Derby数据库是一个纯用Java实现的内存数据库，属于Apache的一个开源项目。由于是用Java实现的，所以可以在任何平台上运行；另外一个特点是体积小，免安装，只需要几个小jar包就可以运行了。

Derby数据库有两种运行模式：

1） 内嵌模式。Derby数据库与应用程序共享同一个JVM，通常由应用程序负责启动和停止，对除启动它的应用程序外的其它应用程序不可见，即其它应用程序不可访问它；

2） 网络模式。Derby数据库独占一个JVM，做为服务器上的一个独立进程运行。在这种模式下，允许有多个应用程序来访问同一个Derby数据库。

在Apache上，Derby有4种发布包，这里以bin发布包为例。bin发布包中含有包含了执行derby数据库工具、设置derby数据库环境的脚本文件、Demo、jar文件等。

1、 安装Derby数据库

只需要从Derby官方网站下载Derby的zip或者tar包，解压就可以了。这里以db-derby-10.4.1.3-bin版本为例，解压后得到以下目录：

1） bin目录，包含了一些工具脚本和设备环境的脚本；

2） demo目录，包含了一些实例程序；

3） docs目录，包含了Derby的文档；

4） javadoc目录，包含了Derby的API文档；

5） lib目录，包含了Derby数据库的jar文件；

6） test目录，Derby的一些测试jar包；

2、 使用Derby脚本

Derby提供了几个脚本来操作Derby数据库，在使用这些脚本前，你必须先设置好Derby的运行环境。

下面的例子都是在命令行下设置环境变量，这些设置都是临时的，如果你新开一个命令行窗口，则需要重新设置一遍，如果想要使环境变量永久生效，可以在我的电脑中进行设置。

首先设置好DERBY\_HOME这个环境变量，为DERBY\_HOME指定你的derby目录，假如你的derby解压到了E:\ db-derby-10.4.1.3-bin目录下，则可以在命令行中如下设置：

set DERBY\_HOME=E:\ db-derby-10.4.1.3-bin

将DERBY\_HOME\bin目录添加到PATH环境变量中：

set path=%DERBY\_HOME%\bin;%PATH%

这样可以简化你稍后在命令行中的输入，否则你每次都必须使用脚本的全路径或者你必须到DERBY\_HOME\bin目录中才能执行脚本。

最后需要Derby的jar包添加到classpath环境变量中，在DERBY\_HOME%\bin目录中提供了几个脚本用于设置classpath，以简化你手工在classpath中添加jar包的麻烦：

1） setEmbeddedCP。当使用内嵌模式来运行Derby时，可以使用该脚本来设置。该脚本将derby.jar和derbytools.jar添加到环境变量中；

2） setNetworkServerCP。当使用网络模式来运行Derby时，用该脚本来设置Derby服务端的classpath变量。该脚本将derbynet.jar添加到环境变量中；

3） setNetworkClientCP。当使用网络模式来运行Derby时，用该脚本来设置Derby客户端的classpath变量。该脚本将derbyclient.jar和derbytools.jar添加到环境变量中。

一般只有当你通过derbyrun.jar来运行Derby工具时才会使用这些脚本。

Derby提供了三个工具脚本：1）sysinfo；2）ij；3）dblook。运行这三个脚本时，如果你没有设置classpath环境变量，这些脚本会自动进行设置。

1） sysinfo

使用sysinfo可以显示你的Java环境信息和Derby的版本信息。使用方法就是在命令行下直接输入：

sysinfo.bat

2） dblook

使用dblook可以将全部或者部分数据库的DDL定义导出到控制台或者文件中。使用方法：

dblook.bat –d <sourceDBUrl> [Options]

3） ij

使用ij工具来进行数据库交互，执行SQL脚本，如查询、增删改、创建表等等。在命令行下输入：

ij.bat

即可启动ij工具，然后就可以开始执行SQL脚本了。当要退出ij工具时，在命令行下输入

exit;

即可。

3、 使用ij脚本

1） 运行内嵌模式的Derby数据库

在命令行中输入ij.bat后启动ij工具。然后通过如下命令创建数据库，并与数据库创建连接：

connect ‘jdbc:derby:firstdb;create=true’;

通过connect命令可以与指定数据库创建连接，通过一个JDBC URL来指定与哪个数据库创建连接。ij命令是不区分大小写的。

参数中jdbc:derby是Derby数据库的驱动协议；firstdb是数据库命，由于没有指定路径，数据库将会被创建在当前你命令行下所在的目录下；create=true表示如果数据库不存在，则创建该数据库；“;”是ij命令的终止符。

当数据库创建成功时，Derby会在当前你命令行下所在的目录下创建一个与数据库命一致（这里是firstdb）的目录，其中存放了数据库的文件。

与数据库连接上后，就可以开始执行SQL脚本了，如创建一个表格：

create table firsttable(id int primary key, name varchar(20));

然后插入记录：

insert into firsttable values(1, ‘Hotpepper’);

也可以执行查询：

select \* from firsttable;

也可以通过run命令来执行sql文件：

run 'E:\derby\demo\programs\toursdb\ToursDB\_schema.sql';

最后通过exit;来退出ij工具。

你可以在当前你命令行下所在的目录中找到一个derby.log的日志文件，derby在其中记录的数据库启动、关闭的信息。

2） 运行网络模式的Derby数据库

这种模式下，需要使用两个控制台窗口，一个用于启动Derby数据库服务端，另一个做为访问Derby数据库的客户端。

可以通过DERBY\_HOME\bin目录下的startNetworkServer.bat来启动Derby数据库服务端，只需要在命令行中输入：

startNetworkServer.bat

数据库就启动了，启动成功会在控制台输出如下信息：

已使用基本服务器安全策略安装了安全管理程序。

Apache Derby Network Server － 10.4.1.3 - (648739) 已启动并且已准备好 2008-09-06

00:38:12.540 GMT 时在端口 1527 上接受连接

在另一个控制台使用ij命令访问Derby数据库服务端，在输入ij.bat启动ij工具后，通过如下命令建立与服务端的连接，并创建一个数据库：

connect 'jdbc:derby://localhost:1527/seconddb;create=true';

参数中的数据库命部分和内嵌模式不同，这里使用了“//localhost:1527/”，访问网络模式的URL需要指定服务器的IP地址和端口，其它的就和内嵌模式一样了。

与服务端连接上后，就可以开始执行SQL脚本了，如创建一个表格：

create table firsttable(id int primary key, name varchar(20));

然后插入记录：

insert into firsttable values(1, ‘Hotpepper’);

也可以执行查询：

select \* from firsttable;

也可以通过run命令来执行sql文件：

run 'E:\derby\demo\programs\toursdb\ToursDB\_schema.sql';

最后通过exit;来退出ij工具

4、 在Java应用程序中访问Derby数据库

使用Java代码访问Derby数据库与访问其它数据库的区别如下：

1） JDBC驱动的不同；

2） 数据库连接URL的不同；

3） 在访问内嵌模式数据库时，需要显示关闭数据库。

下面分别实例访问内嵌模式和网络模式Derby数据库的代码

1） 访问内嵌模式Derby数据库

String driver = “org.apache.derby.jdbc.EmbeddedDriver”;

String url = “jdbc:derby:firstdb;create=true”;

Connection conn;

try {

Class.forName(driver);

conn = DriverManager.getConnection(url);

}catch(Exception e) {

……

}finally {

……

DriverManager.getConnection("jdbc:derby:;shutdown=true");

}

建立好连接后，其它的数据操作，如查询、更新数据都和其它数据库一样，这里不详述。有一点需要注意，通过Java应用程序访问内嵌模式Derby数据库时，应用程序有责任需要在程序结束时关闭Derby数据库，如上面代码finally中的

DriverManager.getConnection("jdbc:derby:;shutdown=true");

shutdown参数用于关闭Derby数据库，如果url中指定了数据库命，则只会关闭指定的数据库，而不会关闭整个Derby数据库。数据库关闭成功时，Derby会抛出一个错误码为XJ015和一个08006的异常表示关闭成功，应用程序可以不处理这两个异常。

2） 访问网络模式Derby数据库

网络模式和内嵌模式的不同出在于：

A． 数据库连接URL的不同；

B． 应用程序退出时无效关闭Derby数据库；

C． 数据库驱动的不同；

String driver = “org.apache.derby.jdbc.ClientDriver”;

String url = “jdbc:derby: //localhost:1527/firstdb;create=true”;

Connection conn;

try {

Class.forName(driver);

conn = DriverManager.getConnection(url);

}catch(Exception e) {

……

}

由于网络模式下，Derby数据库做为一个独立运行的数据库，可以被多个应用程序所访问，所以应用程序在运行结束时不应该关闭Derby数据库。

5、 使用derbyrun.jar

通过derbyrun.jar也可以使用上面提到的sysinfo、ij、dblook三个脚本的功 能，在使用derbyrun.jar前，除了DERBY\_HOME环境变量外，你必须显示设置好classpath环境变量，可以通过上面提到的 setEmbeddedCP、setNetworkServerCP、setNetworkClientCP几个脚本来设置，具体使用哪个脚本取决于你如 何使用Derby数据库。

可以通过java –jar %DERBY\_HOME%\lib\ derbyrun.jar来运行derbyrun中的工具：

1） 使用sysinfo。java –jar %DERBY\_HOME%\lib\ derbyrun.jar sysinfo

2） 使用ij。java –jar %DERBY\_HOME%\lib\ derbyrun.jar ij

3） 使用dblook。java –jar %DERBY\_HOME%\lib\ derbyrun.jar dblook