

教你建立简单 JDBC 程序

江苏 无锡 缪小东

本文章主要使用 sun 提供的 JDBC-ODBC 驱动，建立简单的数据库应用。为了大家都能运行以下程序，我使用 Access 作为后台数据库（其它使用非 JDBC-ODBC 桥的程序比这个简单——没有这么复杂的预备工作，但是建立应用程序的过程基本一致）。

建立 JDBC 应用程序一般有 7 个步骤：

1. 加载 JDBC 驱动；
2. 定义连接的 URL；
3. 建立连接；
4. 创建 statement 对象；
5. 执行查询或更新操作；
6. 处理结果；
7. 关闭连接。

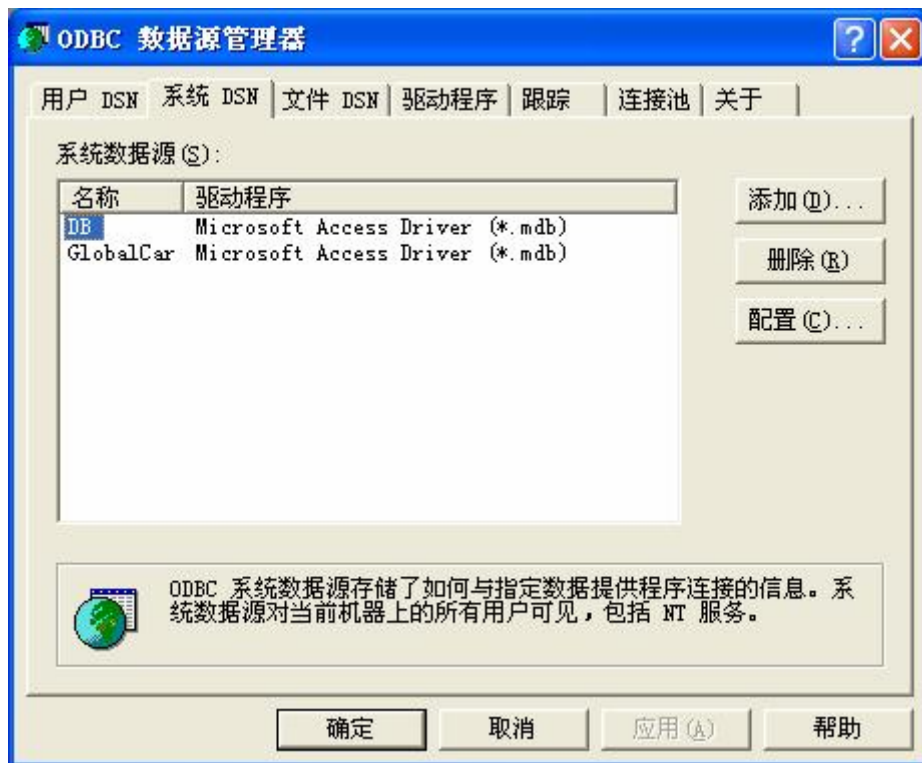
预备工作——建立 JDBC-ODBC 的数据源（Access）

建立数据源对初学者来说可能比较复杂！下面是操作的详细过程！

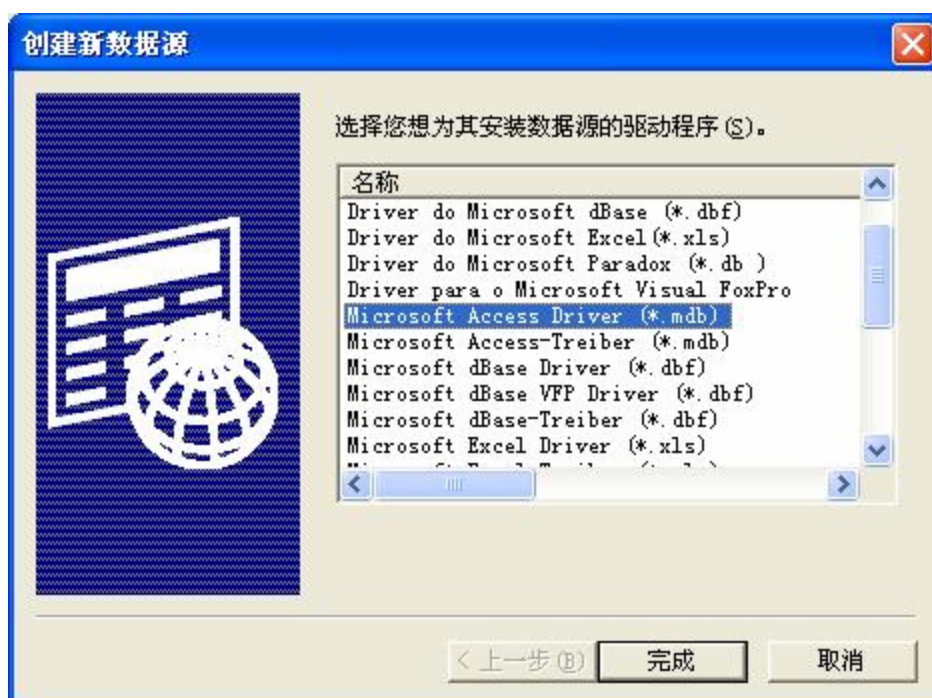
1. 打开你的开始菜单。
2. 双击控制面板，看到“管理工具”图标了吧！
3. 双击管理工具，有好多图标哦！看到“数据源(ODBC)”了吧！



4. 双击数据源，接着可以看到“ODBC 数据源管理器的窗口”



5. 下面的任务就是添加 Access 数据源了。点击上面的“系统 DNS”选项卡，然后点击“添加”按钮，接着跳出一个“添加数据源的窗口”（拖动看看，有什么 Access 的驱动，Excel 的驱动、CSV 格式的驱动甚至 FoxPro、SqlServer 的驱动），我们选用“Microsoft Access Driver (*.mdb)”哦！继续点击“完成”按钮就可以了！



6. 从上一步可以看到一个“ODBC Microsoft Access 安装”的窗口——见下图。红色的是我们输入的字符串，对应于 JDBC URL `jdbc:odbc:userDb` 中的 `userDb` 哦！（记住这个名字，JDBC.java 中使用它）点击蓝色的“高级(A)...”会出现“”窗口，它用来设置建立连接中 URL 的用户名和密码哦！



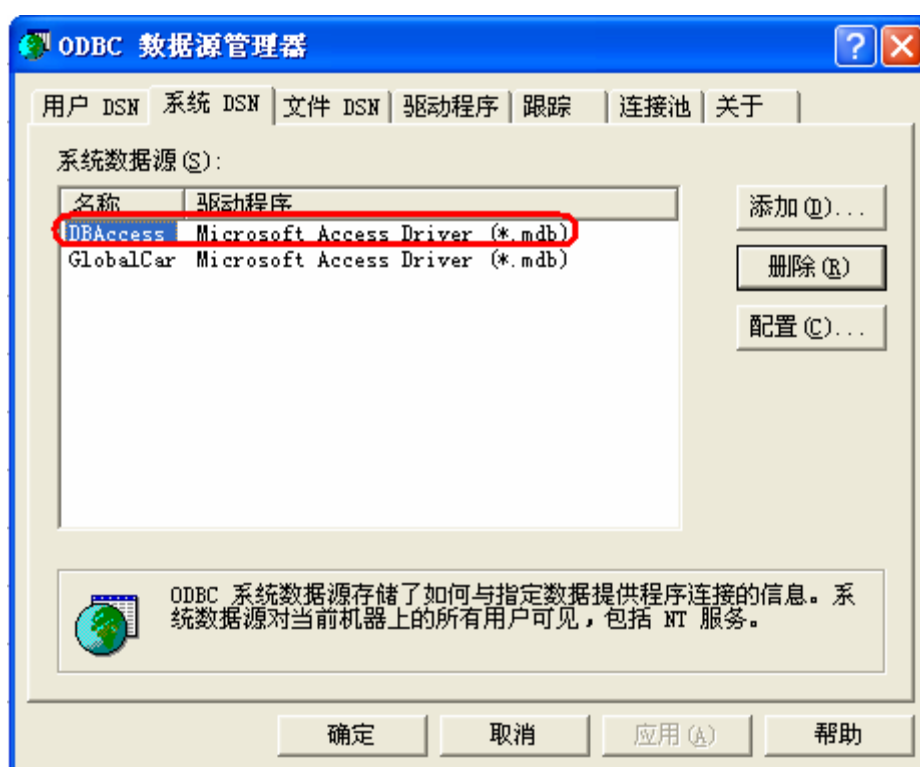
7. 输入“登陆名称”和“密码”即 JDBC 中建立连接的用户名和密码（都是 MXD！你可以自己输入你的用户名和密码，一定要记住！），输好了按“确定”就可以了！



8. 接着就是定位数据库文件了（相对很重要）。将你的 Access 数据库拷贝到指定的目录中，通过下面的“连接数据库”窗口中的“驱动器”、“目录”、“数据库名”依次定位到你的 Access 数据库文件！接着依次确定就可以了！



9. 确定后可以看到你的“ODBC 数据源管理器”中多了名称为“DBAccess”的““Microsoft Access Driver (*.mdb)”驱动程序了！



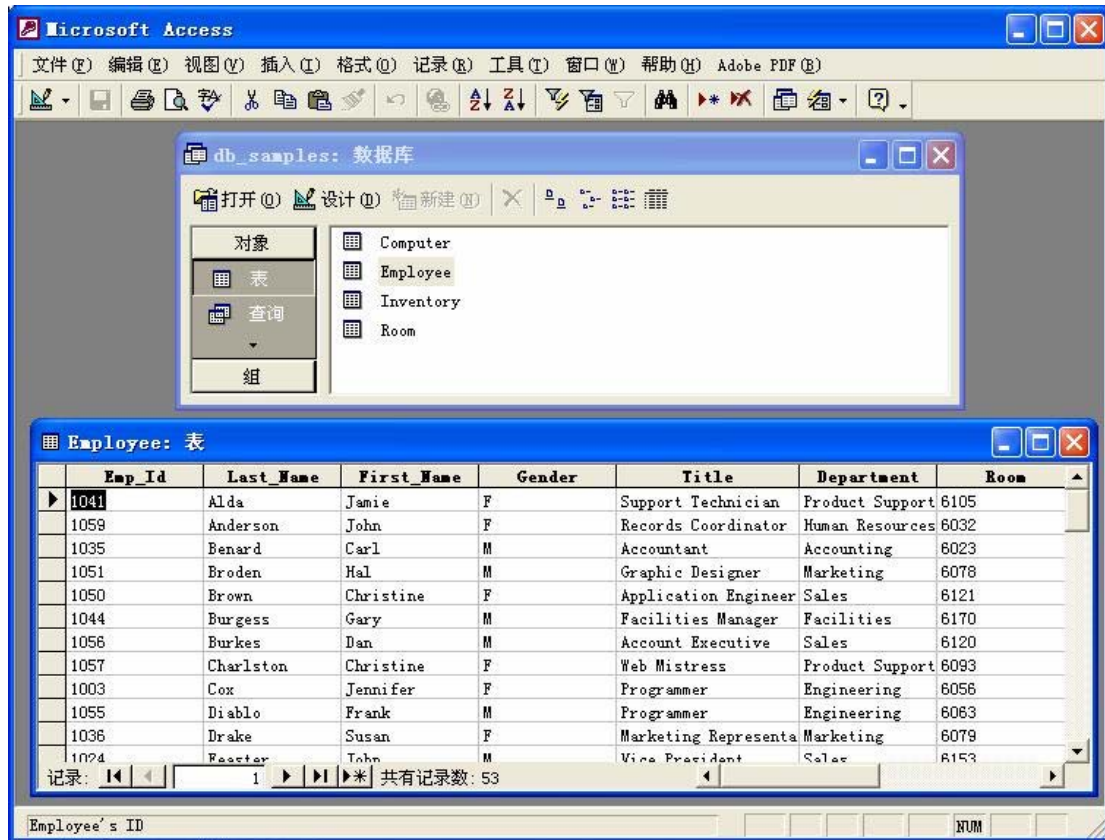
总结:

以上操作对初学者相当复杂。其实是很简单的——你只要选择 Access 驱动，设置数据库名称，设置用户名和用户密码就可以了！

没有 Access 数据库文件的朋友，可以到网上搜索搜索一个后缀为 mdb 文件就可以了！用你的 Access 软件打开它，看看数据库中的表！在你的 JDBC 程序中就是操作这些表。下面是我从我机器里搜索的名为“db_samples.mdb”的 Access 文件，它包含“Computer”“Employee”等表。因此我的 JDBC.java 程序中使用下面的语句：

```
ResultSet rs = stmt.executeQuery("select * from Employee");
```

操作 Employee 表，你必须按你 Access 数据库的结构将 Employee 换成数据库中实际存在的表名！



一、加载驱动

加载驱动指一般是将某个数据库提供商提供的相关数据库的驱动类通过 Class.forName 方法加载到内存中（后面将有专门的文章讲述该过程的细节）。Class.forName 方法接收类型为 String 的参数，该参数表示该驱动类的位置，一般地不同 JDBC 提供商的驱动，此驱动位置是不一样的！但是它们有一些共性：

1. 数据库提供商提供的以 jar 为后缀的驱动程序包必须位于系统的 classpath 下；
2. 驱动类中肯定有个类实现 java.sql.Driver 接口，我们的 Class.forName 就是加载的该类。有读者可能问我的 classpath 中没有所谓的 sun.jdbc.odbc.JdbcOdbcDriver 啊！有的，只是你没有发现罢了！sun 公司提供了默认的 JDBC-ODBC 的实现！打开你 JDK 的安装目录找到 jre\lib 目录了吧！看到一个 rt.jar 文件吧！用 WinRAR 打开它，看它的目录机构，有类似 sun/jdbc/odbc 吧！它就是整个 JDBC-ODBC 的实现！同时你可以看到其中的类名去掉前面的 JdbcOdbc 是不是和你研究的 JDBC API 中的接口名称基本相同啊！想知道为什么吗？继续关注本博客吧！
3. 该参数代表驱动类的位置即驱动的路径，因此它总是以 “.” 分隔，区别与后面的 URL 哦！

下面是其一般格式：

```
try {
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    //sun 的 JDBC-ODBC 的驱动
    //Class.forName("oracle.jdbc.driver.OracleDriver"); //Oracle 的驱动
    //Class.forName("com.sybase.jdbc.SybDriver"); // sybase 的驱动
} catch (ClassNotFoundException e) {
```



```
        System.err.println("Error loading driver: " + e);
    }
}
```

二、定义连接的 URL

一旦加载了数据库的驱动程序，你就可以通过使用与特定数据库对应的 URL 得到其连接了！此 URL 的一般格式为 `jdbc:subprotocol:userDb`，`jdbc` 是所有 JDBC 应用程序的协议名，`subprotocol` 是对应于不同数据库提供商的子协议，`sun` 公司的 JDBC-ODBC 桥的子协议为 `odbc`，开源的 `Mocki` 软件为 `mocki` 等等，你只要查找相应提供商的手册即可！`userDb` 一般为用户数据库，在我们的例子中使用的就是通过“数据源”设置的数据库名“**DbAccess**”。（URL 相对比较复杂——可以是不同数据库提供商的、可以是本地的或网络的等等，具体的可以阅读不同数据库提供商的操作手册）下面是本例中 URL 的代码片断：

```
String url = "jdbc:odbc:DB";
```

此 URL 与我们通常的 `http`、`ftp` 等协议的 URL 格式一样，以“:”分隔哦！

三、建立连接

建立连接一般指建立客户程序和数据库的连接。建立此连接一般包含三个参数：1.URL；2.用户名；3.用户密码。一般操作如下：

```
String username = "mxd";
String password = "mxd";
Connection con = DriverManager.getConnection(url, username, password);
```

四、创建 Statement 对象

`Statement` 对象是一个用于客户发送查询语句得到结果集的对象，没有该对象就不能数据库的操作。一般使用下面的代码片断：

```
Statement statement = con.createStatement();
```

五、执行 SQL 查询或者其它 SQL 操作

在建立 `Statement` 对象后可以向数据库发送 SQL 语句了。这些 SQL 语句可以是 DML 的查询等语句，也可以是 DDL 的创建数据库等语句。一般使用最多的是 DML 的查询语句——`select`，与查询对于的方法为 `executeQuery`，返回 `ResultSet` 类型的结果集。DDL 语句一般对应于方法 `executeUpdate`，返回表示该操作影响的数据的行数。在不知道究竟是 DML 和 DDL 语句时可以使用 `execute` 方法，然后再根据返回的类型确定是执行 DML 语句还是 DDL 语句。下面是简单查询语句的代码片断：UPDATE, INSERT, or DELETE

```
String query = "select * from Employee ;
ResultSet rs = stmt.executeQuery(query);
```

六、处理结果

执行上面的查询操作后，返回 `ResultSet` 类型的结果集，下面的工作就是处理此结果集了。结果集一般是按行处理，使用 `ResultSet` 的 `next` 方法从一行移动到另外一行。移到某一行再使用该结果集的 `getXXX` 方法，可以按列的名称或者按列的序号（列是从 1 开始的）得到结果集中每行对应的列的值！下面是本例中遍历结果集中各行，取得前四行，然后打印它的代码片断（实际使用时，可以先通过该结果集的元素据得到有多少行、多少列及其各列名称，然后再取列的值）：

```
while(rs.next()){
    System.out.print(rs.getString(1));System.out.print("      ");
    System.out.print(rs.getString(2));System.out.print("      ");
    System.out.print(rs.getString(3));System.out.print("      ");
    System.out.print(rs.getString(4));System.out.println("      ");
}
```

七、关闭连接

建立连接往往需要耗费资源，且数据库中建立连接的个数一般也是有限的，在不需要此连接后，一般要显式地关闭连接。下面是关闭连接的代码：

```
connection.close();
```

以下是全部代码：

```
import java.sql.DriverManager;
import java.sql.Connection ;
import java.sql.Statement;
import java.sql.ResultSet;
import javax.swing.JTable ;
public class JDBC{
    public static void main(String[] args)throws Exception {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection con = DriverManager.getConnection("jdbc:odbc:DB","mxd","mxd");
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery("select * from Computer");
        while(rs.next()){
            System.out.print(rs.getString(1));System.out.print("      ");
            System.out.print(rs.getString(2));System.out.print("      ");
            System.out.print(rs.getString(3));System.out.print("      ");
            System.out.print(rs.getString(4));System.out.println("      ");
        }
    }
}
```

更多精彩请关注：

<http://blog.163.com/miaoxiaodong78/>