NetBeans JDBC 驱动程序设置

陈拓

2006年5月20日

最后修改日期 2006 年 9 月 5 日

本文中我们介绍在 NetBeans 5.0 中怎样设置和测试 Oracle 和 SQL Server 的 JDBC 驱动程序,以及故障排除。

以下对数据库的操作以 SQL Server2000 和 Oracle 10g 为例。

一、 从 Oracle 和 Microsift 网站上下载 JDBC 驱动程序

一、 下载 Oracle JDBC 驱动程序

1. 网址:

http://www.oracle.com/technology/global/cn/software/tech/java/sqlj_jdbc/index.html

- 2. 在 netbeans 安装目录,例如 D:\Program Files\netbeans-5.0 下创建路径: D:\Program Files\netbeans-5.0\jdbc\OracleJDBC\lib。
- 3. 下载 4 个 java 归档文件放入其中:
 - 1) ocrs12.jar
 - 2) ojdbc14.jar
 - 3) ojdbc14dms.jar
 - 4) orai18n.jar

■ 下载 SQL Server 2000 Driver for JDBC Service Pack 3, 支持 JDK 1.4

1. 网址:

http://www.microsoft.com/downloads/details.aspx?FamilyID=07287b11-0502-461a-b138-2aa54bfdc03a&displaylang=en

- 2. 执行 setup.exe 安装 SQL Server JDBC 驱动程序。在安装目录下有帮助文件。
- 3. 在 netbeans 安装目录,例如 D:\Program Files\netbeans-5.0 下创建路经: D:\Program Files\netbeans-5.0\jdbc\ SQLServerJDBC\lib。
- 4. 将 setup.exe 安装的下面 3 个 java 归档文件放入其中:
 - 1) msbase.jar
 - 2) mssqlserver.jar
 - 3) msutil.jar

二、添加库

要将 JAR 文件添加到现有应用程序中,可使用库管理器。库管理器提供对打包为 jar 文档的库文件的管理,可以很方便地将库添加到应用程序中。

■ 使用库管理器:

1. 在主菜单中选择"工具">"库管理器"。

- 2. 在"库管理器"对话框中,单击"新建库"。
- 3. 在"新建库"对话框中,命名该库,例如 OracleJDBC。单击"确定"。



4. 单击"添加 JAR/文件夹", 然后找到并添加类路径、源或 Javadoc 所需的 JAR 文件 (前面下载的驱动程序)。



5. 单击"确定"。

SQL Server 驱动程序库的添加方法类似。

三、 在运行环境窗口中进行测试

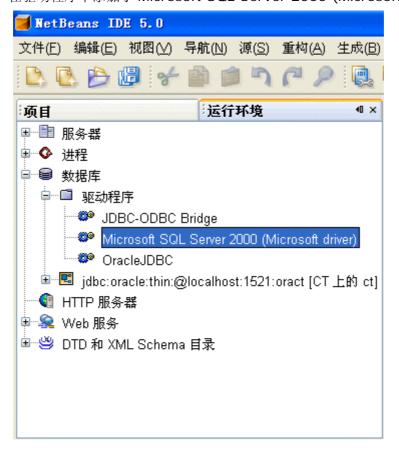
在 NetBeans 的运行环境中可以通过连接数据库对 JDBC 驱动程序进行测试。在运行环境中还可以进行一些数据库管理工作,如创建维护表和视图、查看数据、执行存储过程、执行 SQL 命令等。下面以 SQL Server 为例进行 JDBC 的测试。

- 1. 切换到"运行"窗口,展开"数据库"。
- 2. 右击 驱动程序 > 添加驱动程序, 打开添加 JDBC 驱动程序对话框。



3. 单击 添加 > 选择前面下载的 3 个 jar 驱动程序文件 msbase.jar、mssqlserver.jar 和 msutil.jar > 确定。

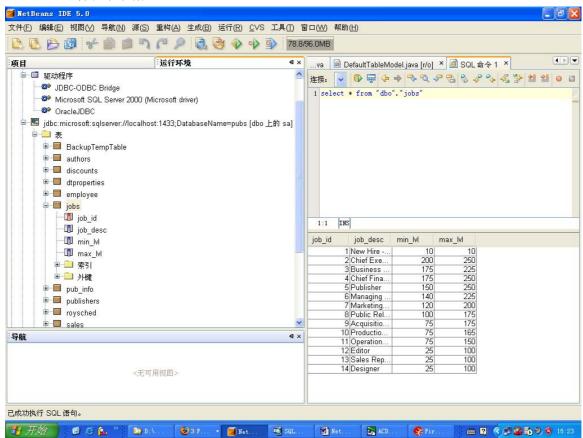
如图,在驱动程序中添加了 Microsoft SQL Server 2000 (Microsoft driver)。



4. 右击这个驱动程序 > 连接方法,打开"新建数据库连接"对话框 > 修改 数据库 URL,输入用户名和口令 > 确定。



5. 查看表中数据。



四、 创建测试项目

1. 在主菜单中选择"文件">"新建项目"。

- 2. 新建项目窗口:
 - 1) 选择项目 > 类别: 常规 > 项目: Java 应用程序 > 下一步
 - 2) 名称和位置 > 项目名称: JDBCtest > 取消创建主类选择 > 完成

五、 在项目库中导入驱动程序库

方法一,从库管理器中添加驱动程序到项目中:

- 1. 在"项目"窗口中展开 JDBCtest 项目,右击"库"
- 2. 选择"添加库..."



3. 在"添加库"窗口中选择库: OracleJDBC和 SQLServerJDBC



4. 单击"添加库"完成导入



方法二,直接添加驱动程序到项目中:

如果没有在库管理器中导入 JDBC 驱动程序,可以在方法一的第 2 步选择"添加 JAR/文件夹...",直接选择驱动程序的 jar 文件,并添加到项目库中。

六、 用于测试连接的代码示例

一、 用于 SQL Server 的测试连接的代码示例

安装了 Microsoft SQL Server 2000 JDBC 驱动程序后,可以通过两种方式从您的程序连接到数据库:使用连接 URL,或使用 JNDI 数据源。本文介绍如何使用连接 URL 配置和测试数据库连接。

连接到数据库的一种方法是通过 JDBC 驱动程序管理器,使用 DriverManager 类的 getConnection 方法。使用此方法时,最简单的方式是使用一个包含 URL、用户名和密码 的字符串参数。本文中的以下几节将介绍如何从 JDBC 程序载入 Microsoft SQL Server 2000 JDBC 驱动程序。

设置 CLASSPATH 变量 (这一步实际上可以省略)

Microsoft SQL Server 2000 JDBC 驱动程序.jar 文件必须在 CLASSPATH 变量中列出。CLASSPATH 变量是 Java 虚拟机 (JVM) 用于在您的计算机上查找 JDBC 驱动程序的搜索字符串。如果驱动程序未在 CLASSPATH 变量中列出,尝试载入驱动程序时将出现以下错误信息:

java.lang.ClassNotFoundException:com/microsoft/jdbc/sqlserver/SQLServerDriver

设置系统 CLASSPATH 变量,加入以下各项:

- \Your installation path\Lib\Msbase.jar
- \Your installation path\Lib\Msutil.jar
- Your installation path\Lib\Mssqlserver.jar

以下是一个配置好的 CLASSPATH 变量的示例:

CLASSPATH=.;c:\program files\Microsoft SQL Server 2000 Driver for JDBC\lib\msbase.jar;c:\program files\Microsoft SQL Server 2000 Driver for JDBC\lib\msutil.jar;c:\program files\Microsoft SQL Server 2000 Driver for JDBC\lib\mssqlserver.jar

注册驱动程序

注册驱动程序的目的是为了通知 JDBC 驱动程序管理器载入哪个驱动程序。当使用 class.forName 函数载入驱动程序时,您必须指定驱动程序的名称。以下是 Microsoft SQL Server 2000 JDBC 驱动程序的名称:

com.microsoft.jdbc.sqlserver.SQLServerDriver

下面的代码示例演示如何注册驱动程序:

Driver d = (Driver)Class.forName("com.microsoft.jdbc.sqlserver.SQLServerDriver").newInstance();

传递连接 URL

必须以连接 URL 的形式传递数据库连接信息。以下是 Microsoft SQL Server 2000 JDBC 驱动程序的模板 URL。请用您数据库的值替换以下值:

jdbc:microsoft:sqlserver://servername:1433

下面的代码示例演示如何指定连接 URL:

con = DriverManager.getConnection("jdbc:microsoft:sqlserver://localhost:1433", "userName", "password");

服务器名称值可以是 IP 地址或主机名(假定您的网络可以将主机名解析为 IP 地址)。您可以通过对主机名执行 PING 命令来进行测试,验证是否可以接收到响应,以及响应的 IP 地址是否正确。

服务器名称后面的数字值是数据库侦听的端口号。上文列出的值是示例默认值。确保用您的数据库使用的端口号替换该值。

要获取连接 URL 参数的完整列表,请参见 Microsoft SQL Server 2000 JDBC 驱动程序 HTML 帮助,或参见联机指南。请参见"连接字符串属性"一节。

下面的代码示例尝试连接到数据库,并显示数据库名称、版本和可用编目。请用您服务器的值替换代码中的服务器属性:

```
import java.*;
public class SQLServerJDBCTest {
   private java.sql.Connection con = null;
   private final String url = "jdbc:microsoft:sqlserver://";
   private final String serverName= "localhost";
   private final String portNumber = "1433";
   private final String databaseName= "pubs";
   private final String userName = "sa";
   private final String password = "wyfd";
   // 告诉驱动器使用服务器端游标,
   // 它允许在一个连接上的多个活动语句。
   private final String selectMethod = "cursor";
   /** Creates a new instance of SQLServerJDBCTest */
   public SQLServerJDBCTest() {
   private java.sql.Connection getConnection(){
    try{
      // A. 注册 SQL Server JDBC 驱动程序
      Class.forName("com.microsoft.jdbc.sqlserver.SQLServerDriver");
      // B. 创建新数据库连接
      con =
java.sql.DriverManager.getConnection(getConnectionUrl(), userName, password)
      if(con!=null) System.out.println("Connection Successful!");
     }catch(Exception e){
      e.printStackTrace();
      System.out.println("Error Trace in getConnection() : " +
e.getMessage());
    }
    return con;
   private String getConnectionUrl(){
    return
url+serverName+":"+portNumber+";databaseName="+databaseName+";selectMethod
="+selectMethod+";";
   }
   private void closeConnection(){
    t.rv{
      if(con!=null)
        con.close();
      con=null;
      }catch(Exception e){
```

```
e.printStackTrace();
      }
   }
      显示驱动器属性,数据库详细信息
   public void displayDbProperties(){
    java.sql.DatabaseMetaData dm = null;
    java.sql.ResultSet rs = null;
      try{
        con= this.getConnection();
        if(con!=null){
         dm = con.getMetaData();
         System.out.println("驱动器信息: ");
         System.out.println("\t 驱动器名字: "+ dm.getDriverName());
         System.out.println("\t 驱动器版本: "+ dm.getDriverVersion ());
         System.out.println("\n 数据库信息: ");
         System.out.println("\t 数据库名字: "+ dm.getDatabaseProductName());
         System.out.println("\t 数据库版本: "+
dm.getDatabaseProductVersion());
         System.out.println("显示可用的数据库目录: ");
         rs = dm.getCatalogs();
         while(rs.next()){
           System.out.println("\tcatalog: "+ rs.getString(1));
          }
         rs.close();
         rs = null;
          closeConnection();
        }else System.out.println("Error: No active Connection");
      }catch(Exception e){
      e.printStackTrace();
      dm=null;
   public static void main(String[] args) {
      SQLServerJDBCTest sQLServerJDBCTest = new SQLServerJDBCTest();
      sQLServerJDBCTest.displayDbProperties();
```

如果此代码运行成功, 其输出结果应类似于以下内容:

```
Connection Successful!
驱动器信息:
驱动器名字: SQLServer
驱动器版本: 2.2.0040

数据库信息:
数据库名字: Microsoft SQL Server
数据库版本: Microsoft SQL Server 2000 - 8.00.2039 (Intel X86)
May 3 2005 23:18:38
Copyright (c) 1988-2003 Microsoft Corporation
Personal Edition on Windows NT 5.1 (Build 2600: Service Pack 2)

显示可用的数据库目录:
catalog: ckgl
```

```
catalog: coal
catalog: master
catalog: msdb
catalog: Northwind
catalog: pubs
catalog: student
catalog: tempdb
生成成功(总时间: 8 秒)
```

有关排除连接故障的基本信息

下面是尝试连接到 SQL 服务器时常见的错误信息:

java.sql.SQLException: [Microsoft] [SQLServer 2000 Driver for JDBC] [SQLServer] Login failed for user 'user'.Reason: Not associated with a trusted SQL Server connection.

如果将 SQL Server 2000 的验证模式设置为"Windows 验证模式",则会出现此错误信息。Microsoft SQL Server 2000 JDBC 驱动程序不支持使用 Windows NT 验证进行连接。您必须将 SQL Server 的验证模式设置为"混合模式",该模式既允许 Windows 验证,也允许 SQL Server 验证。

java.sql.SQLException:[Microsoft][SQLServer 2000 Driver for JDBC]This version of the JDBC driver only supports Microsoft SQL Server 2000. You can either upgrade to SQL Server 2000 or possibly locate another version of the driver.

当您尝试连接到 SQL Server 2000 以前的 SQL Server 版本时,则会出现此错误信息。 Microsoft SQL Server 2000 JDBC 驱动程序仅支持与 SQL Server 2000 进行连接。

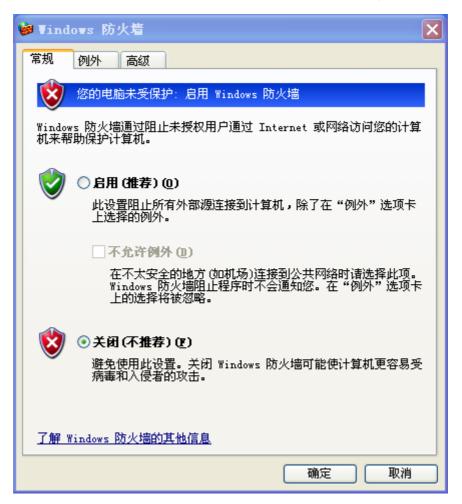
二、 用于 Oracle 的测试连接的代码示例

```
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.SQLException;
public class OracleJDBCTest {
   /** Creates a new instance of OracleJDBCTest */
   public OracleJDBCTest() {
   public
            static
                     void
                            main(String[] args) throws SQLException,
ClassNotFoundException {
      // A. 注册 Oracle JDBC 驱动程序。下面两种方法都可以
      //DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
      Class.forName("oracle.jdbc.driver.OracleDriver");
      // B. 创新新数据库连接
      Connection
                                          conn
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:oract", "hr",
      // C. 取得连接数据(元数据)
      DatabaseMetaData md = conn.getMetaData();
      System.out.println("数据库版本: ");
System.out.println("----
      System.out.println(md.getDatabaseProductVersion());
      System.out.println();
      System.out.println("驱动程序名称与版本:");
System.out.println("-----
```

注意: 有时连接错误是由防火墙造成的,所以在用 JDBC 进行数据库开发时要关闭防火墙,包括 winXP 自带的防火墙。

关闭 winXP 自带的防火墙

• 开始 > 控制面板 > Windows 防火墙 > 打开 Windows 防火墙



• 选择关闭 > 确定。

[参考文献]

1. NetBeans IDE 5.0 帮助

2.	如何开始使用 Microsoft JDBC,
۷.	http://support.microsoft.com/default.aspx?scid=kb;zh-cn;313100