# DERBY系列7-DERBY程序部署



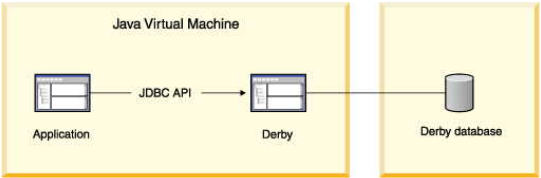
# 介绍

在通常的情况下，一旦你已经开发出一种DERBY应用程序和数据库时，你的DERBY库和和数据库会通过某些方式分配到你的用户中，这种方式叫做部署，本章介绍了derby部署的详细信息

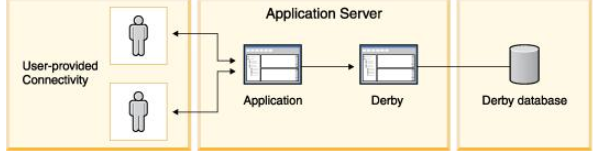
# 嵌入式应用程序部署概述

在嵌入式环境下，derby应用程序会运行在同一个JVM下，应用程序可以作为单用户和多用户服务器下，derby在用户定义的服务器框架下嵌入应用程序进行关联

下面单用户部署流程图



多用户



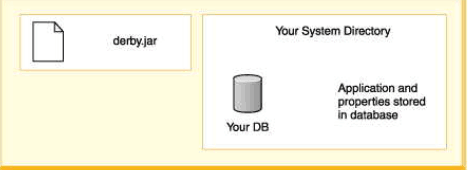
## 部署derby环境介绍

你需要准备这些包才可以保证derby在JAVA应用程序下运行

1. DERBY.JAR包
2. 为应用程序准备的库
3. 在系统目录下的数据库

在下图中，是应用程序部署的环境，在上图中是应用程序下所需要的东西，一个derby的配置属性文件和4个对象，在下图中是在应用程序中单独存在的数据库





### 嵌入式系统和属性

在数据库中，database-wide在对属性做简单的部署，使参数能够更方便的去响应接收，有关database-wide可以在JDBC基础下进行详细了解

1. 如果你对system-wide属性进行了设置，你可以去看下database-wide的属性是否被更改
2. 那就是任何的属性都会被设置在derby.proprties文件中吗？有一些属性只能建立在system-wide的基础上，这样可以使你在部署数据库时更加方便，而且derby会自动帮你进行简单的部署配置，所以你不用担心在目录下的部署文件

### 创建一个只读的数据库媒体

如果你先创建一个只读的数据库媒体，你需要准备一下步骤

1. 创建一个可填充的读写媒体
2. 提交所有事务和关闭所约定的derby，如果你不关闭所约定的derby，derby将会下次系统启动时不自信只读媒体
3. 删除数据库目录下的TMP文件，如果存在于这个目录，derby会尝试去删除并返回启动制度媒体下的数据库
4. 为只读下的数据库中的DERBY.STORAGE.TEMPDIRECTORY写入本地路径，derby需要使用SQL命令行写入指令

|  |
| --- |
| derby.storage.tempDirectory=c:/temp/mytemp |

1. 配置数据库，为输出流发送错误日志

有关derby.stream.error.file属性的更多信息，可以去参考derby手册

|  |
| --- |
| derby.stream.error.file=c:/temp/mylog.LOG |

确保你已经是设置了这些属性，以便去部署数据库。

### 部署数据库下只读媒体

为了去部署数据库的制度媒体，你需要准备几个步骤

1. 将数据库目录移动到只读媒体，包括数据库自动生成的字典目录和服务属性配置文件
2. 在这时，你已经不能为数据库插入或更新任何数据

### 将只读数据库转移到归档文档中(jar或zip)

1. 一旦你在derby中创建了数据库，他可以以只读模式中存储在JAR或zip文件，事实上，一个jar或zip文件中包含了许多derby数据库。

你可以使用以下命令将Sales目录放入JAR文件

|  |
| --- |
| cd C:\london  jar cMf C:\dbs.jar sales |

你也可以将多个数据库放进jar

|  |
| --- |
| cd C:\london  jar cMf C:\dbs.jar sales products\boiledfood |

### 进入zip/jar文件下的只读数据库

通过以下命令进入

|  |
| --- |
| jdbc:derby:jar:(pathToArchive)databasePathWithinArchive |

**pathToArchive是进入文件的绝对路径，databasePathWithinArchive**进入数据库的相对路径，比如

|  |
| --- |
| jdbc:derby:jar:(C:/dbs.jar)products/boiledfood  jdbc:derby:jar:(C:/dbs.jar)sales |

如果你在此过程中出现了问题，使用工具去检查文件的信息，databasePathWithinArchive必须匹配对应的路径，你可能需要重新检查URL地址是否是正确的填写

|  |
| --- |
| jdbc:derby:jar:(C:/dbs.jar)/products/boiledfood |

### 利用类路径进入JAR文件下的数据库

一旦你建数据库存储到归档文件中，他会放置在类路径中，这可以更方便的进入数据库

当jar或者归档文件是类路径的一部分，你需要在JAR包中声明类路径才能去连接

你可以使用这些方法

1. 再启动derby前在jar文件中设置这些属性

|  |
| --- |
| CLASSPATH="C:\dbs.jar;%CLASSPATH%" |

2.通过jar或者归档文件中的URL连接地址去连接数据库

|  |
| --- |
| jdbc:derby:classpath:databasePathWithinArchive |
| jdbc:derby:classpath:products/boiledfood |

## 在数据库中的只读媒体和可改变数据库数据

在DatabaseMetaData.isReadOnly属性中返回TRUE

### 在数据库中加载类

在嵌入式环境中，但数据库中存在程序逻辑，derby可以从jar文件中加载derby类

### 类加载

在数据库中通过存储一个或多个应用程序类和资源数据库，使数据库能够加载类，以便访问，但是你需要在程序中去调用这些类。

这些是实现步骤

1. 一个标准的JAVA包(JAVA.\*,JAVAX);
2. 将这些类关联进JAVA环境

这样derby系统会加载来自JAR文件下的方言和数据库所需要的类，derby会为用户自动加载所需要的jar环境，下面是简单的实例

|  |
| --- |
| jar cf travelagent.jar travelagent/\*.class. |

在一些IDE工具会生成基于JAR文件的应用，如果你的应用程序需要加载其他文件夹的类，你可以利用方法去建立关系

1. 从jar文件中提取第三方类，使用这个方法，你只需要从类中提取一部分jar文件
2. 将第三方jar文件存储在数据库中

使用这个方法你需要更多的jar文件，你需要将derby升级到最高版本

1. 在类路径下进行部署

使用这个方法是类已经状态到机器中

**在数据库中添加jar文件或文件夹**

在数据库中时候用存储过程去安装替换和删除jar文件，当你已经在数据库中安装了jar文件，通过SQLIDENTIFER去定义derbyjar名字

**NOTE:一旦jar文件被加载，你不能在jar文件中新增任何的类，你只能通过替换jar文件**

**例子**

安装jar文件

通过SQLJ.INSTALL\_JAR存储过程来进行安装

|  |
| --- |
| -- SQL statement  CALL sqlj.install\_jar(  'tours.jar', 'APP.Sample1', 0)  -- SQL statement  -- using a quoted identifier for the  -- Derby jar name  CALL sqlj.install\_jar(  'tours.jar', 'APP."Sample2"', 0) |

.删除JAR文件

|  |
| --- |
| -- SQL statement  CALL **sqlj.remove\_jar**(  'APP.Sample1', 0) |

替换JAR文件

|  |
| --- |
| -- SQL statement  CALL **sqlj.replace\_jar**(  'c:\myjarfiles\newtours.jar', 'APP.Sample1') |

**启动数据库类加载属性**

一旦你在数据库中已经添加一个或多个jar文件，你必须在jar文件中的derby.Database.classpth属性中去设置属性，如果derby没有找到所需的类，他会在类路径中重新检索这些类

比如

|  |
| --- |
| CALL SYSCS\_UTIL.SYSCS\_SET\_DATABASE\_PROPERTY(  'derby.database.classpath',  'APP.ToursLogic:APP.ACCOUNTINGLOGIC') |

你不用去考虑数据库与JDBC的交互方式，应用程序会匹配安全的方式去启动derby。

**NOTE:在derby中，GETRESOURCE方法并不支持**

### **在类路径中动态更改jar文件**

类路径动态化可以是存在于数据库中的jar文件具有灵活性，当你想利用SQL工具去改变类路径，derby可以在不用重新启动时去加载更改的新类

为了确保derby能够在不用重新启动时去加载新的类

1. 保证配置数据库时的数据库级正确，在derby.database.classpath中设置两个部分有效的名字
2. 它会在derby.database.classpath属性中替换新类文件并提供有效的值

如果配置不成功，你可以重新启动试试

**动态更换所需要知道的：**

1. **当你在**derby.database.classpath属性中改变熟悉，所有的类会重新加载，JAR文件也会重新加载
2. 记住用户路径的搜索内容
3. 任何存在的声明会加载进类中，他可能会报出“CLASSNOTFOUND”的错误
4. 缓存对象不会将新加载的类存储，如果加载进了，他会报出异常信息

# derby服务器端编码

这章告诉你derby在服务器端的编码方式

## database-wide JDBC下的规则

方法在程序端被称用程序调用方法，方法在derby中被称为数据库端规则

一个程序端的方法同样可以被称为数据库端规则，唯一的区别是你调用的方式。

## 数据库端JDBC规则和网络连接

许多数据库通过调用规则实现连接，DERBY也不例外，JDBC下提供了相同的连接方式去连接数据库，通过**connection URL\_jdbc:default:connection** 去连接使用数据库

|  |
| --- |
| Connection conn = DriverManager.getConnection("jdbc:default:connection"); |

Derby中URL属性不支持，带用户名密码的连接方式，他会忽略掉这些信息

### 数据库端JDBC网络连接的条件

为了遵守JDBC的规则，它具有以下限制条件

1. 只在一个存储过程中提交回滚
2. 不能自动提交连接属性
3. 不能以“ROUTINE”命名（在使用增删改查时使用**changeTables**存储过程 **，changeTables**不能存在”/”）
4. 不能删除命名为routine的表
5. 不能在一个类静态初始化器质性DDL语句
6. 在连接对象时，网络连接属性会将自动提交的模式默认为FALSE;

### Database-side JDBC下进行本地连接