

DB2, JavaBeans

Submit/Retrieve to DB2 - Xpages Button (Part 2)

Xpages

System Requirements:

Download DB2 Express-C

<http://www-01.ibm.com/software/data/db2/express/download.html>

Download Domino Designer 8.5.3 Environment (DDE)

<http://www.ibm.com/developerworks/downloads/ls/dominodesigner/>

Introduction:

Configure CLASSPATH on Windows 7 OS for the DB2 Application, reference necessary jar files through your project Build Path, connect to DB2 using a Xpage form button load and retrieve data; see log.nsf for the results. Information that can be used to later search the current DOMINODB database dynamically via links on Xpages...

Disclaimer:

Information contained in the following is presented as is. This tutorial assumes you have basic programming knowledge. All tutorials are based on an Eclipse/Eclipse-based software. Should you need to familiarize yourself with a certain Eclipse environment, prior to continuing this tutorial, please stop now and see our Tutorials page...

Configure ClassPath/Build Path (Reference Jar Files)

Let's add ClassPath variables and load jar files from the DB2 installation directory to Lotus Notes installation directory to begin the process. You will need to configure your project's Build Path to include the specific jar files to your programs. We do recommend restarting DB2 and DDE after configuration steps have been met. See screenshots below, then Copy and paste code samples to your environment... areas of interest have been highlighted for your convenience.

CONTROL PANEL

Do these steps for Windows 7 ClassPath

1. Open Control Panel
2. Select System and Security
3. Click on System
4. Then choose Advanced System Settings option @ left
5. In Pop-Up, choose Advanced Tab
6. Find and click Environment Variables button below
7. Select and Edit CLASSPATH information in System Variables Window
8. Load jar files as shown below from the following DB2 directory
C:\PROGRA~1\IBM\SQLLIB\java\db2java.zip;C:\PROGRA~1\IBM\SQLLIB\java

Figure 1

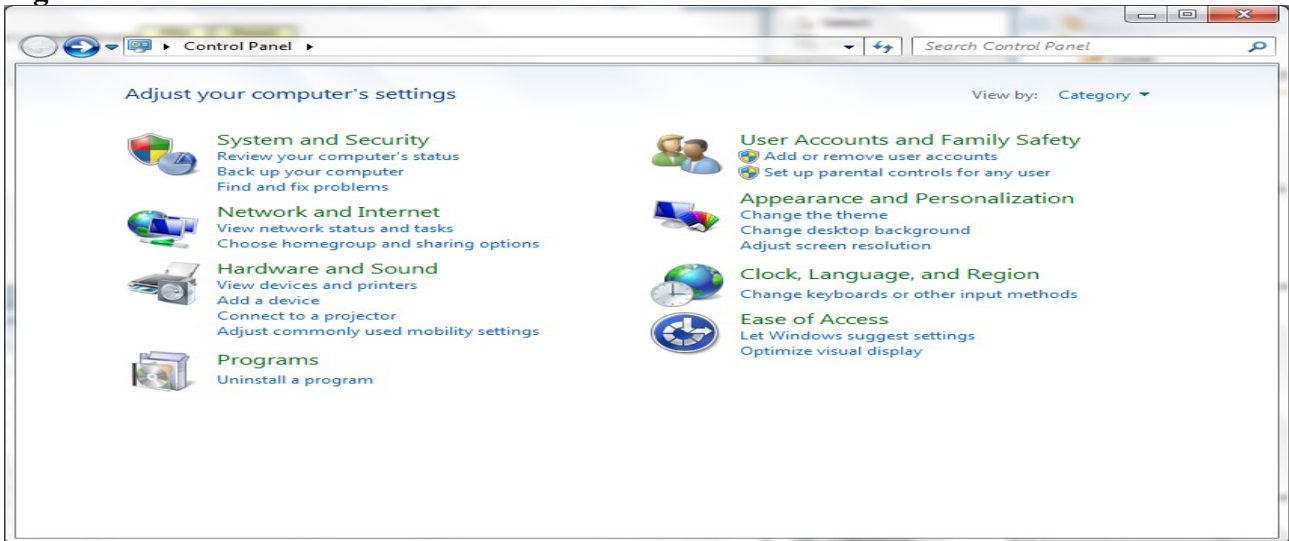


Figure 2

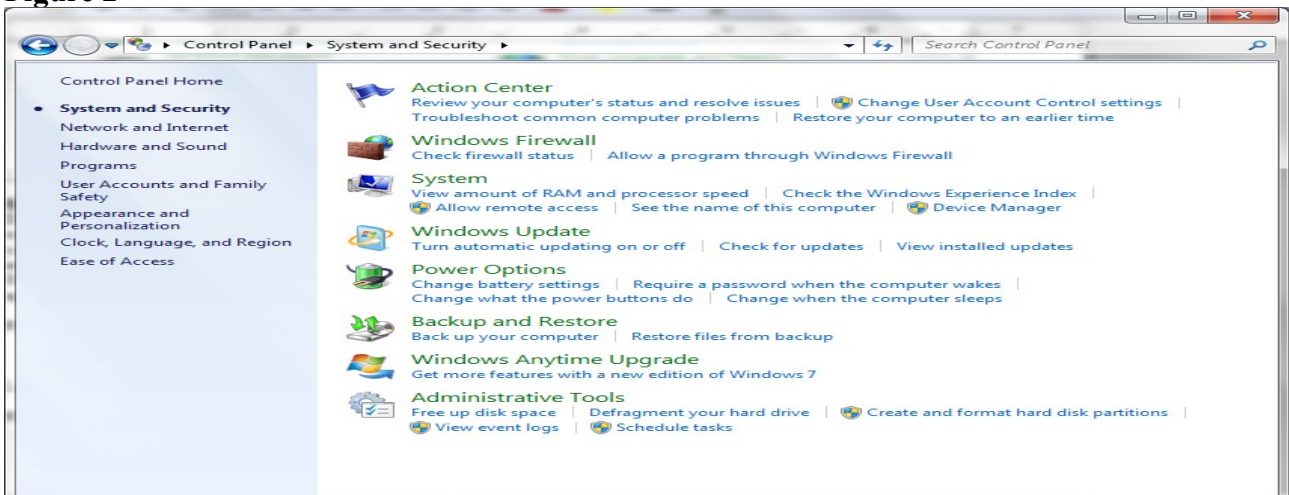


Figure 3

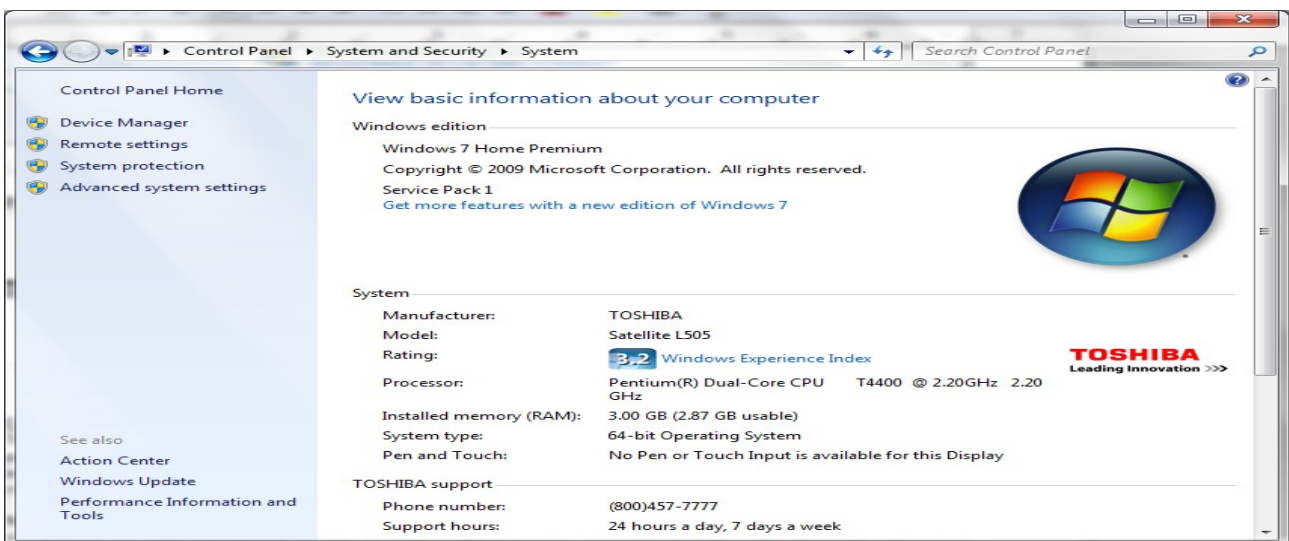


Figure 4

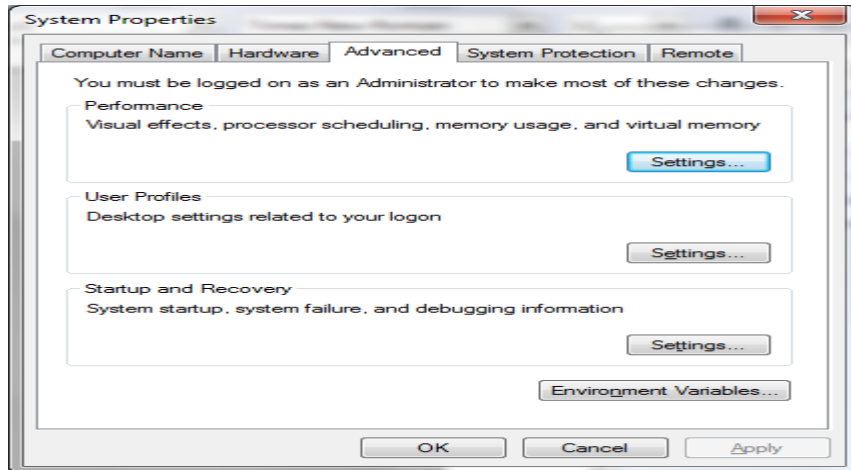


Figure 5

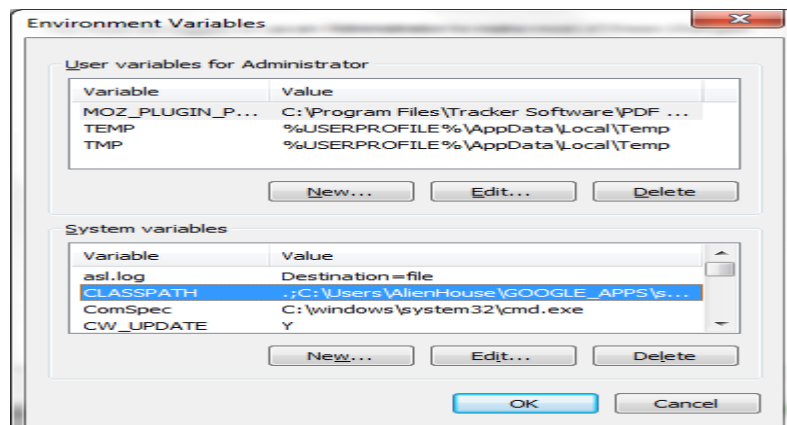
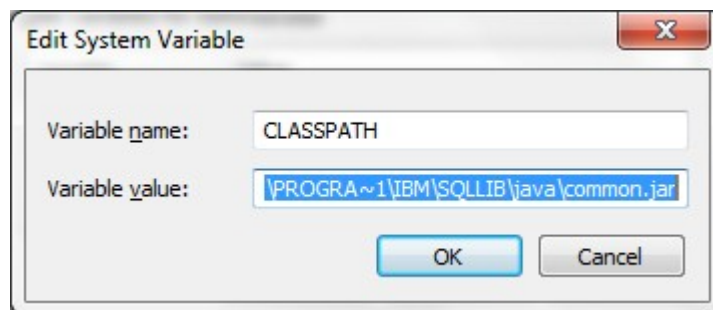


Figure 6



.;C:\PROGRA~1\IBM\SQLLIB\java\db2java.zip;C:\PROGRA~1\IBM\SQLLIB\java\db2jcc4.jar;

Figure 7

this is achieved by right-clicking WebContent/WEB-INF/YourSourceFolder, and choosing Build Path

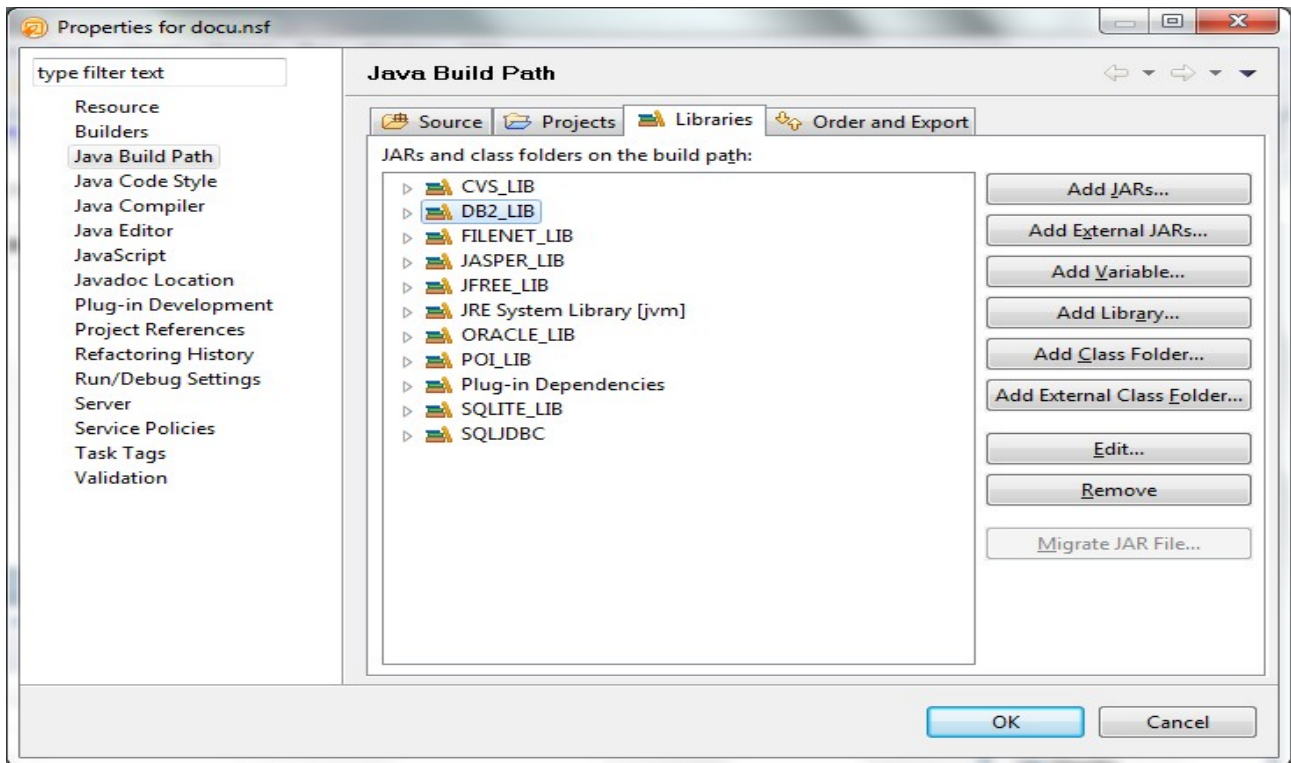


Figure 8

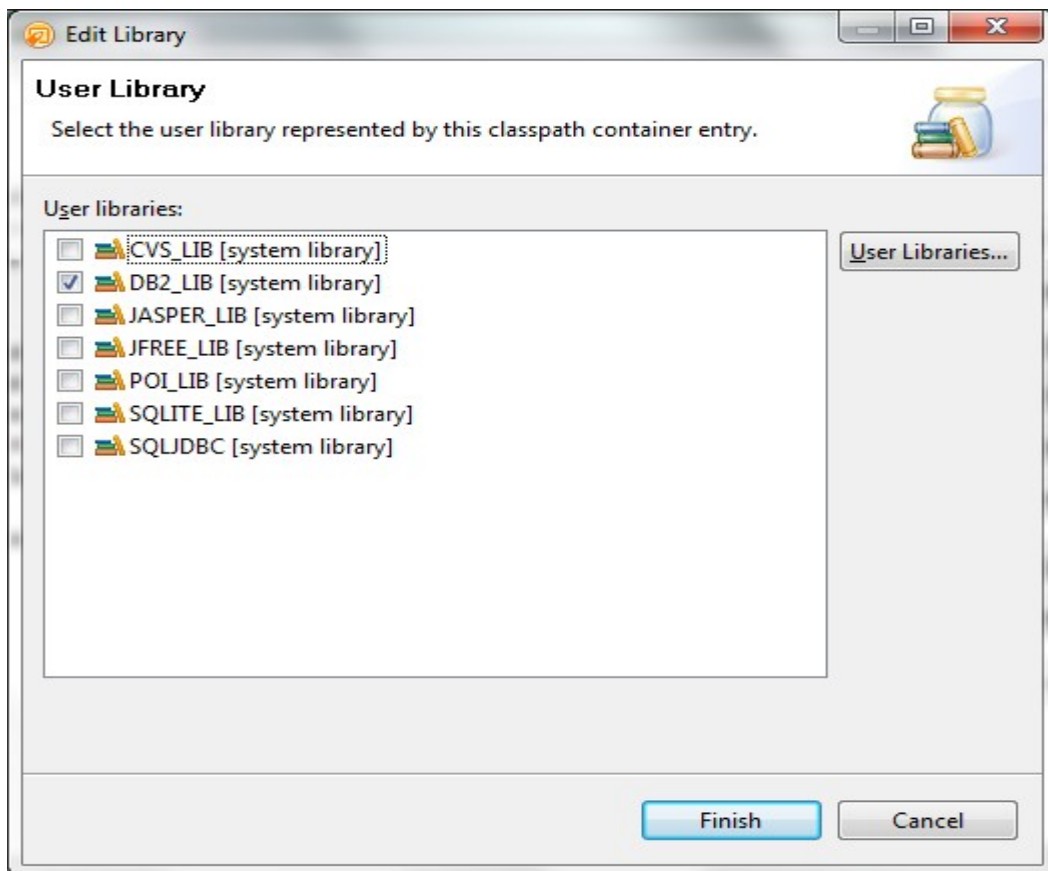
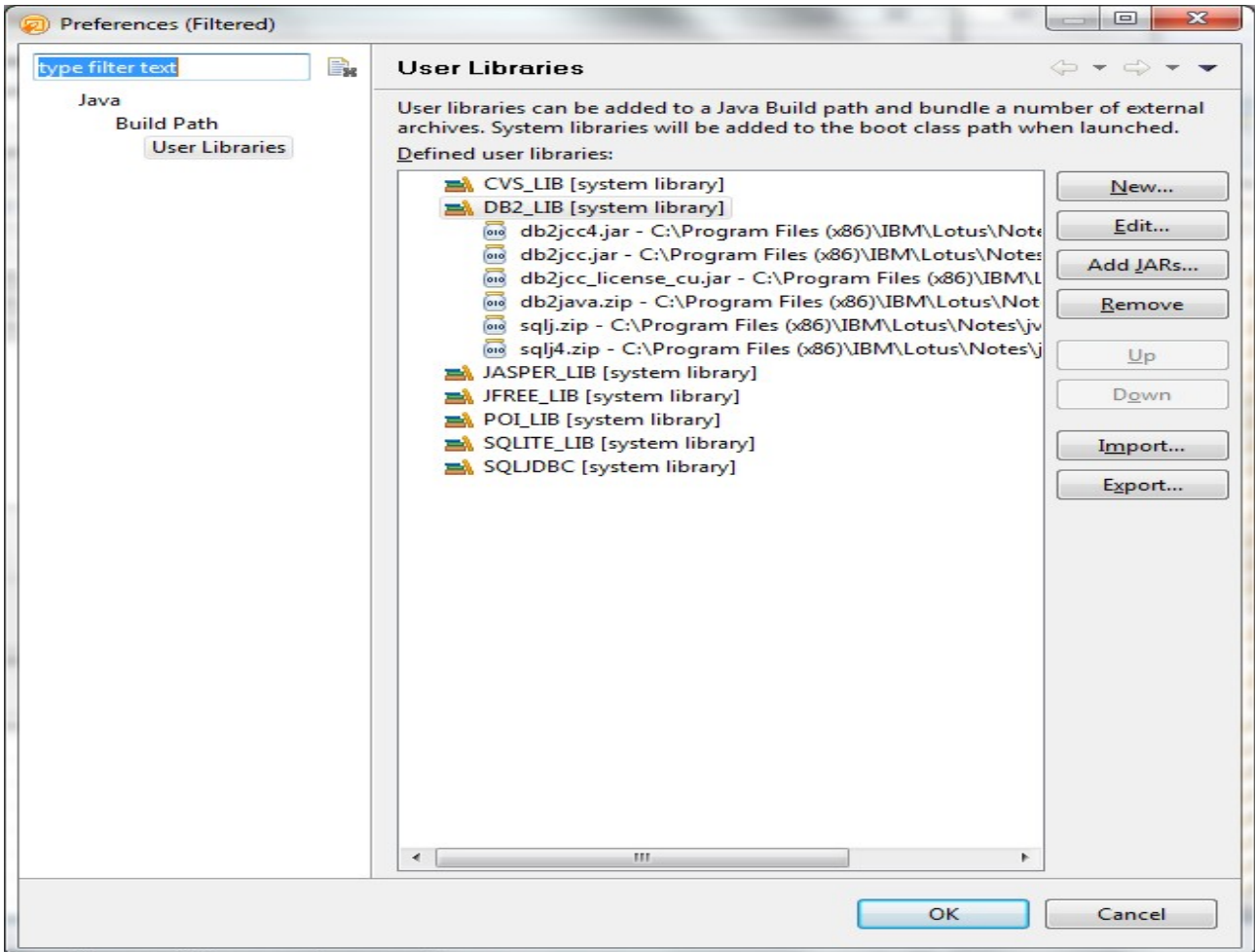
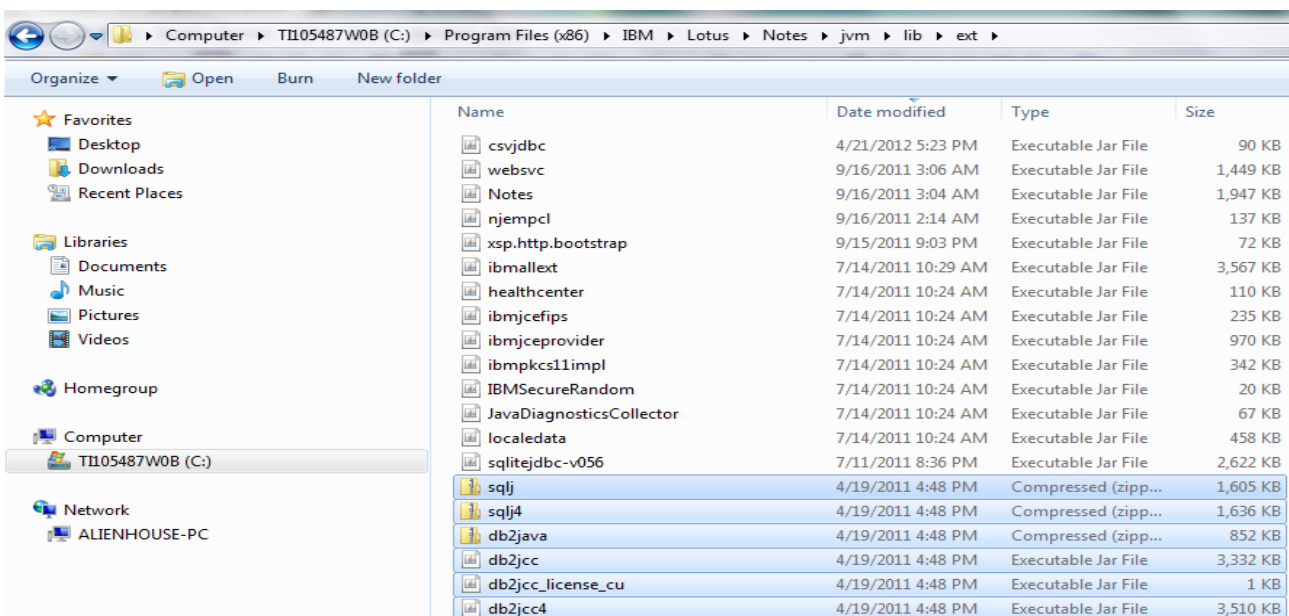


Figure 9



If you do not have these jar files, please visit this directory for DB2:
 C:\PROGRA~1\IBM\SQLLIB\java\db2java.zip;C:\PROGRA~1\IBM\SQLLIB\java then grab all jar files shown in screenshot above. Include them all in your Lotus Notes installation directory...



Result of this exercise is shown below from log.nsf...

```

10/31/2012 02:52:36 AM HTTP JVM: LoggedInCheck: afterPhase Listener class method...xprundb2
10/31/2012 02:52:36 AM HTTP JVM: LoggedInCheck: loggedIn Phase Listener session block...null
10/31/2012 02:52:36 AM HTTP JVM: Entering actual connection...
10/31/2012 02:52:36 AM HTTP JVM: DB2Connector Connection Started...
10/31/2012 02:52:39 AM HTTP JVM: DB2Connector Class obtained...
10/31/2012 02:52:42 AM HTTP JVM: DB2Connector Connection forwarded...
10/31/2012 02:52:42 AM HTTP JVM: Begin statement...
10/31/2012 02:52:42 AM HTTP JVM: Entering query...
10/31/2012 02:52:42 AM HTTP JVM: Values added...
10/31/2012 02:52:42 AM HTTP JVM: Submitting to console...
10/31/2012 02:52:42 AM HTTP JVM: ID |AllegationCode|Office |Investigator|DateOpen |Age |AllegationNumber|Licensee |Subject |Region |
10/31/2012 02:52:42 AM HTTP JVM: XYZDB2007 |XYZDB2NAME10007 |New York |nyusername10xyz |10/28/2012 |42 |XYZDB2007USERNAME10 |007123456789XYZ |Dököll Solutions, Inc. |null
10/31/2012 02:52:42 AM HTTP JVM: ID |AllegationCode|Office |Investigator|DateOpen |Age |AllegationNumber|Licensee |Subject |Region |
10/31/2012 02:52:42 AM HTTP JVM: XYZ008ABC |XYZ008ABCALB |Albany |XYZ008CBCNAME11 |01/01/1971 |50 |2012.05.18.12.30.AM |10029870899 |Good Buddy |Capital District
10/31/2012 02:52:42 AM HTTP JVM: ID |AllegationCode|Office |Investigator|DateOpen |Age |AllegationNumber|Licensee |Subject |Region |
10/31/2012 02:52:42 AM HTTP JVM: XYZ009CBC |XYZ009CBCALB |Albany |XYZ008CBCNAME12 |10/01/1990 |25 |2010.11.11.10.PM |22224444811 |SmashBook, Inc. |Capital District
10/31/2012 02:52:42 AM HTTP JVM: ID |AllegationCode|Office |Investigator|DateOpen |Age |AllegationNumber|Licensee |Subject |Region |
10/31/2012 02:52:42 AM HTTP JVM: XYZ000DBC |XYZ000DBCALB |Albany |XYZ000DBCNAME13 |10/01/2003 |9 |2010.11.11.10.PM |00004444811 |BookWorm.com |Capital District
10/31/2012 02:52:42 AM HTTP JVM: ID |AllegationCode|Office |Investigator|DateOpen |Age |AllegationNumber|Licensee |Subject |Region |
10/31/2012 02:52:42 AM HTTP JVM: XYZ001EBC |XYZ001EBCBOS |Boston |XYZ001EBCNAME13 |01/01/1943 |70 |2000.01.11.11.10.PM |33335555899 |GuntherBerg, Inc. |Commonwealth
10/31/2012 02:52:42 AM HTTP JVM: ID |AllegationCode|Office |Investigator|DateOpen |Age |AllegationNumber|Licensee |Subject |Region |
10/31/2012 02:52:42 AM HTTP JVM: XYZ002FBC |XYZ002FBCCAN |Montréal |XYZ002FBCNAME14 |01/01/1980 |39 |2010.12.19.1.10.AM |99994444811 |SmashBook, Inc. |Oh Canada
10/31/2012 02:52:42 AM HTTP JVM: ID |AllegationCode|Office |Investigator|DateOpen |Age |AllegationNumber|Licensee |Subject |Region |
10/31/2012 02:52:42 AM HTTP JVM: XYZ002FBC |XYZ002FBCCHI |Chicago |XYZ002FBCNAME13 |10/01/2011 |20 |2011.10.11.11.10.AM |88885555199 |HamBerg, Inc. |Windy City
10/31/2012 02:52:42 AM HTTP JVM: ID |AllegationCode|Office |Investigator|DateOpen |Age |AllegationNumber|Licensee |Subject |Region |
10/31/2012 02:52:42 AM HTTP JVM: XYZ002FBC |XYZ002FBCCHI |Chicago |XYZ002FBCNAME14 |01/01/2012 |49 |2011.10.19.1.10.AM |77778888111 |GreenRoofTops, Inc. |Windy City

```

Establish a connection to DOMINODB (DB2)

See below two Java class files, identical to previous work submitted for SQLite Database, nothing fancy. Copy and paste to your environment, areas of interest have been highlighted for your convenience.

DB2Connector;

```

package com.dokoll.solutions.inc.Utils;
/**
 * Copyright 2012 Dököll Solutions, Inc.
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 *
 * @AppName: Docu.nsf
 * Program: DB2Connector.java
 * Created from Copy: 2012.10.31.12.12.AM
 */
/**
 *
 * @author Dököll Solutions, Inc.
 * @version 2012.10.31.12.12.AM
 */
import java.sql.*;

//load program
public class DB2Connector {
    //get a connection
    public static Connection getConnection() {
        System.out.println("DB2Connector Connection Started...");
        //declare and initialize
        Connection connection = null;
        //entering try catch
        try {

```

```

        //load driver class
        Class.forName("COM.ibm.db2.jdbc.app.DB2Driver").newInstance();
        System.out.println("DB2Connector Class obtained...");
        //connect to DB2 DOMINODB database
        connection =
DriverManager.getConnection("jdbc:db2://localhost:50000/DOMINODB","db2admin","yo
urpassport");
        //...
        System.out.println("DB2Connector Connection forwarded...");
    } catch (Exception e) {
    e.printStackTrace();
    }
    //...
    return connection;
}

//      public static void main(String[] args) throws Exception {
//          //Use this block to test as Java program
//          getConnection();
//          System.out.println("DB2Connector Program Successful...");
//      }
//
//
}

```

Submit and retrieve DB2 Data from DOMINODB

You have a connector class and it looks like it will work, Now what? Well, submit static data to DOMINODB and retrieve @ once. Copy and paste to your environment, areas of interest have been highlighted for your convenience.

DB2AddDataBean;

```

/**
 * Created: 2012.10.28.1.40.PM
 * DB2AddDataBean
 * DB2 Data Loading Bean
 */
package com.dokoll.solutions.inc.db2.test;
/**
 * @author Dököll Solutions, Inc.
 * @version 2012.10.28.1.40.PM
 *
 */
import java.sql.*;

import com.dokoll.solutions.inc.Utills.DB2Connector;

public class DB2AddDataBean {

    public DB2AddDataBean() {
        //...
    }

    //button code...
    public void doGetDB2Data() throws Exception {

        System.out.println("Entering actual connection...");
        Connection connection = DB2Connector.getConnection();
        System.out.println("Begin statement...");
        Statement stat = connection.createStatement();
        System.out.println("Entering query...");
        PreparedStatement prep = connection.prepareStatement("insert into

```

```

DB2ADMIN.USER_DETAILS values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");

//      //Added Data for report
//      //2012.10.28.9.03.PM
prep.setString(1, "XYZ002FBC");
prep.setString(2, "username16");
prep.setString(3, "XYZ002FBCCHI");
prep.setString(4, "Chcago");
prep.setString(5, "XYZ002FBCNAME13");
prep.setString(6, "10/01/2011");
prep.setString(7, "20");
prep.setString(8, "HamBerg, Inc.");
prep.setString(9, "88885555199");
prep.setString(10, "2011.10.11.11.10.AM");
prep.setString(11, "Windy City");
prep.addBatch();
prep.setString(1, "XYZ002FBC");
prep.setString(2, "username15");
prep.setString(3, "XYZ002FBCCHI");
prep.setString(4, "Chicago");
prep.setString(5, "XYZ002FBCNAME14");
prep.setString(6, "01/01/2012");
prep.setString(7, "49");
prep.setString(8, "GreenRoofTops, Inc.");
prep.setString(9, "77778888111");
prep.setString(10, "2011.10.19.1.10.AM");
prep.setString(11, "Windy City");
prep.addBatch();

System.out.println("Values added...");
connection.setAutoCommit(false);
prep.executeBatch();
connection.setAutoCommit(true);
ResultSet rs = stat.executeQuery("select * from DB2ADMIN.USER_DETAILS");
System.out.println("Submitting to console...");
while (rs.next()) {
    System.out.println("ID      |"+"AllegationCode|"+"Office
|"+"Investigator|"+"DateOpen      |"+"Age      |"+"AllegationNumber|"+"Licensee
|"+"Subject      |"+"Region      |");
    System.out.println(rs.getString("ID")+
|"+"rs.getString("AllegationCode")+      |"+"rs.getString("Office")+      |"+"
rs.getString("Investigator")+      |"+" rs.getString("DateOpen")+      |"+"
rs.getString("Age")+      |"+" rs.getString("AllegationNumber")+      |"+"
rs.getString("Licensee")+      |"+" rs.getString("Subject")+      |"+"
rs.getString("Region"));
}
rs.close();
connection.close();
}
}

```

Submit and retrieve DB2 Data with Xpages

You have a connector class and a JavaBean rigged to submit and retrieve data from DB2 database (DOMINODB), but you need to run code and perform tasks... Plug your JavaBean's method into the following Xpages form button, and check log.nsf database for the results. Copy and paste to your environment, areas of interest have been highlighted for your convenience.

xprundb2.xsp;


```

<?xml version="1.0" encoding="UTF-8"?>
<xp:view xmlns:xp="http://www.ibm.com/xsp/core">

    <html>
        <head>
            <title>DB2 Example</title>
        </head>
        <body>
            <xp:form>
                <table>

                    <tr>
                        <td style="font-weight:bold">Add/Retrieve DB2
Data</td>
                        <td>

                            <xp:button value="Submit"
                                id="button1">
                                <xp:eventHandler
                                event="onclick"
                                submit="true"
                                refreshMode="complete" immediate="false"
                                id="eventHandler1">

                                    <xp:this.action><![
[CDATA[#{javascript:DB2AddDataBean.doGetDB2Data()}}]></xp:this.acti
on>
                                    </xp:eventHandler>
                                </xp:button>

                            </td>
                        </tr>
                    </table>
                </xp:form>
            </body>
        </html>
    </xp:view>

```

Conclusion:

You can now use an Xpages form button to connect to DB2 database to add and retrieve data. Build on this one to retrieve records into a dataTable, perhaps submit values as documents in Lotus Notes, search back-end via Xpages.

Questions, comments, please post a brief message on our [Contact](#) form on the main site.

Thank you for coming...