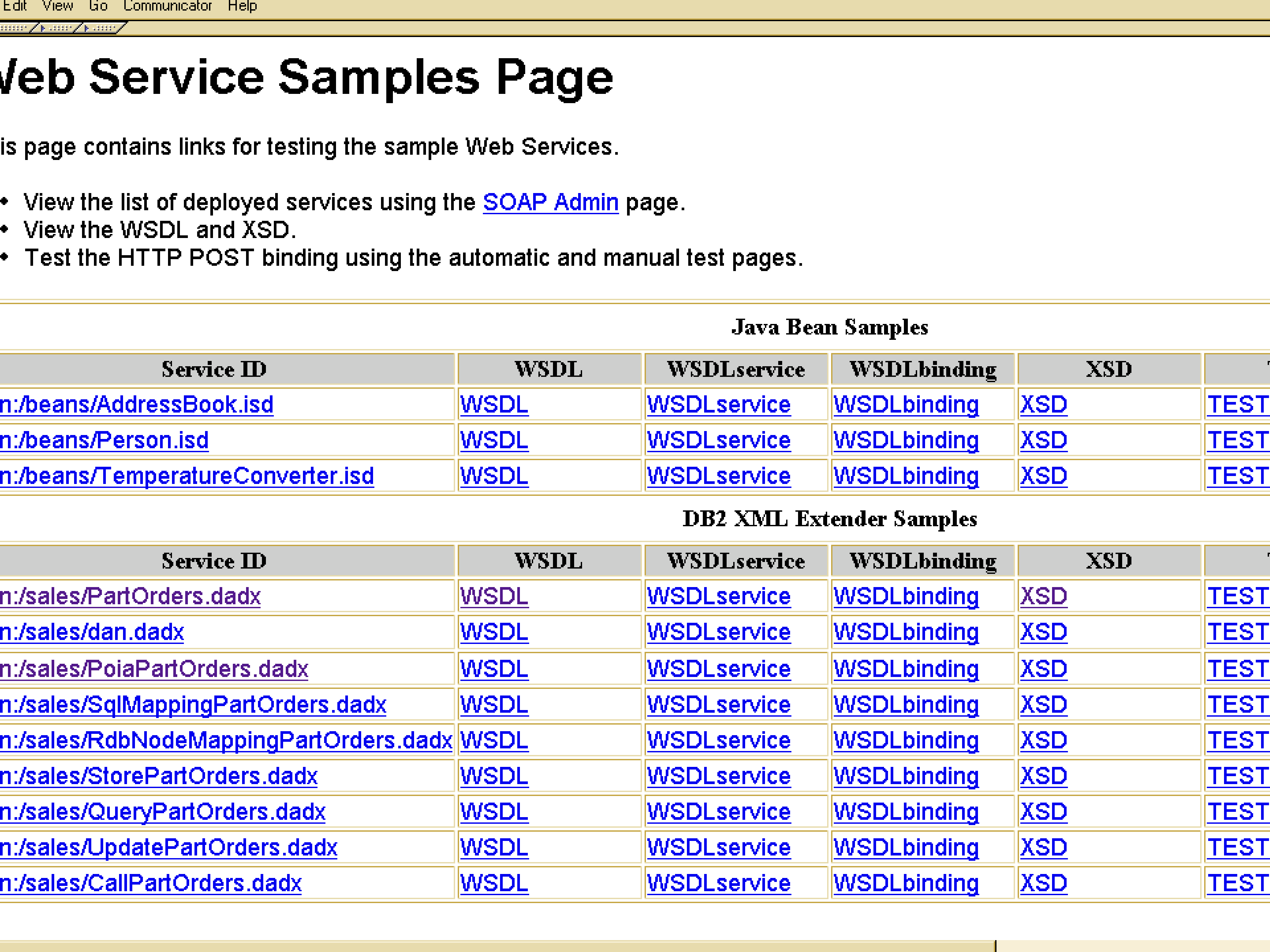
```
<?xml version="1.0"?>
<!DOCTYPE Order SYSTEM "c:\dxx\samples\dtd\getstart.dtd">
<Order key="1">  <Customer>
                    <Name>American Motors</Name> <Email>parts@am.com</Email>
                  </Customer>
 <Part color="black ">
         <key>68</key><Quantity>36</Quantity>
         <ExtendedPrice>34850.16</ExtendedPrice> <Tax>6.00</Tax>
        <Shipment>
              <ShipDate>2002-08-19</ShipDate> <ShipMode>boat </ShipMode>
        </Shipment>
```

Generated document (cont)

```
         <Shipment>
                        <ShipDate>2002-08-19</ShipDate><ShipMode>boat</ShipMode>
               </Shipment>
               <Shipment>
                        <ShipDate>2002-08-19</ShipDate><ShipMode>air</ShipMode>
               </Shipment>
  </Part>

<Part color="red">
        <key>128</key>
        <Quantity>28</Quantity>
       <ExtendedPrice>38000.00</ExtendedPrice>
         <Tax>7.0</Tax>
         <Shipment>
               <ShipDate>2002-12-30</ShipDate><ShipMode>truck</ShipMode>
        </Shipment>
            <Shipment>
                <ShipDate>2002-12-30</ShipDate><ShipMode>truck</ShipMode>
               </Shipment>
   </Part>
</Order>
```

Steps for using DB2 Web Services

- Show standard test page that is shipped as part of worf
- Show directories 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



/eb Service Samples Page

is page contains links for testing the sample Web Services.

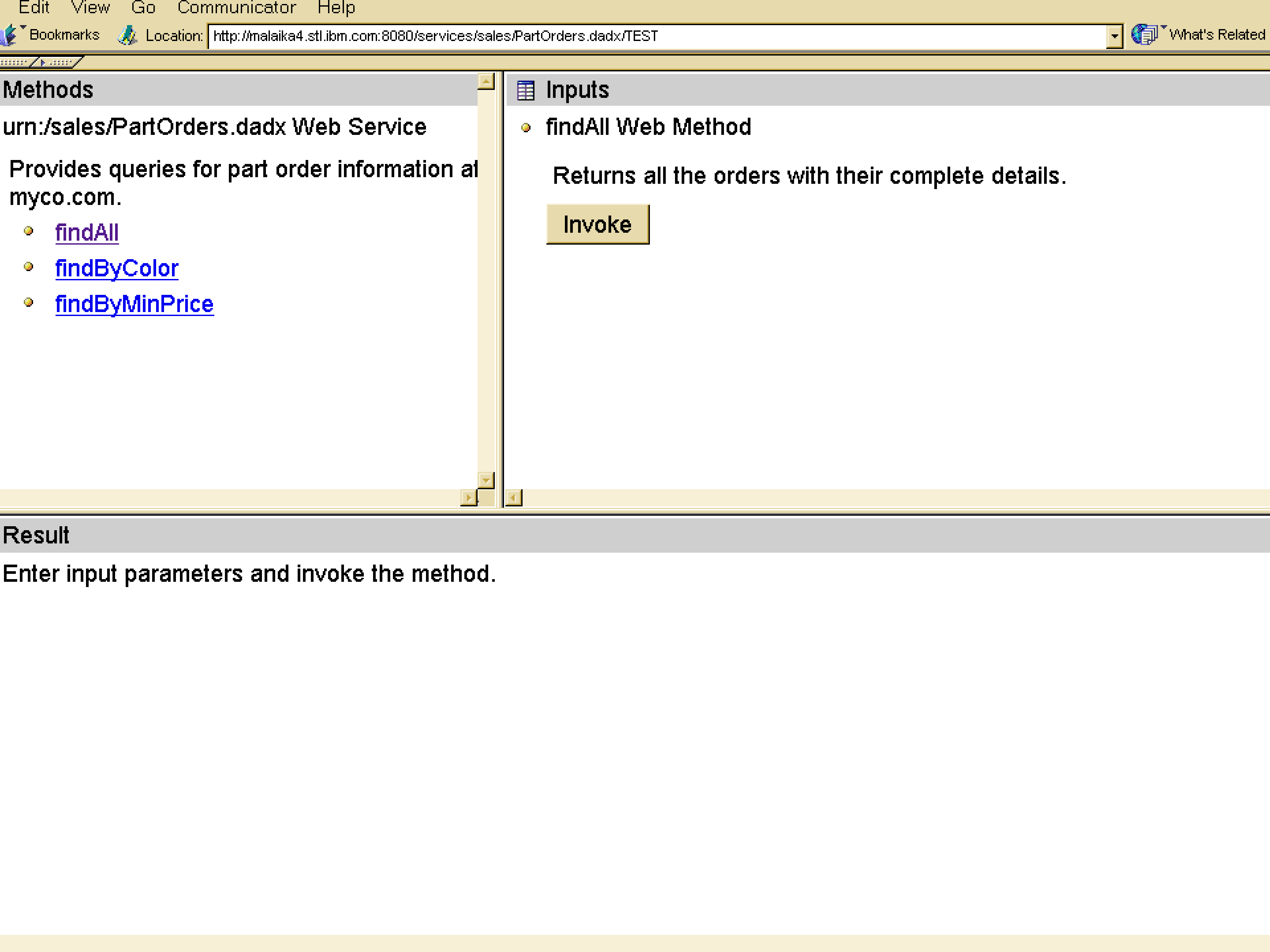
- View the list of deployed services using the SOAP Admin page.
- View the WSDL and XSD.
- Test the HTTP POST binding using the automatic and manual test pages.

Java Bean Samples

Service ID	WSDL	WSDLservice	WSDLbinding	XSD	
n:/beans/AddressBook.isd	WSDL	WSDLservice	WSDLbinding	XSD	TEST
n:/beans/Person.isd	WSDL	WSDLservice	WSDLbinding	XSD	TEST
n:/beans/TemperatureConverter.isd	WSDL	WSDLservice	WSDLbinding	XSD	TEST

DB2 XML Extender Samples

Service ID	WSDL	WSDLservice	WSDLbinding	XSD	
n:/sales/PartOrders.dadx	WSDL	WSDLservice	WSDLbinding	XSD	TEST
n:/sales/dan.dadx	WSDL	WSDLservice	WSDLbinding	XSD	TEST
n:/sales/PoiaPartOrders.dadx	WSDL	WSDLservice	WSDLbinding	XSD	TEST
n:/sales/SqlMappingPartOrders.dadx	WSDL	WSDLservice	WSDLbinding	XSD	TEST
n:/sales/RdbNodeMappingPartOrders.dadx	WSDL	WSDLservice	WSDLbinding	XSD	TEST
n:/sales/StorePartOrders.dadx	WSDL	WSDLservice	WSDLbinding	XSD	TEST
n:/sales/QueryPartOrders.dadx	WSDL	WSDLservice	WSDLbinding	XSD	TEST
n:/sales/UpdatePartOrders.dadx	WSDL	WSDLservice	WSDLbinding	XSD	TEST
n:/sales/CallPartOrders.dadx	WSDL	WSDLservice	WSDLbinding	XSD	TEST



http://malaika4.stl.ibm.com:8080/services/sales/PartOrders.dadx/TEST

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.

Result

Enter input parameters and invoke the method.

DADX Example PartOrders.dadx – Part 1

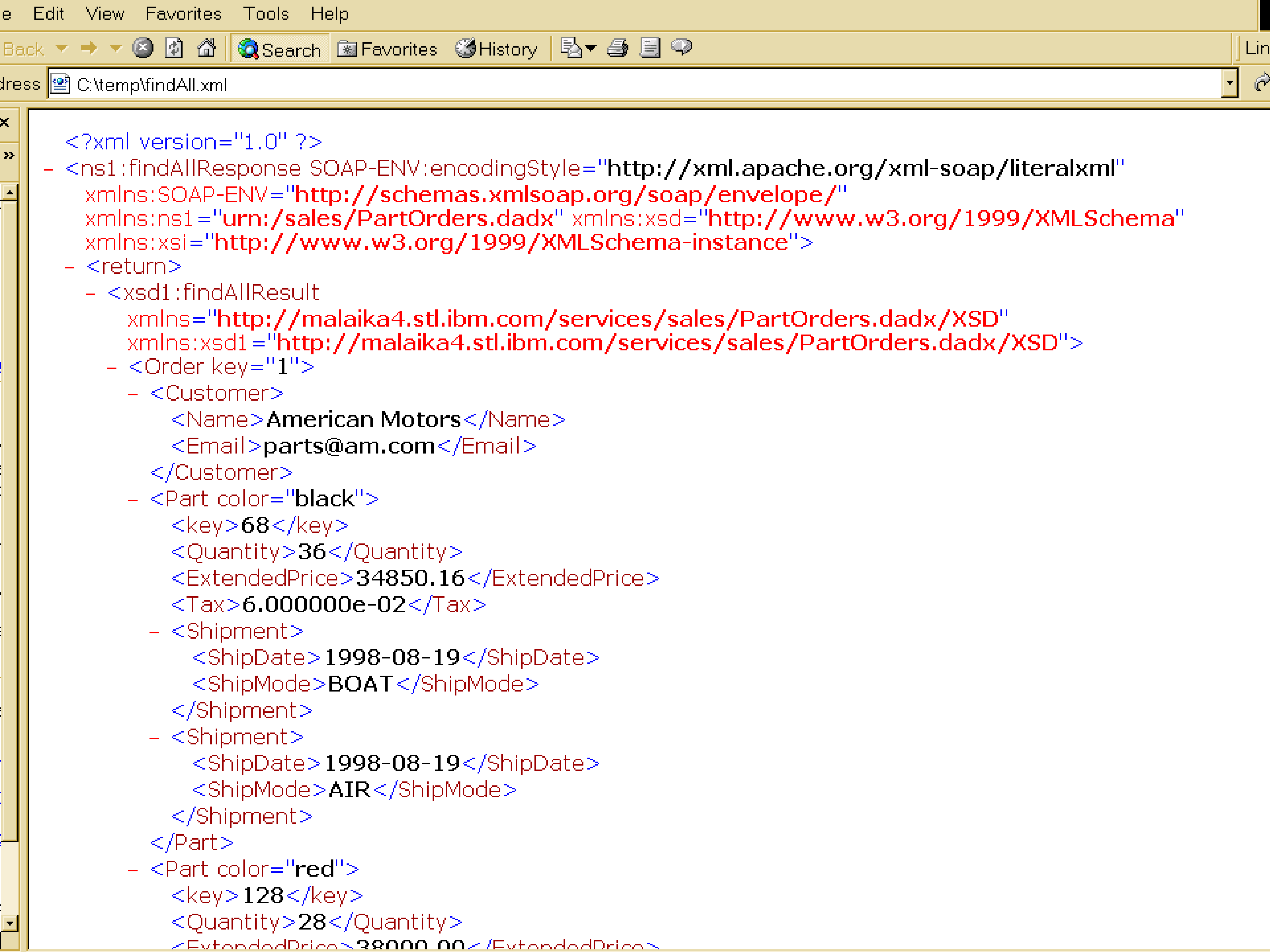
```
<?xml version="1.0"?>
<DADX xmlns="urn:ibm.com:dxx:dadx"
     xmlns:xsd="http://www.w3.org/1999/XMLSchema">
 <documentation <documentation xmlns="http://schemas.xmlsoap.org/wsdl/">
 Provides queries for part order information at myco.com.</documentation>

 <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

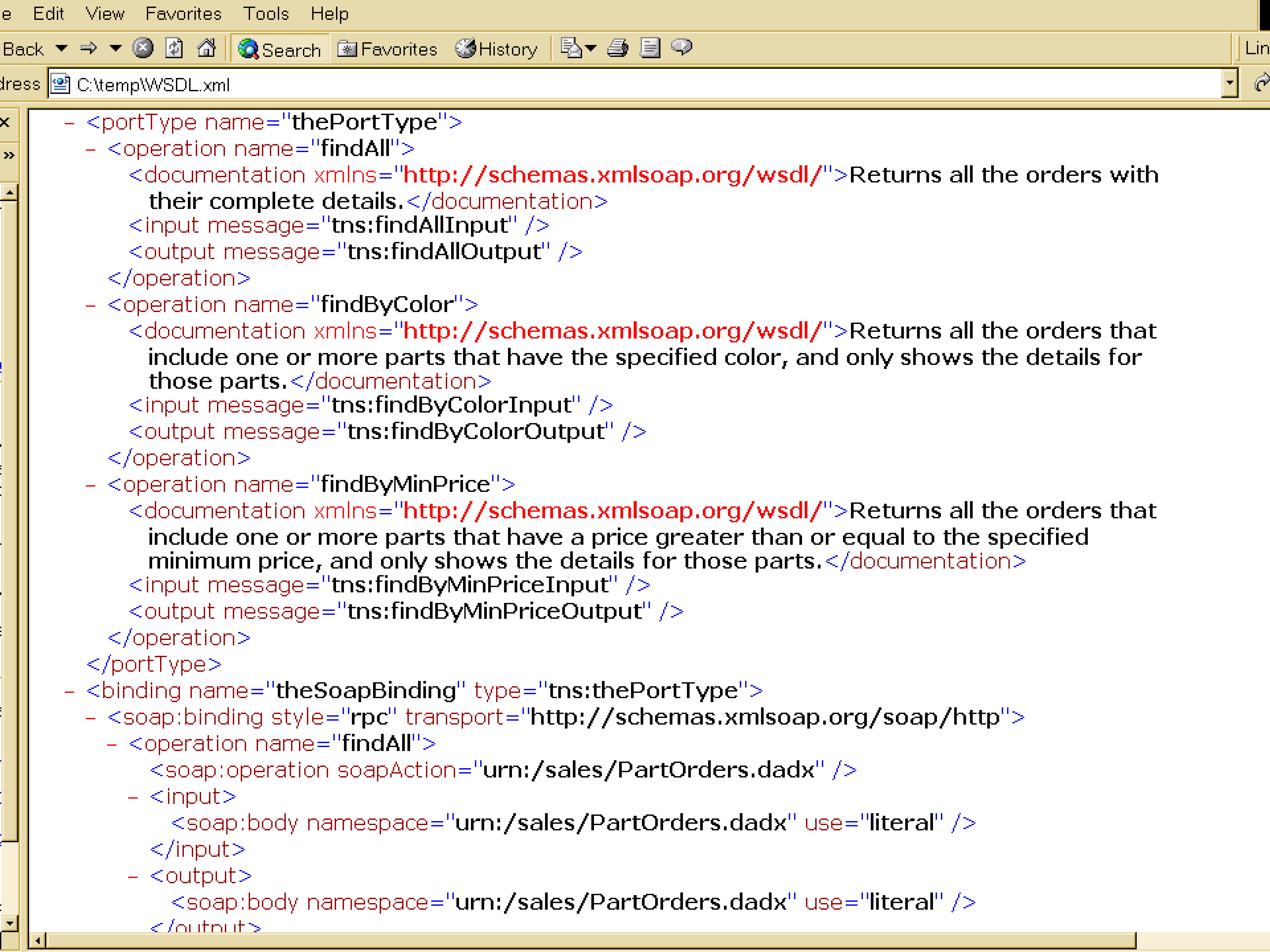
```
<operation name="findByColor">
  <documentation xmlns="http://schemas.xmlsoap.org/wsdl/">Returns all the orders
   that include one or more parts that have the specified color, and only shows
   the details for those parts.</documentation>
  <retrieveXML>
    <DAD_ref>getstart_xcollection.dad</DAD_ref>
    <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>
```



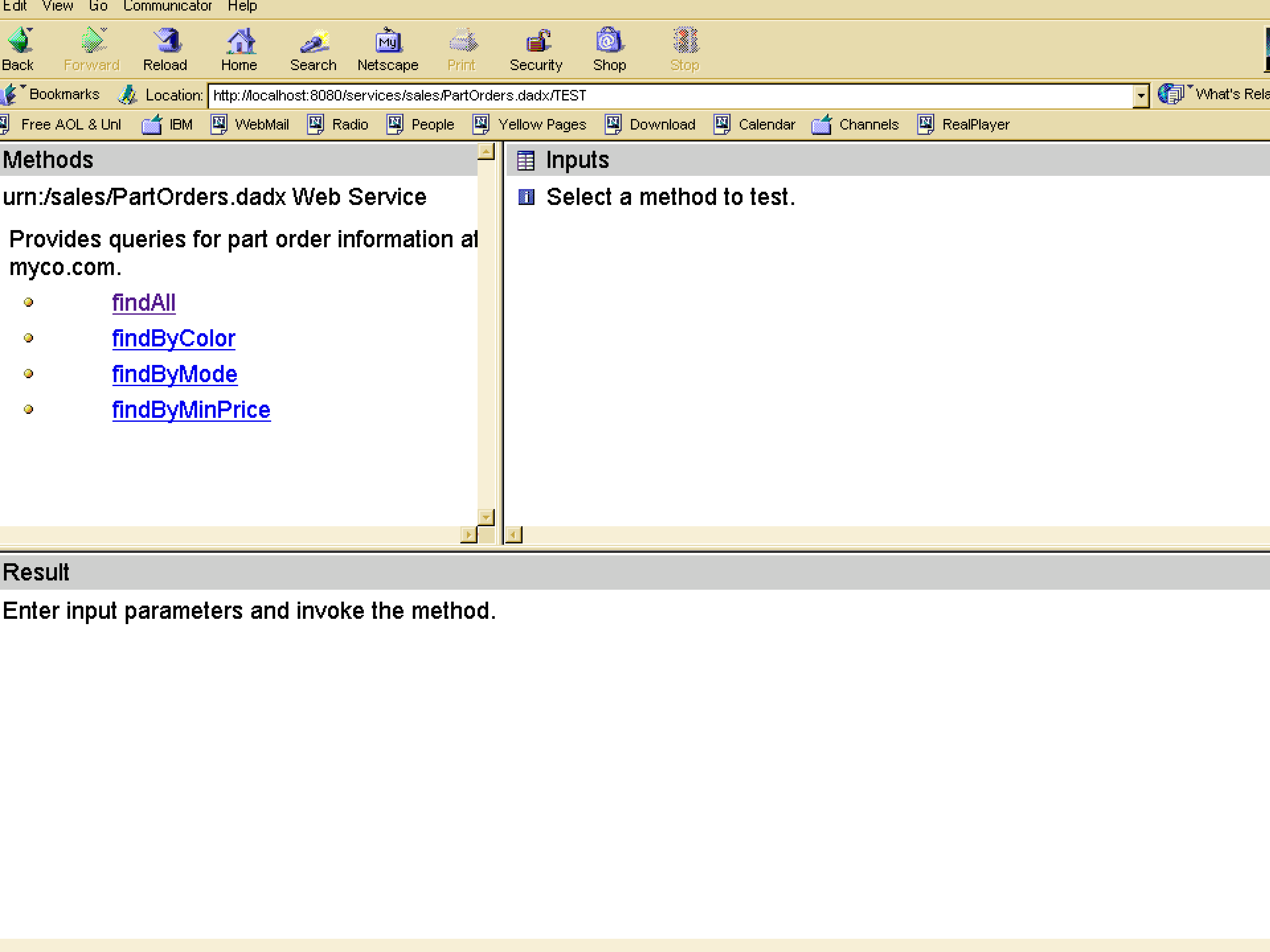
```
  <?xml version="1.0" ?>
- <ns1:findAllResponse SOAP-ENV:encodingStyle="http://xml.apache.org/xml-soap/literalxml"
   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:ns1="urn:/sales/PartOrders.dadx" xmlns:xsd="http://www.w3.org/1999/XMLSchema"
   xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance">
  - <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>
            <ExtendedPrice>38000.00</ExtendedPrice>
```


e Edit View Favorites Tools Help
Back Search Favorites History
dress C:\temp\WSDL.xml

```
  - <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" />
          </output>
```

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>
```



Methods

urn:/sales/PartOrders.dadx Web Service

Provides queries for part order information at myco.com.

- findAll
- findByColor
- findByMode
- findByMinPrice

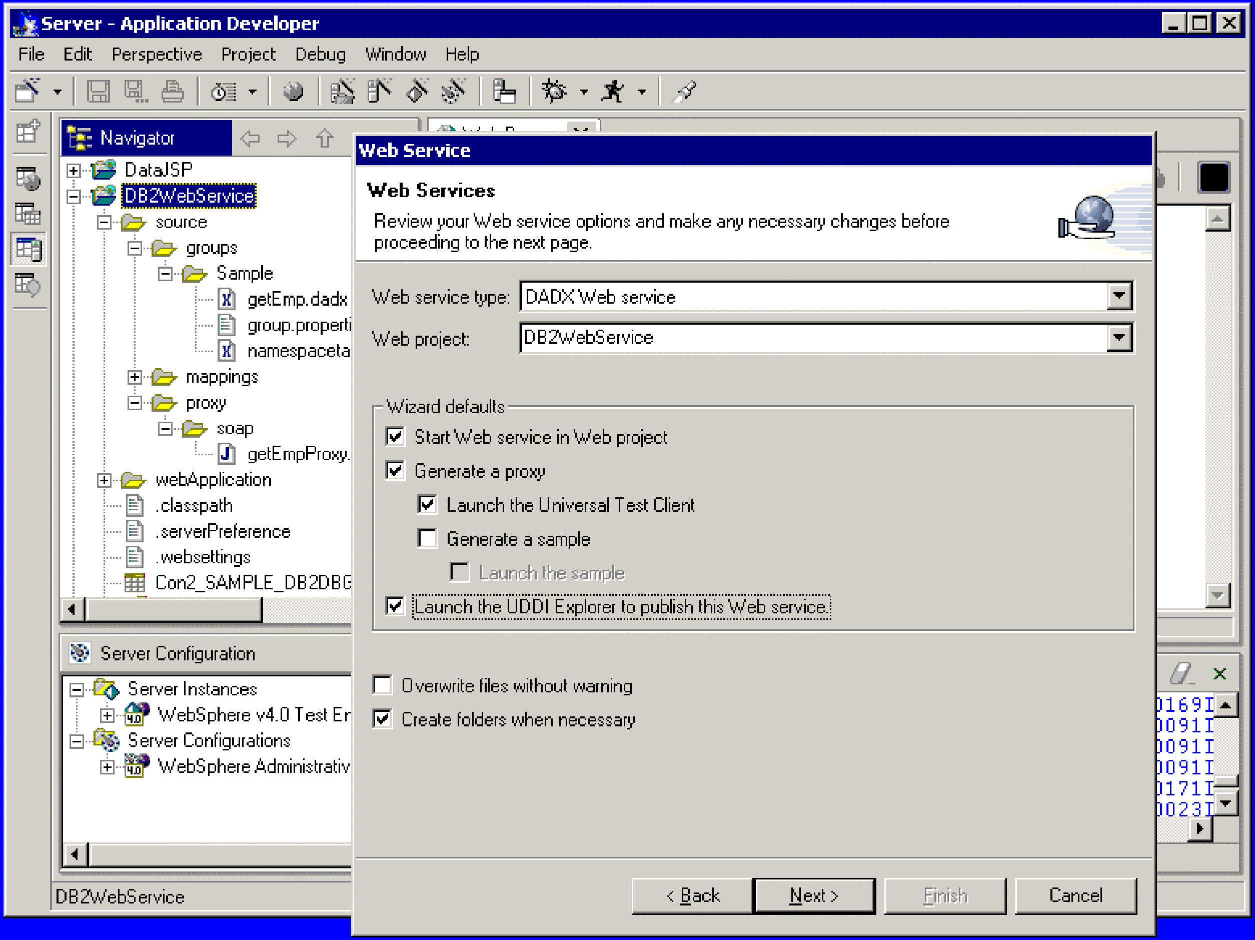
Inputs

Select a method to test.

Result

Enter input parameters and invoke the method.

WebSphere Studio



Web Service

Web Services

Review your Web service options and make any necessary changes before proceeding to the next page.

Web service type: DADX Web service

Web project: DB2WebService

Wizard defaults
- [✓] Start Web service in Web project
- [✓] Generate a proxy
  - [✓] Launch the Universal Test Client
  - [ ] Generate a sample
    - [ ] Launch the sample
- [✓] Launch the UDDI Explorer to publish this Web service.

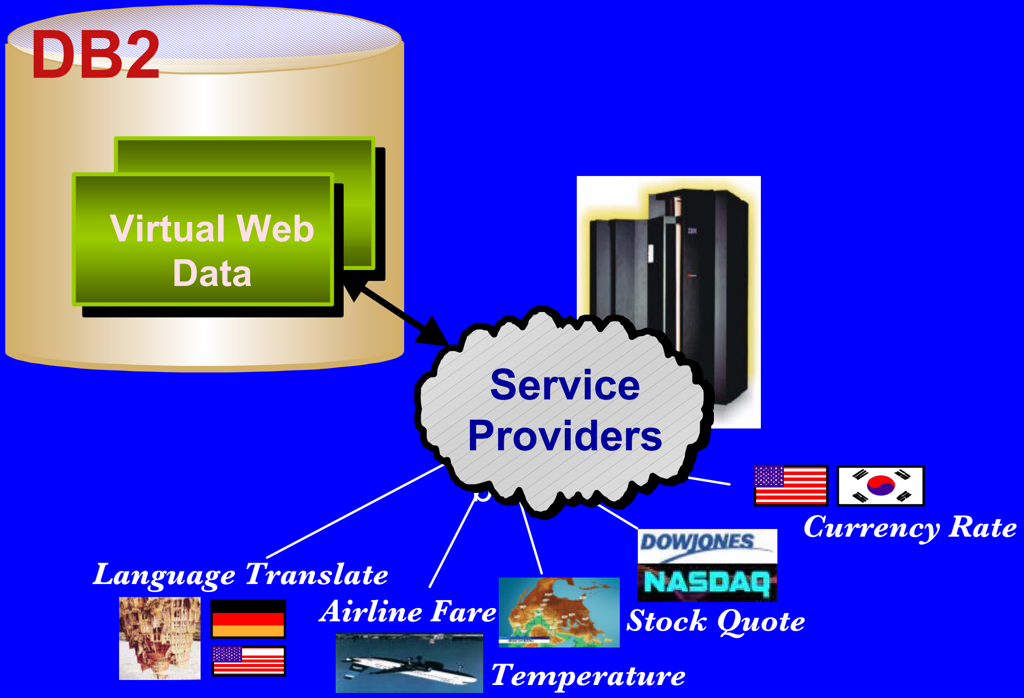
- [ ] Overwrite files without warning
- [✓] Create folders when necessary

< Back | Next > | Finish | Cancel

Web Services Directions

- More Platforms
- Axis Interfaces
- XQuery (XML Query Language)
- Web Services Security
- Web Services Transactions
- Other items:
  - Web Services form the underpinning for grid computing

More Directions: Accessing Web Services from DB2



```
SELECT city,
GetTemperature(city)
FROM location
```

- Integrate SQL statements and Web Service invocations
- Support for generating SQL scalar and table UDFs based on WSDL web service description
  - Command line version
  - Tool support integrated into Web Sphere Studio

XML Resources: From IBM

- International Components for Unicode (ICU)
  - http://oss.software.ibm.com/icu/
- IBM XML parsers XML4C, XML4J, Xalan-C and Xalan-J (LotusXSL)
  - http://www.alphaworks.ibm.com/
- IBM XML Toolkit (parsers etc) for OS390 (zSeries)
  - http://www.ibm.com/servers/eserver/zSeries/software/xml/
- IBM Developerworks for XML
  - http://www.ibm.com/developerworks/xml/
- IBM Developerworks for Web Services
  - http://www.ibm.com/developerworks/webservices/
- WebSphere Studio: http://www.ibm.com/software/ad/studioappdev/
- DB2 XML Extender
  - http://www.ibm.com/software/data/db2/extenders/xmlext/
- DB2 XML Extender Hints and Tips
  - http://www.ibm.com/software/data/db2/extenders/xmlext/support.htm
- DB2 Web Services http://www7b.boulder.ibm.com/dmdd/zones/webservices/
- Xperanto
  - http://www.ibm.com/software/data/developer/demos/xperanto/

XML Resources: IBM Papers

- Red books and red papers
  - http://www.redbooks.ibm.com/
    - Integrating XML with DB2 XML Extender and DB2 Text Extender SG24-6130
    - DB2 for OS/390 and z/OS Powering the World's e-business Solutions SG24-6257 (Chapter on XML Extender)
    - DB2 XML Extender Hints and Tips (red paper)
      - http://www.redbooks.ibm.com/redpapers/pdfs/redp0135.pdf.
- 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

XML Resources: IBM Papers and Downloads

Download DB2 Web Services V7.2

http://www7b.software.ibm.com/dmdd/zones/webservices/worf/

(also available in DB2 V8.1 (Windows & UNIX) and in WebSphere Studio)

DB2 and XML Web Services Papers

- DB2 and Web Services: The Big Picture
  http://www7b.boulder.ibm.com/dmdd/zones/webservices/bigpicture.html
- Running DB2® Web Services on WebSphere® Application Server Advanced Edition 4.0 by Reto Preisig
  http://www7b.boulder.ibm.com/dmdd/library/techarticle/preisig/0108preisig.html

DAD Checker

http://www.ibm.com/software/data/db2/extenders/xmlext/download/beta/dadcheck_rn.html

Web Services Summary

- Web Services Roles
  - Requestor (sometimes called consumer)
  - Provider
- Web Services Standards
  - WSDL (Web Services Description Language)
    - Interface description
  - SOAP (Simple Object Access Protocol)
    - Message content
  - UDDI (Universal Description, Discovery, and Integration)
    - Web Services Registry
- Web Services Activities
  - Providing
  - Invoking
  - Publishing
  - Discovering

DB2 Web Services Summary

- DB2 as Web Service Provider
  - Available now
    - Web download V7.2
    - Part of DB2 V8.1
    - Part of WebSphere Studio
  - Interface described through DADX
    - Generates WSDL
    - Three types of support: Regular SQL, SPs, XML Extender
  - Runtime implemented through DB2 WORF
  - WebSphere Studio assists in building DADXs
- Direction: DB2 as a Web Service Requestor (Consumer)