IBM Software Group

WORF:
Developing DB2 UDB based Web Services on a Websphere Application Server

DB2 Information Management Software

e business on demand software

Agenda

- **DB2 UDB v8.1:**
  - **Storing Data**
  - **Integration with Business Logic**
  - **Integration with XML**
- **DB2 Web Services:**
  - **SOAP service communication with DB2: WORF**
  - **Web Service specification: DADX document**
  - **SQL based queries**
  - **XML based queries**
- **Calling DB2 Web Services:**
  - **Changes to Client Programs**

IBM Software Group

DB2 UDB v8.1

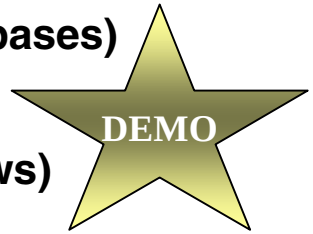
e business on demand software

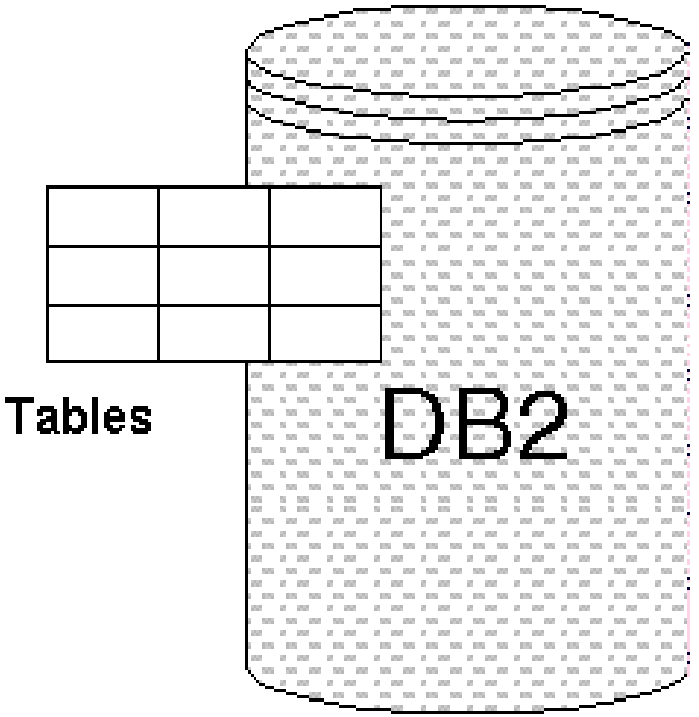
Agenda

- **DB2 UDB v8.1:**
  - ***Storing Data***
  - **Integration with Business Logic**
  - **Integration with XML**
- **DB2 Web Services:**
  - **SOAP service communication with DB2: WORF**
  - **Web Service specification: DADX document**
  - **SQL based queries**
  - **XML based queries**
- **Calling DB2 Web Services:**
  - **Changes to Client Programs**

Storing data… is evolved to:

- **Checking data (Constraints)**
- **Maintaining data (Triggers, Generated Columns)**
- **Gathering data (Federated Databases)**
- **Tailoring data (Updateable Views)**
- **Generating data (Identity Columns, Generated Sequence)**



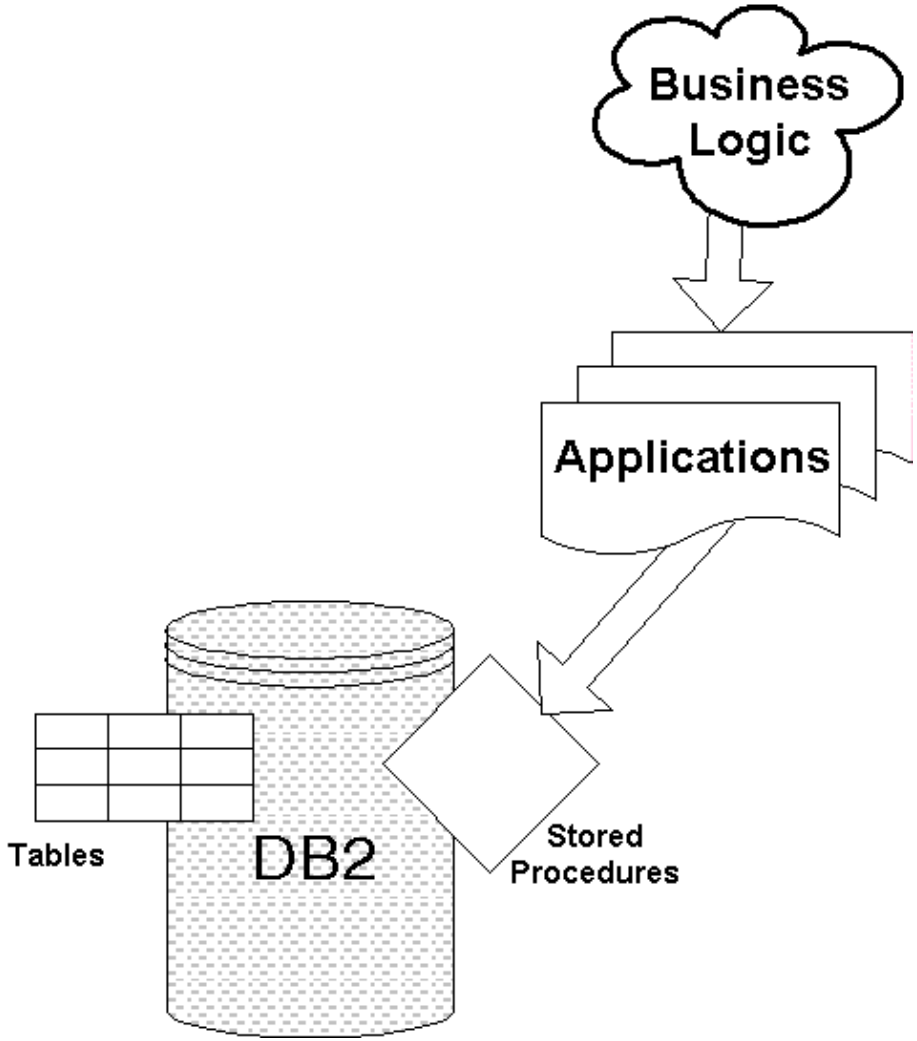


Agenda

- **DB2 UDB v8.1:**
  - **Storing Data**
  - ***Integration with Business Logic***
  - **Integration with XML**
- **DB2 Web Services:**
  - **SOAP service communication with DB2: WORF**
  - **Web Service specification: DADX document**
  - **SQL based queries**
  - **XML based queries**
- **Calling DB2 Web Services:**
  - **Changes to Client Programs**

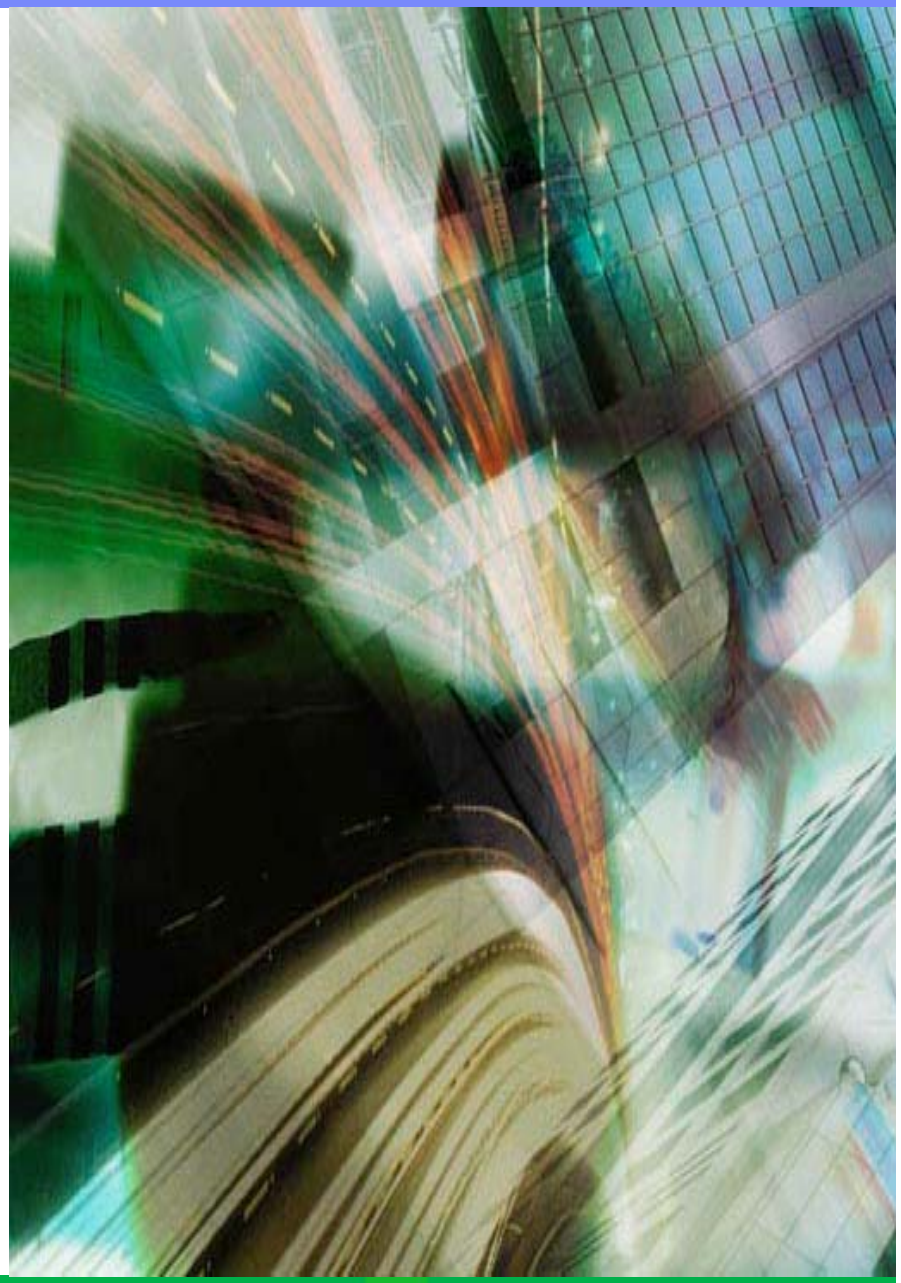
Integration with Business Logic

Stored Procedures



Advantages of Stored Procedures

- **Faster execution**
- **Reduced network traffic**
- **Modular programming**
- **Increased security**



Writing Stored Procedures

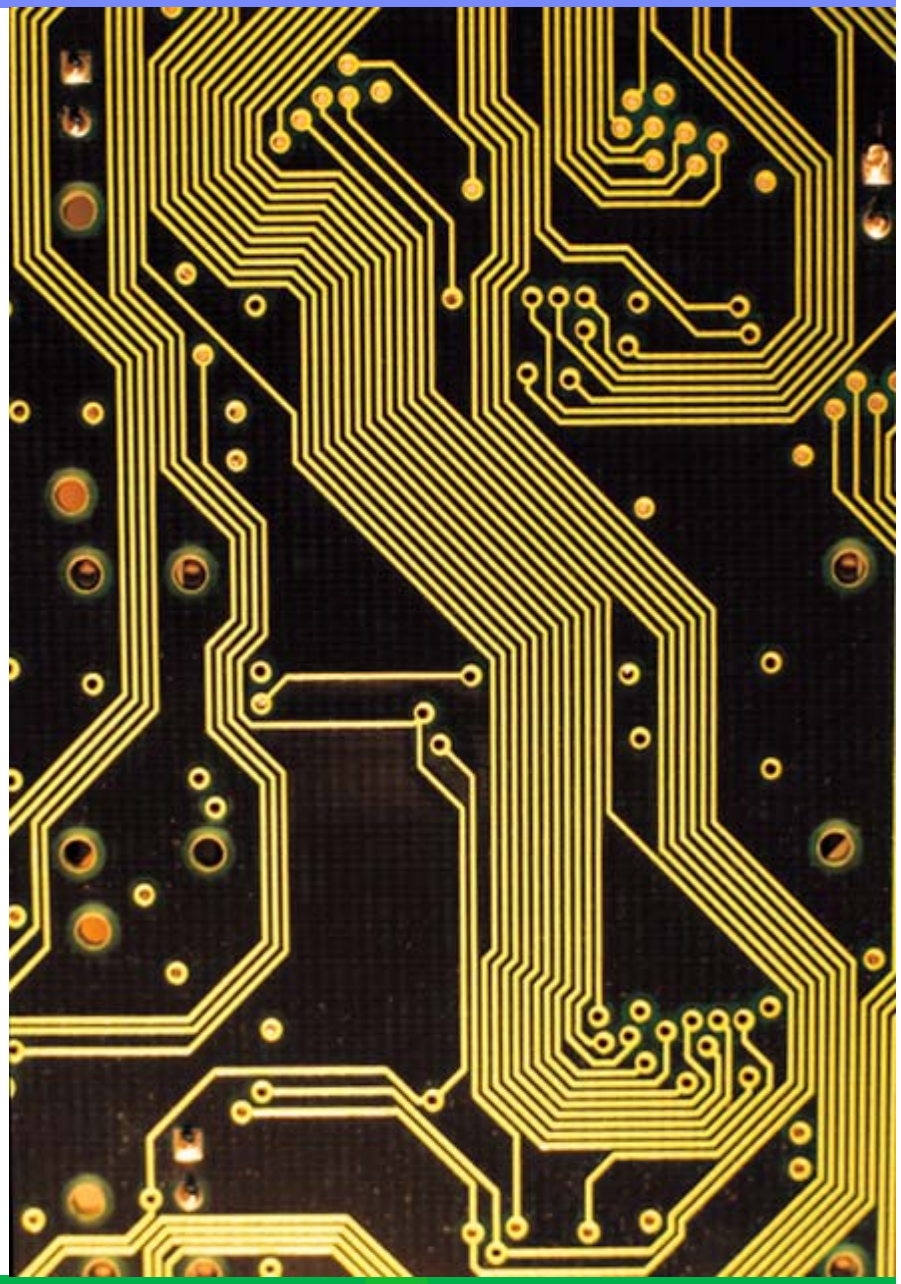
- **Languages: Java, C,…**
- **Generation with GUI:**
  - **Stored Procedure Builder (v7)**
  - **Development Center (v8)**

DEMO

- **SQL/PL**

DEMO

DB2 UDB v8 fully supports the dynamic CALL statement



Agenda

- **DB2 UDB v8.1:**
  - **Storing Data**
  - **Integration with Business Logic**
  - ***Integration with XML***
- **DB2 Web Services:**
  - **SOAP service communication with DB2: WORF**
  - **Web Service specification: DADX document**
  - **SQL based queries**
  - **XML based queries**
- **Calling DB2 Web Services:**
  - **Changes to Client Programs**

What is an XML-document?

- **Adds metadata to your data**
- **Hierarchical structure**
- **Consists of Elements and Attributes**

```
<Course id="1">
    <Title>DB2 UDB for OS/390</Title>
    <Description>This Course reveals the secrets of the leading RDBMS ...
    </Description>
    <Prereq_list>
        <Prereq>Know the concepts of a relational Database</Prereq>
        <Prereq>Know some basic SQL</Prereq>
        <Prereq>Have some experience on an OS/390 platform</Prereq>
    </Prereq_list>
</Course>
```

Structure of XML-document is defined: Schema / DTD

```
<!ELEMENT Course (Title,Description,Prereq_list)>
<!ATTLIST Course id CDATA #REQUIRED>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Description (#PCDATA)>
<!ELEMENT Prereq_list (Prereq+)>
<!ELEMENT Prereq (#PCDATA)>
```

DTD:

Document Type Definition

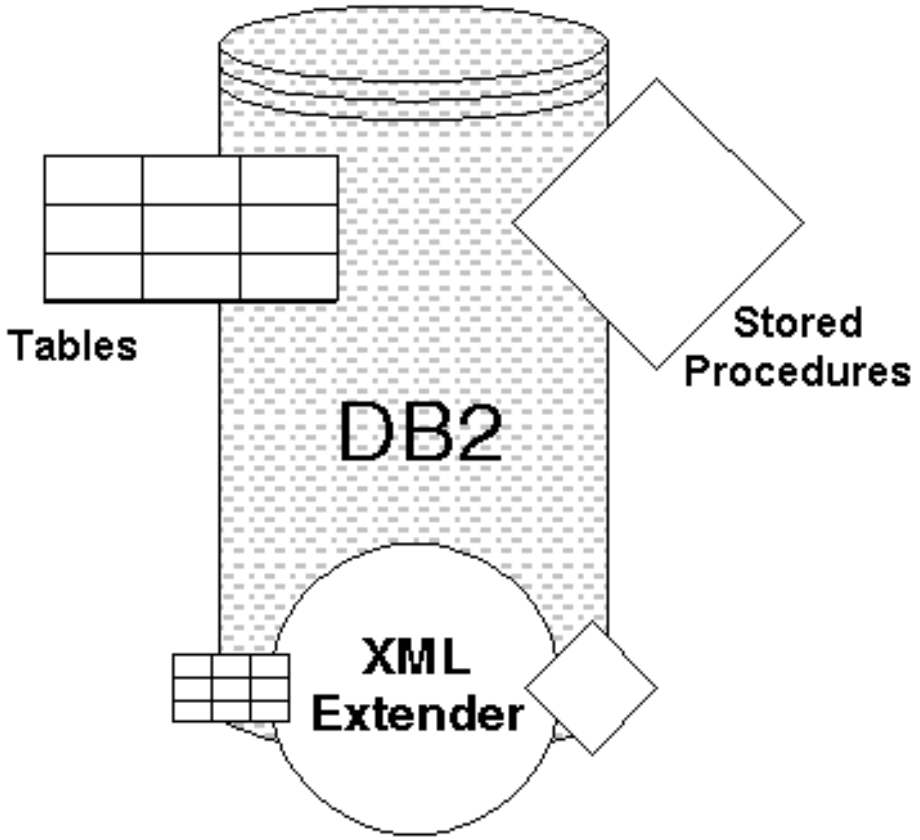
```
<Course id="1">
    <Title>DB2 UDB for OS/390</Title>
    <Description>This Course reveals the secrets of the leading RDBMS ...
    </Description>
    <Prereq_list>
        <Prereq>Know the concepts of a relational Database</Prereq>
        <Prereq>Know some basic SQL</Prereq>
        <Prereq>Have some experience on an OS/390 platform</Prereq>
    </Prereq_list>
</Course>
```

The DB2 XML Extender offers you...

- **Stored Procedures**
- **Triggers**
- **User Defined Functions (UDF)**
- **User Defined Datatypes (UDT)**
- **Supporting tables**

…to extend the DB2 functionality!

The DB2 XML Extender:



Preparing your Database for use by XML Extender

Using the DB2 Command Line Processor

- **Use the administration interface**
  - `dxxadm …`
- **Enable your database**
  - `dxxadm enable_db <database>`

Using the XML Extender GUI / administration wizard.

- **Start the administration wizard**
  - `java com.ibm.dxx.admin.Admin`
  - `jre com.ibm.dxx.admin.Admin`
- **Connect to your database**
- **Enable the database**

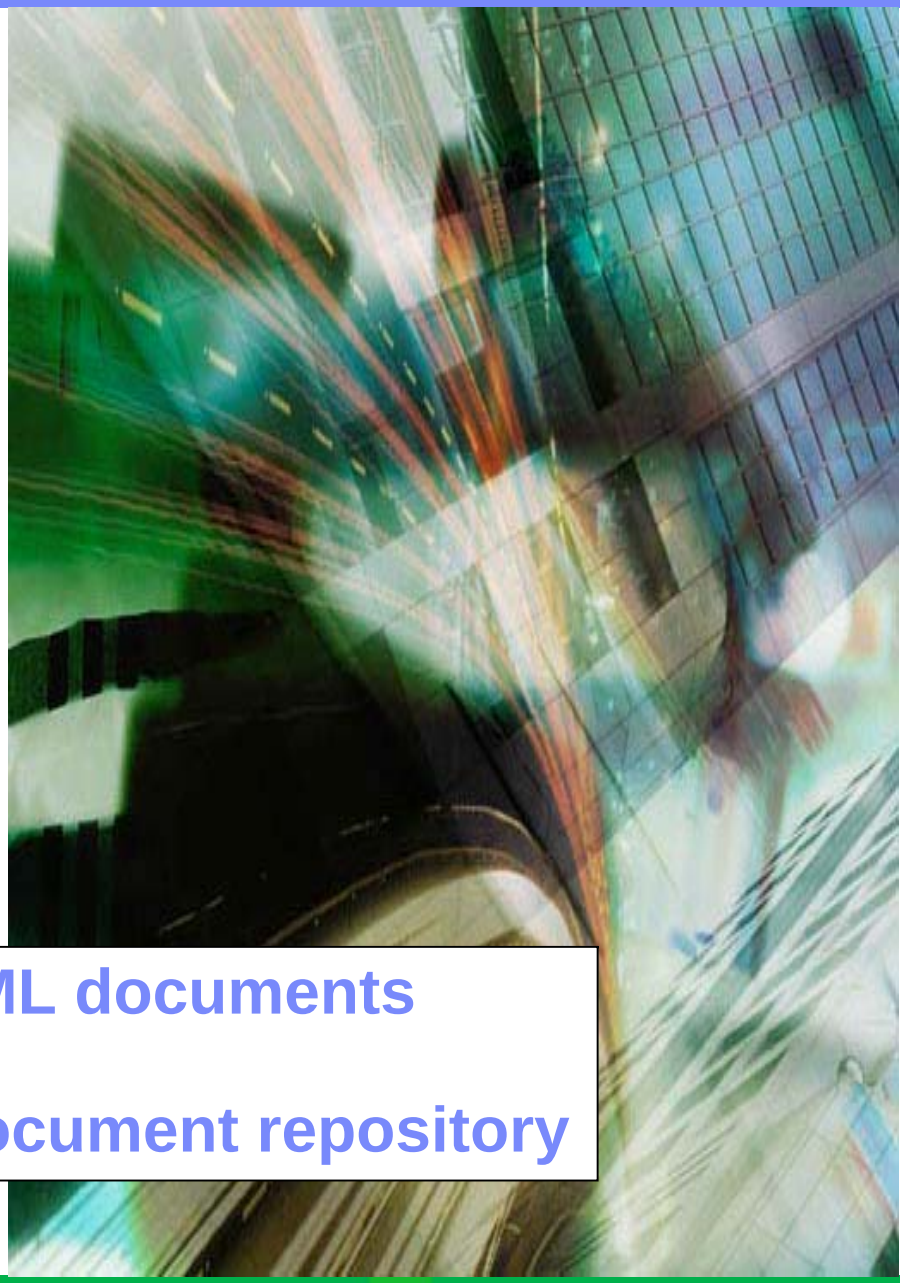
Two methods to work with XML in DB2

XColumn

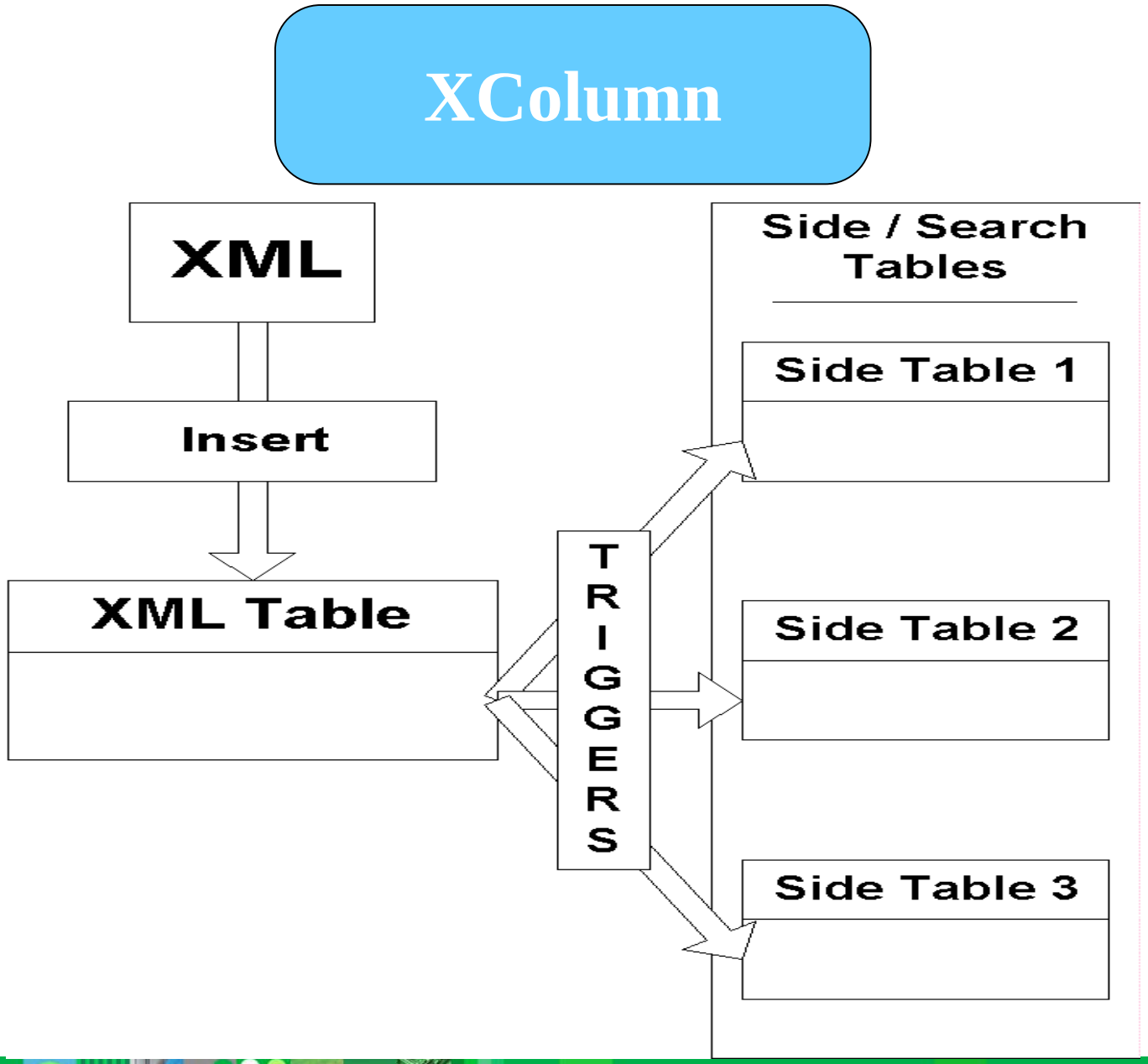
XCollection

XColumn

- **Validation of XML documents**
- **Storing XML documents**
- **Fast Search through XML documents**



Column data = XML documents
or
DB2 is used as XML document repository

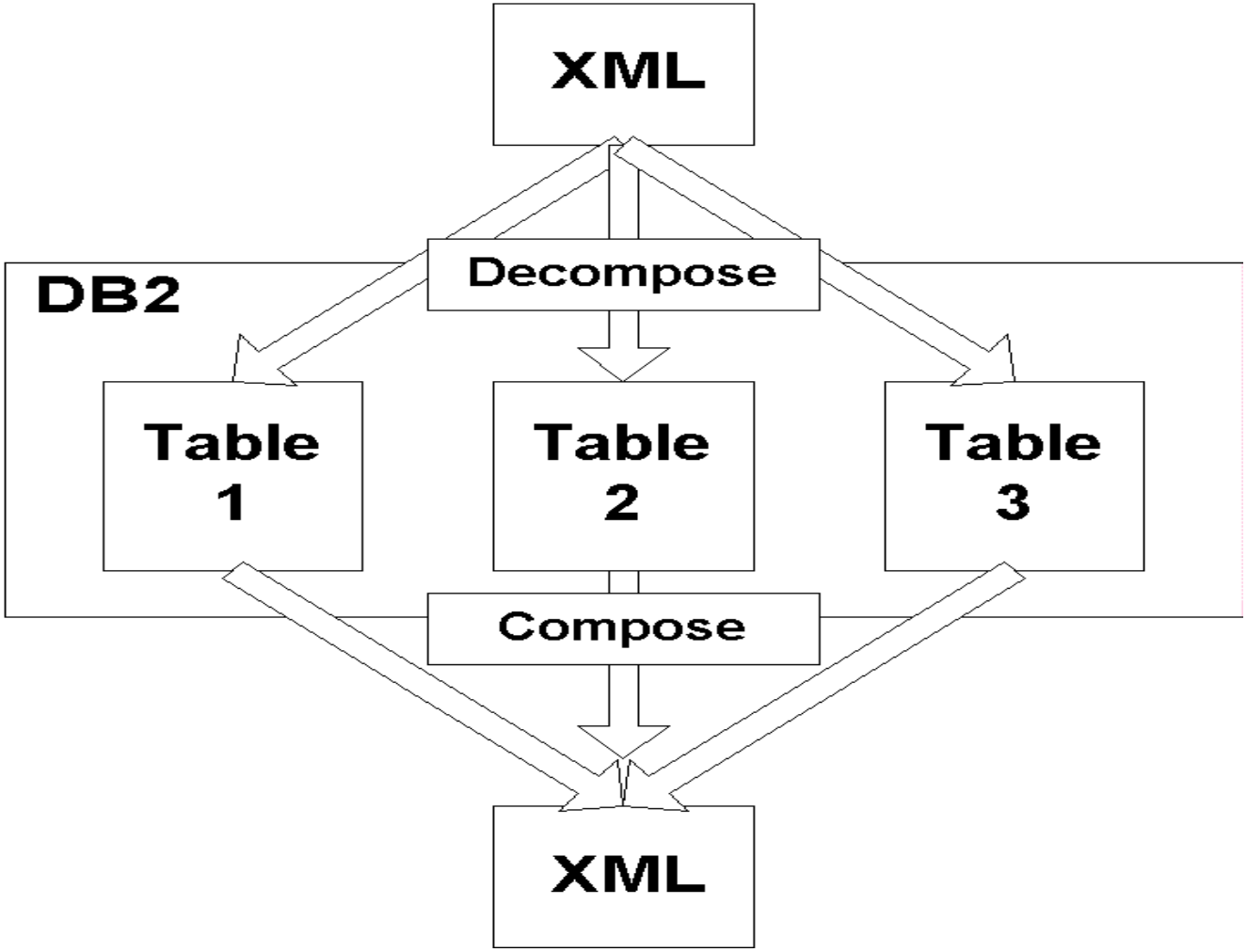

XColumn
XML
Insert
XML Table
TRIGGERS
Side / Search Tables
Side Table 1
Side Table 2
Side Table 3

XCollection

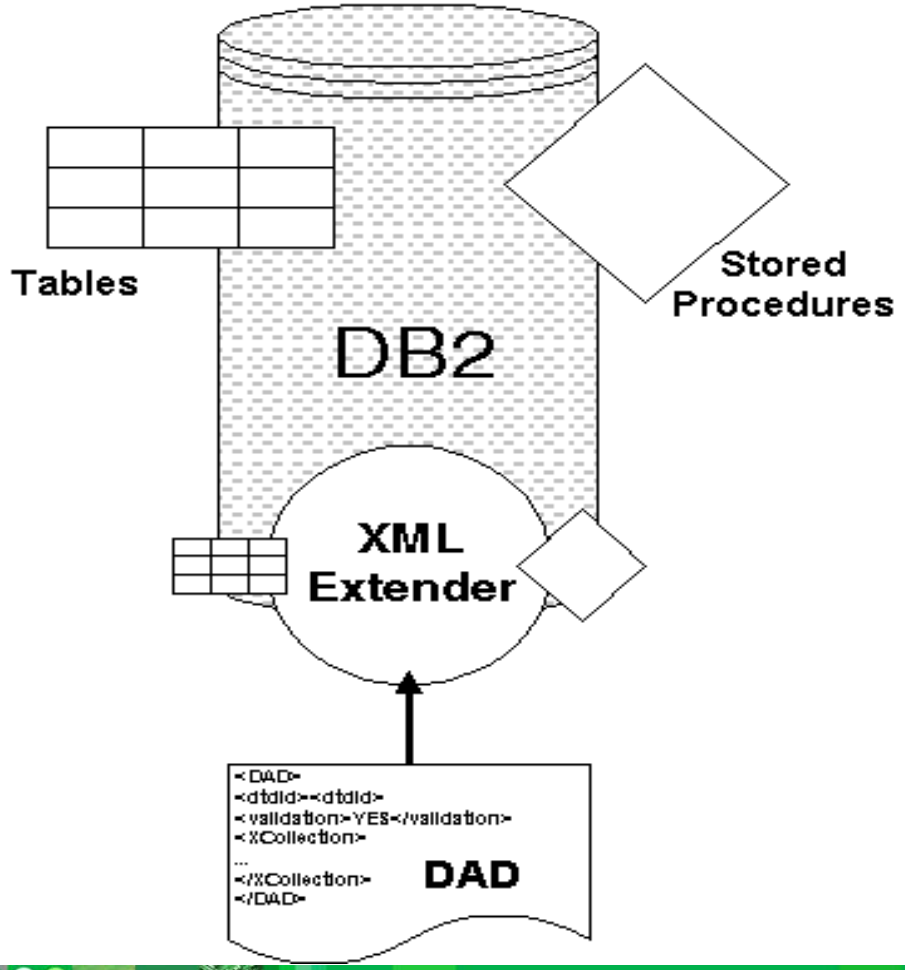
- **Validation of XML documents**
- **Decomposing XML documents to DB2 data**
- **Composing XML documents from DB2 data**

DB2 is used as *data*base (no XML in DB2)
or
XML is used as “transport language” to / from DB2

XCollection



Communication with XML Extender through DAD's (Document Access Definition)



DAD = XML Extender option description

- **XML document**
  - **Validate: YES / NO**
  - **DTD-id**
  - **Method: XColumn / XCollection**
  - **Detailed description of data mapping: DB2 vs. XML**

```
<DAD>
<dtdid>e:\xmldocs\companyRegistration.dtd<dtdid>
<validation>YES</validation>
<XCollection>
...
</XCollection>
</DAD>
```

(De)composing XML document to relational data: RDB-node mapping

- **Table specifications**
- **Join conditions**
- **Attribute - column mapping**
- **Element - column mapping**

XCollection

RDB-node mapping

- **Table specifications**
- **Join conditions**

XCollection

```
<Xcollection>
    <prolog>?xml version="1.0" encoding="UTF-8"?</prolog>
    <doctype>!DOCTYPE companyRegistration SYSTEM "E:\XMLdocs\companyRegistration.dtd”
    </doctype>
    <root_node>
        <element_node name="companyRegistration">
            <RDB_node>
                <table name="companies" key="id"/>
                <table name="persons" key="sofi"/>
                <table name="loans"/>
                <condition>companies.company_id = loans.loan_company_id
                AND persons.person_sofi = loans.loan_person_sofi</condition>
            </RDB_node>
```

RDB-node mapping

- **Attribute - column mapping**

```
<attribute_node name="id">
        <RDB_node>
                <table name="companies"/>
                <column name="company_id" type="smallint"/>
        </RDB_node>
</attribute_node>
```

XCollection

- **Element - column mapping**

```
<element_node name="name">
        <text_node>
                <RDB_node>
                        <table name="companies"/>
                        <column name="company_name" type="char(10)"/>
                </RDB_node>
        </text_node>
</element_node>
```

Composing XML document from relational data: SQL mapping

XCollection

- **SQL statement**
- **Attribute - column mapping**
- **Element - column mapping**

SQL mapping

- **SQL statement**

XCollection

```
<Xcollection>
   <SQL_stmt>select person_sofi, person_lastname, person_firstname, loan_no,
         loan_nrmonths * loan_monthlypayment as atp
         from db2.persons, db2.loans
         where person_paybehaviour = 'B'
            and loan_person_sofi = person_sofi
            and loan_company_id = 12
         order by person_sofi, loan_no
   </SQL_stmt>
   <prolog>?xml version="1.0" encoding="UTF-8"?</prolog>
   <doctype>!DOCTYPE list_loans SYSTEM "E:\UTD-seminar\XMLdocs\list-loans.dtd"</doctype>
```

SQL mapping

- **Column - attribute mapping**

XCollection

```
<element_node name="loan_list">
    <element_node name="loan" multi_occurrence="YES">
        <attribute_node name="no">
            <column name="loan_no" type="smallint"/>
        </attribute_node>
```

- **Column - element mapping**

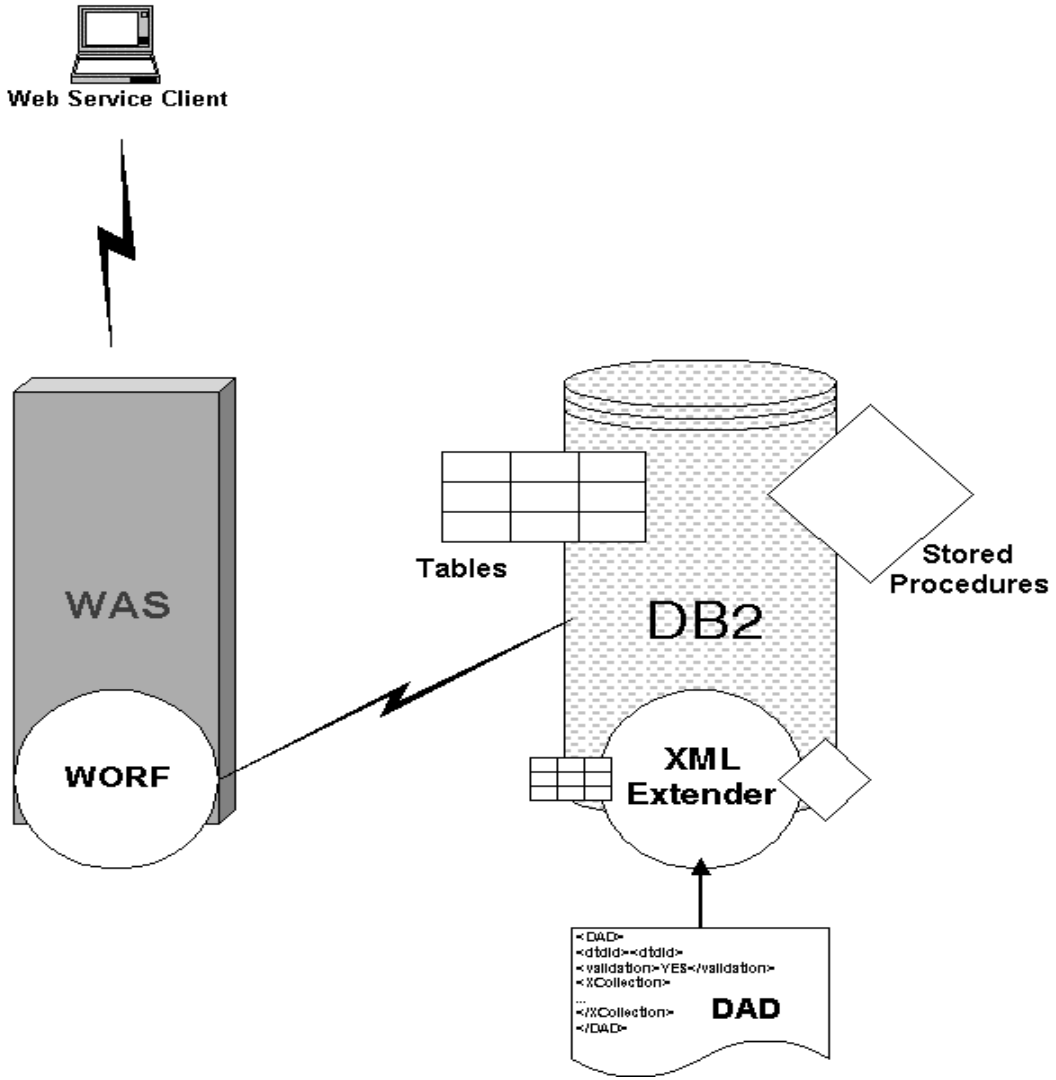

```
    <element_node name="amount_to_pay">
        <text_node>
            <column name="atp" type="integer"/>
        </text_node>
    </element_node>
```


IBM Software Group
DB2 Web Services
DB2 Information Management Software
e business on demand software

Agenda

- **DB2 UDB v8.1:**
  - **Storing Data**
  - **Integration with Business Logic**
  - **Integration with XML**
- **DB2 Web Services:**
  - ***SOAP service communication with DB2: WORF***
  - **Web Service specification: DADX document**
  - **SQL based queries**
  - **XML based queries**
- **Calling DB2 Web Services:**
  - **Changes to Client Programs**

Basic Architecture



WORF: a piece of software plugged in the Websphere application server used by all DB2 Web Service enterprise applications.

Web Service Object Runtime Framework (WORF) provides the features:

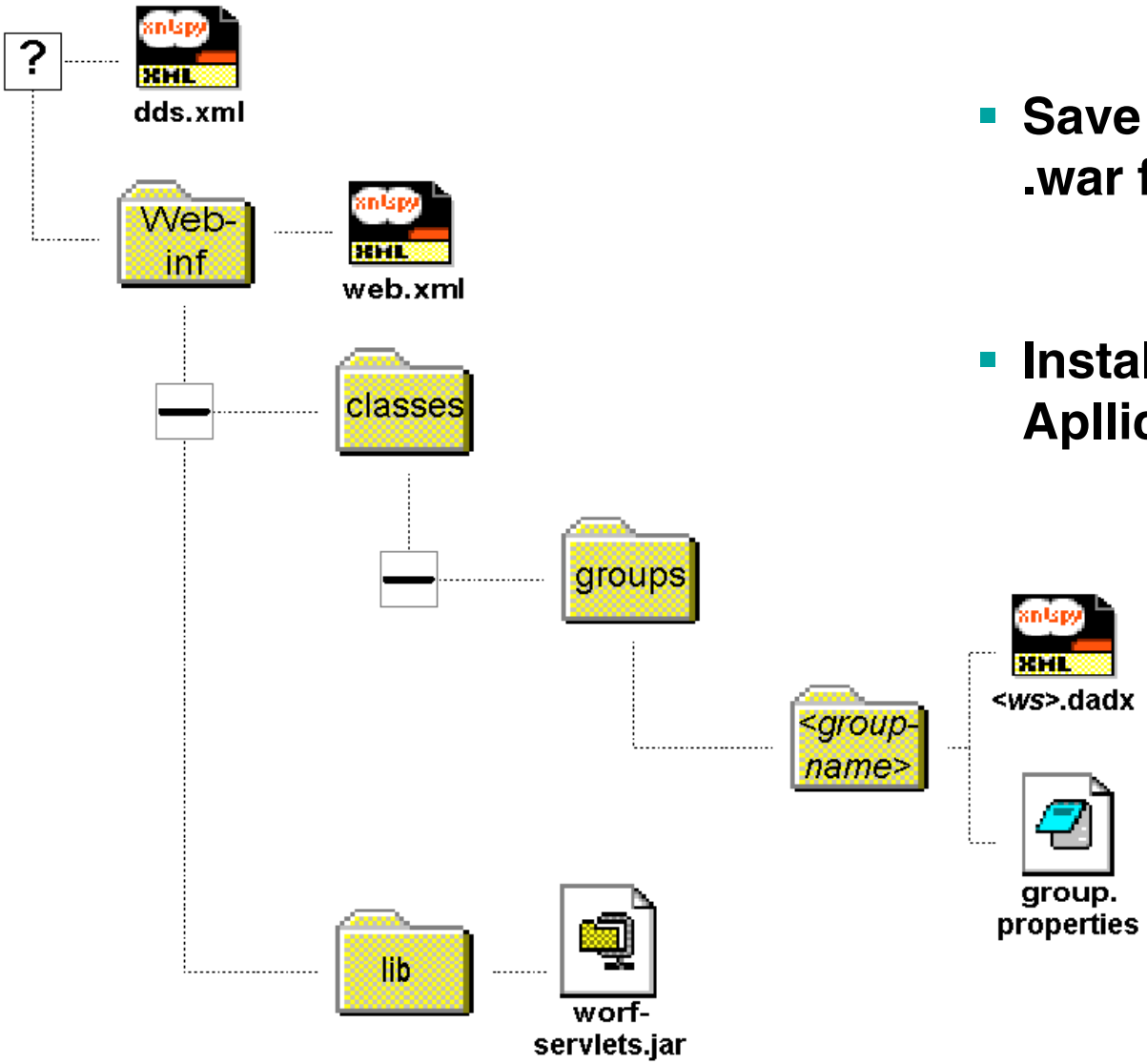
- **Resource-based deployment and invocation, i.e.**
  - **DADX based**
  - **optionally other resources that help define the web service**
- **Automatic service redeployment**
- **Automatic WSDL and XSD generation**
- **Automatic documentation**
- **Automatic test page generation**

Necessary files for a WORF web service deployment:

- **web.xml**
  - **Servlet-mapping: mapping between URL and physical directory**
- **dds.xml (deployment descriptor)**
  - **Description of each deployed web service**
- **group.properties**
  - **For each group of web services: database specifications**
    - database-name
    - userid
    - password
    - ...
- ***<db2 web service name>*.dadx**

Together with a “worf-servlets.jar” file
Embedded in a predefined directory structure

Predefined directory structure



- **Save this structure into a .war file**
- **Install the .war as Enterprise Apllication in the WAS**

Agenda

- **DB2 UDB v8.1:**
  - **Storing Data**
  - **Integration with Business Logic**
  - **Integration with XML**
- **DB2 Web Services:**
  - **SOAP service communication with DB2: WORF**
  - ***Web Service specification: DADX document***
  - **SQL based queries**
  - **XML based queries**
- **Calling DB2 Web Services:**
  - **Changes to Client Programs**

The DADX document: Document Access Definition eXtension

- **XML document**
- **a DADX for each Web Service**
- **Web Service methods = DADX operations**
- **Web Service and method documentation**
- **Query Description**

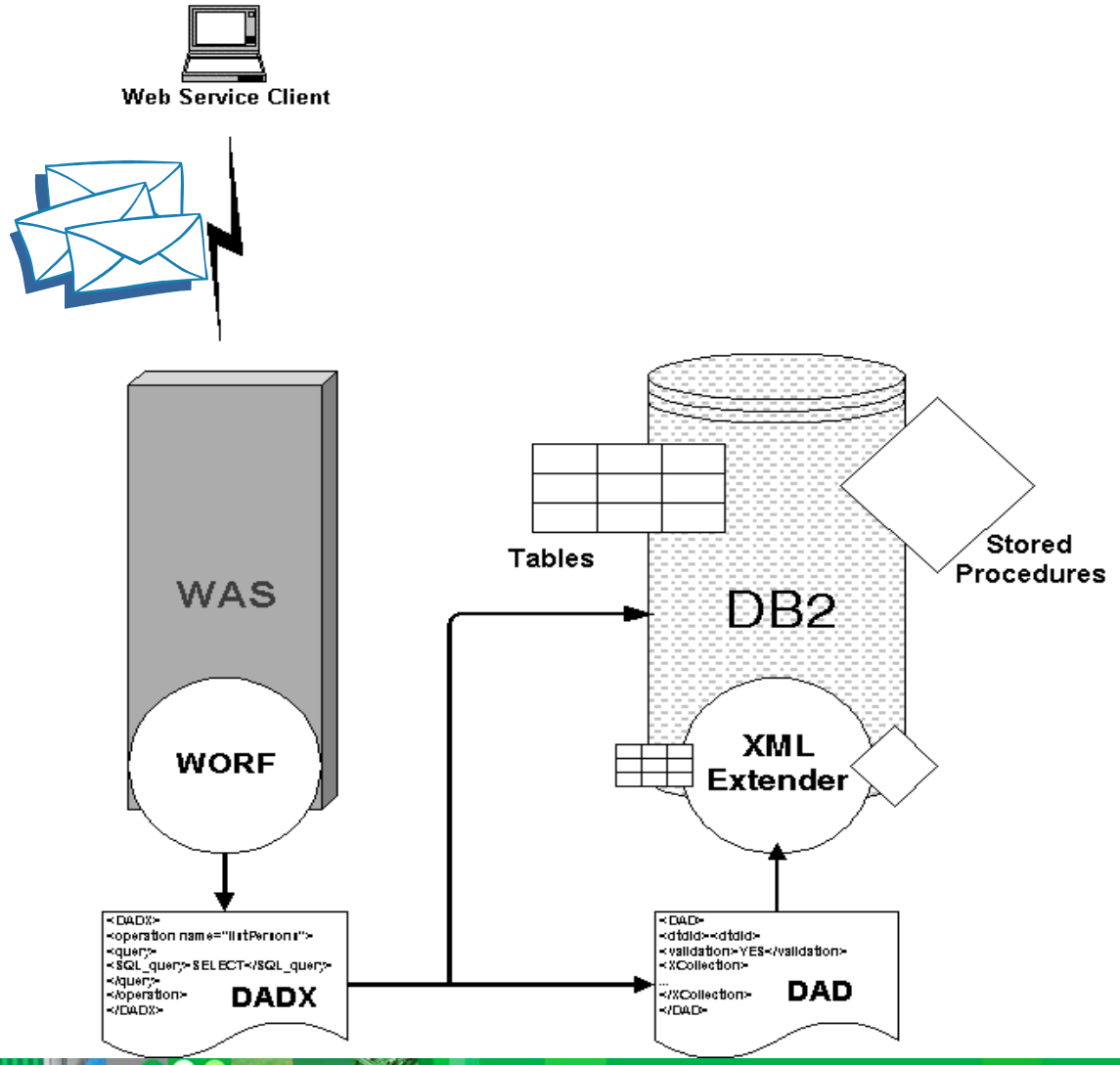
```
<DADX xmlns="http://schemas…">
  <operation name="listPersons">
    <query>
      <SQL_query>SELECT pfname, plname FROM
          db2.persons</SQL_query>
    </query>
  </operation>
  <operation name="InsertPerson">
    <call>
      <SQL_call>call db2.InsertPerson(:lastname,
          :comment) </SQL_call>
      <parameter name="lastname" type="xsd:string"/>
      <parameter name="comment" type="xsd:string"
          kind="out"/>
    </call>
  </operation>
</DADX>
```

SOAP request processing by WORF

- **loads the DADX**
- **optionally loads a DAD file**
- **replaces query parameters**
- **connects to DB2**
- **runs the SQL statement**
- **commits the database transaction**
- **formats the result into XML**
- **returns the response in a SOAP envelope**

Note: One SQL statement for each Web Service invocation

SOAP request processing by WORF



Possible DADX operations / DB2 manipulations

SQL based queries

- SELECT
- INSERT
- UPDATE
- DELETE
- CALL

XML based queries

- retrieveXML
- storeXML

Agenda

- **DB2 UDB v8.1:**
  - **Storing Data**
  - **Integration with Business Logic**
  - **Integration with XML**
- **DB2 Web Services:**
  - **SOAP service communication with DB2: WORF**
  - **Web Service specification: DADX document**
  - ***SQL based queries***
  - **XML based queries**
- **Calling DB2 Web Services:**
  - **Changes to Client Programs**

INSERT / UPDATE / DELETE operations

- <update> - operation
- Input parameters indicated with colon
- Parameter description
- Return value: 0 or 1

```
<DADX xmlns="http://schemas.ibm.com/db2/dxx/dadx"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <operation name="newPerson">
      <update>
         <SQL_update>insert into persons
         (person_sofi, person_lastname, person_firstname)
         values (:sofi, :lastname, :firstname)
         </SQL_update>
         <parameter name="sofi" type="xsd:string"/>
         <parameter name="lastname" type="xsd:string"/>
         <parameter name="firstname" type="xsd:string"/>
      </update>
   </operation>
</DADX>
```

SQL based queries

SELECT operations

SQL based queries

- **<query> - operation**
- **Input parameters indicated with colon**
- **Parameter description**
- **Return: result table in default XML-tagging (column = element name)**


```
<operation name="listPersonTotals">
   <wsdl:documentation>A DADX-query. Lists the persons
   with more than "nrloans" loans in company with id
   "compid"</wsdl:documentation>
   <query>
      <SQL_query>Select person_sofi, person_lastname,
         person_firstname, count(*) as number_of_loans,
         sum(loan_nrmonths * loan_monthlypayment) as
         amount_to_pay
      from persons, loans
      where loan_person_sofi = person_sofi
        and loan_company_id = :compid
      group by person_sofi, person_lastname,
                        person_firstname
      having count(*) >= :nrloans
      </SQL_query>
      <parameter name="compid" type="xsd:short"/>
      <parameter name="nrloans" type="xsd:long"/>
   </query>
</operation>
```

CALL operations

SQL based queries

- <call> - operation
- Input / Output parameters indicated with colon
- Parameter description
- Return:
  - Output parameters
  - Result set(s) in default XML-tagging
- Result set description

CALL operation without result set

```
<operation name="updatePaybehaviour">
   <wsdl:documentation>A DADX-call.</wsdl:documentation>
   <call>
         <SQL_call>call javaUpdatePaybehaviour (:newPaybehaviour, :firstname,
              :lastname)
         </SQL_call>
         <parameter name="newPaybehaviour" type="xsd:string"/>
         <parameter name="firstname" type="xsd:string"/>
         <parameter name="lastname" type="xsd:string"/>
   </call>
</operation>
```

SQL based queries

CALL operation with result set

SQL based queries

```
<DADX xmlns="http://schemas.ibm.com/db2/dxx/dadx" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
   <result_set_metadata name="listPersonInfo" rowName="personInfoRow">
         <column name="PERSON_SOFI" type="CHAR" as="SofiNr" nullable="false"/>
         <column name="PERSON_LASTNAME" type="CHAR" as="LastName" nullable="true"/>
         <column name="PERSON_FIRSTNAME" type="CHAR" as="FirstName" nullable="true"/>
   </result_set_metadata>
   <operation name="monthlyUpdate">
         <call>
               <SQL_call>call SqlplMonthlyUpdate (:compid, :nrUpdates, :comment) </SQL_call>
               <parameter name="compid" type="xsd:short"/>
               <parameter name="nrUpdates" type="xsd:long" kind="out"/>
               <parameter name="comment" type="xsd:string" kind="out"/>
               <result_set name="badPayers" metadata="listPersonInfo"/>
         </call>
   </operation>
</DADX>
```

Agenda

- **DB2 UDB v8.1:**
  - **Storing Data**
  - **Integration with Business Logic**
  - **Integration with XML**
- **DB2 Web Services:**
  - **SOAP service communication with DB2: WORF**
  - **Web Service specification: DADX document**
  - **SQL based queries**
  - ***XML based queries***
- **Calling DB2 Web Services:**
  - **Changes to Client Programs**

XML based queries

Store XML operations

- **<storeXML> - operation**
- **DAD needed for XML decomposition**
- **Name Space Table needed**
  - *<name-space-table-name>*.nst
  - indicate name in group.properties file

```
<operation name="newCompanyRegistration">
   <wsdl:documentation>storeXML</wsdl:documentation>
   <storeXML>
         <DAD_ref>companyRegistration.dad</DAD_ref>
   </storeXML>
</operation>
```

XML based queries

Retrieve XML operations

- **<retrieveXML>**
- **DAD needed for XML composition**
- **SQL override**
- **XML override**
- **Input parameters indicated with colon**
- **Return: XML document in DAD-defined tagging**

XML based queries

Retrieve XML operations

```
<operation name="listPersonLoans">
    <wsdl:documentation>retrieveXML.</wsdl:documentation>
    <retrieveXML>
        <DAD_ref>badPayers.dad</DAD_ref>
        <SQL_override>select person_sofi, person_lastname, person_firstname, loan_no,
            loan_nrmonths * loan_monthlypayment as atp
            from db2.persons, db2.loans
            where person_lastname = :lastname
            and person_firstname = :firstname
            and loan_person_sofi = person_sofi
            and loan_company_id = :compid
            order by person_sofi, loan_no
        </SQL_override>
        <parameter name="lastname" type="xsd:string"/>
        <parameter name="firstname" type="xsd:string"/>
        <parameter name="compid" type="xsd:short"/>
    </retrieveXML>
</operation>
```


IBM Software Group
Calling DB2 Web Services
DB2 Information Management Software
e business on demand software

Agenda

- **DB2 UDB v8.1:**
  - **Storing Data**
  - **Integration with Business Logic**
  - **Integration with XML**
- **DB2 Web Services:**
  - **SOAP service communication with DB2: WORF**
  - **Web Service specification: DADX document**
  - **SQL based queries**
  - **XML based queries**
- **Calling DB2 Web Services:**
  - ***Changes to Client Programs***

Question:

"If we compare a DB2 web service with other web services. Do we have to make any changes to Client Programs?"

Answer:

“No!”