APPENDIX A

Sample Java Program

Overview

This appendix contains the files that you will need to create a Java program that uses Java Database Connectivity (JDBC) to connect to an SQL database, creates and executes an SQL statement, and then retrieves the results. The sample files are described in Table A-1.

Table A-1 Files in Sample Java Program

File	Description
Example A-1 JTest.java	Main program that allows you to connect to a database, create and execute a statement, and then retrieve the results.
Example A-2 ConnectionPoolManager.java	Definition of class ConnectionPoolManager that allows you to connect to the database.
Example A-3 Switch.java	Definition of a class switch that allows you to create a statement.
Example A-4 db.properties	Database-specific definitions. You must set these values to reflect the JDBC driver used, and the URL, user, and password of the database accessed.

For an introduction to JDBC in Wikipedia and the JDBC API Guide follow these links:

- http://en.wikipedia.org/wiki/JDBC
- http://java.sun.com/j2se/1.5.0/docs/guide/jdbc/

After you install the JDBC API, you can find the data types for use in statements in the install directory file constant-values.html.

Example A-1 JTest.java

```
import java.sql.*;
import java.io.*;
import java.util.*;

public class JTest {

    public static void main(String args[]) {

        Connection con;
        Statement stmt;
        SQLWarning warning = null;
```

```
boolean ret = false;
ResultSet results = null;
int updateCount = 0;
Properties prop = new Properties();    // contains contents of propertyFileName
String    drivername = null;
String    drivername2 = null;
String    name;
String    URL;
String    user;
String    password;

System.out.println("Java Test");

try {
    // Get the Connection Props.
    InputStream is = new BufferedInputStream(new FileInputStream(new File("db.properties")));
     prop.load(is);
     is.close();

     name = prop.getProperty("DS.name");
     URL =  prop.getProperty("DS.url");
     user = prop.getProperty("DS.user");
     password = prop.getProperty("DS.password");
     drivername = prop.getProperty("DS.driver");
     drivername2 = prop.getProperty("DS.driver2");

     System.out.println(name);
     System.out.println(URL);
     System.out.println(drivername);

    // REGISTER DRIVER
    Driver d = (Driver)Class.forName(drivername).newInstance();

    if ( drivername2 != null) {
        Driver d2 = (Driver)Class.forName(drivername2).newInstance();
        System.out.println(drivername2);
    }

    // GET CONNECTION
    con = DriverManager.getConnection(URL,user,password);

    // GET CONNECTION WARNINGS
    try {
        warning = con.getWarnings();

        if (warning == null){
        System.out.println("No Warnings");
        //return;
        }

        while (warning != null) {
        System.out.println("Warning: "+warning);
        warning = warning.getNextWarning();
        }

    } catch (Exception e){
        System.out.println(e);
    }

    // CREATE STATEMENT
    stmt = con.createStatement();

    // EXECUTE SQL
    ret = stmt.execute("select * from EMP");
```

```
            if (ret == true){
                results = stmt.getResultSet();
            }
            else{
                updateCount = stmt.getUpdateCount();
            }

            // GET ALL RESULTS
            StringBuffer buf = new StringBuffer();
            try {
                ResultSetMetaData rsmd = results.getMetaData();
                int numCols = rsmd.getColumnCount();
                int i, rowcount = 0;

               // get column header info
                for (i=1; i <= numCols; i++){
                    if (i > 1) buf.append(",");
                    buf.append(rsmd.getColumnLabel(i));
                }
                buf.append("\n");

                // break it off at 100 rows max
                while (results.next() && rowcount < 100){
                    // Loop through each column, getting the column
                    // data and displaying

                    for (i=1; i <= numCols; i++) {
                        if (i > 1) buf.append(",");
                        buf.append(results.getString(i));
                    }
                    buf.append("\n");
                    rowcount++;
                }
                results.close();
                System.out.println(buf);
            } catch (Exception e) {
                System.out.println(e);
                return;
            }


        } catch (Exception e) {
            System.out.println(e);
        }


    }

}

/*
DISCLAIMER: The sample code is not supported under any DataDirect Technologies support program or service.
The sample code is provided on an "AS IS" basis.  DataDirect Technologies makes no warranties, express or
implied, and disclaims all implied warranties including, without limitation, the implied warranties of
merchantability or of fitness for a particular purpose. The entire risk arising out of the use or performance
of the sample code is borne by the user. In no event shall DataDirect Technologies, its employees, or anyone
else involved in the creation, production, or delivery of the code be liable for any damages whatsoever
(including, without limitation, damages for loss of business profits, business interruption, loss of business
information, or other pecuniary loss) arising out of the use of or inability to use the sample code, even if
DataDirect Technologies has been advised of the possibility of such damages.
*/
```

Example A-2 ConnectionPoolManager.java

```
public class ConnectionPoolManager {
    //specify which database schema will be used
    private final static String Alias = "dbname";
    //specify the database vendor libary that implements JDBC api
    private final static String DbDriver = "org.hsqldb.jdbcDriver";
    //other attributes for connecting database
    private static String DbUrl = "jdbc:hsqldb:hsql://localhost";
    private static String DbUser = "db_username";
    private static String DbPass = "_db_password";
    private final static String[] DbFiles = {
        "dbname.data",
        "dbname.script",
        "dbname.backup",
        "dbname.properties",
        "dbname.log"
    };

   public static ConnectionPoolManager Instance;

   public static ConnectionPoolManager getInstance(){
 if(Instance == null){
             Class.forName(_DbDriver).newInstance();
             Instance= new ConnectionPoolManager(300);
                 Instance.addAlias(Alias, DbDriver, DbUrl, DbUser, DbPass, 6,  300, 10, 10);
            }
     return Instance;
  }

   public Connection getConnection() throws SQLException {
        return DriverManager.getConnection("jdbc:bitmechanic:pool:"+ Alias, null, null);
    }

    public static void returnConnection(Connection conn) throws SQLException {
        conn.close();
    }
}
```

Example A-3 Switch.java

```
public final class Switch  {
  final static String QuerySQLByFabricID =
        "select id, wwn, ip_address, is_mds, type, is_managable, non_mds_model, sys_name, sys_contact,
sys_location, sys_uptime, active_sup_slot, conn_unit_status, standby_sup_state, feature_flag,
is_license_violation, version, is_present, serial_number, unmanagable_cause, last_scan_time, num_ports,
is_trap_registered, is_syslog_registered, standby_sup_slot, module_index_offset from switch where
fabric_id=?";

public static ArrayList loadFromDB(long fabricId)
        throws SQLException {
        Connection con = ConnectionPoolManager.getInstance().getConnection();
        PreparedStatement stat = con.prepareStatement(QuerySQLByFabricID);
        ResultSet rs = null;

        try {
            stat.setLong(1, fabricId);
            rs = stat.executeQuery();
            ArrayList al = new ArrayList();
            //parsing result set and put items to the list
            //....
            //....
```

```
            rs.close();
            return al;
        } catch (SQLException ex) {
            return null;
        }
        finally {
            if (rs != null) {
                rs.close();
            }
            ConnectionManager.getInstance().returnConnection(con);
        }
    }
  }
```

Example A-4 db.properties

```
DS.driver=com.ddtek.jdbc.oracle.OracleDriver
DS.name=ddtek
DS.url=jdbc:datadirect:oracle://servername:1521;SID=ORASID
DS.user=uid
DS.password=pwd

//DS.driver=com.ddtek.jdbc.sequelink.SequeLinkDriver
//DS.name=ddtek
//DS.url=jdbc:sequelink://servername:19996
//DS.user=uid
//DS.password=pwd

//DS.driver=com.ddtek.jdbcspy.SpyDriver
//DS.driver2=com.ddtek.jdbc.oracle.OracleDriver
//DS.name=ddtek
//DS.url=jdbc:spy:{jdbc:datadirect:oracle://servername:1521;SID=ORASID;user=uid;password=pwd};log=(file)C:\\t
emp\\spy.log
//DS.user=scott
//DS.password=tiger
```