



# Java DB

## Oracle Distribution of Apache Derby

*An open source relational database management system*

# Java DB At a Glance

---

- **Java DB** is a relational database management system (RDMS) based on **Java** and **SQL**
- **Java DB** is the Oracle release of the **Apache Derby**  
without any modification to the underlying source code
  - Full-featured and easy-to-use
  - Transaction-protected and crash-recoverable
  - Embeddable in applications
  - Pure Java and portable
    - CDC FP 1.1, Java 5, Java 6, and Java 7 (everywhere from *tablets* to *mainframes*)
  - Included in the JDK 7
    - In the `db` directory of JDK installation
    - Otherwise, you can always download it
  - Compact (2.6 MB)

# Deployment Options

---

- Embedded (configuration or mode)
  - Derby being started by a simple single-user Java application.
  - Derby runs in the same JVM as the application.
  - Derby can be almost ***invisible*** to the end user because it is started and stopped by the application and often requires no administration.
- Server (or Server-based or Network Server or client/server configuration)
  - Derby being started by an application that provides multi-user connectivity to Derby databases across a network.
  - Derby runs in the JVM that hosts the Server. Applications connect to the Server from different JVMs to access the database.

# Derby Libraries

Engine library	
<code>derby.jar</code>	Derby engine and embedded JDBC driver
Tools libraries	
<code>derbytools.jar</code>	Derby tools such as ij, dblook and importer/exporter etc.
<code>derbyrun.jar</code>	helper to run Derby tools and network server
Network Server library	
<code>derbynet.jar</code>	Derby Network Server and a reference to the engine <code>derby.jar</code> file
Network client library	
<code>derbyclient.jar</code>	JDBC driver
Locale libraries	
<code>derbyLocale_cs.jar</code>	provides translated messages for the Czech locale
<code>derbyLocale_de_DE.jar</code>	provides translated messages for the German locale
...	...
<code>derbyLocale_zh_TW.jar</code>	provides translated messages for the Traditional Chinese locale

# Available Drivers

---

```
org.apache.derby.jdbc.EmbeddedDriver
```

A driver for embedded environments, when Derby runs in the same JVM as the application.

```
org.apache.derby.jdbc.ClientDriver
```

A driver for the Network Server environment. The Network Server sets up a client/server environment.

# Database Connection URL

---

- For the embedded driver

```
jdbc:derby:databaseName[;URLAttributes]
```

where

- *databaseName*

The name of the database that you want to connect to

- *URLAttributes*

One or more of the supported attributes of the database connection URL, such as *;territory=ll\_CC* or *;create=true*.

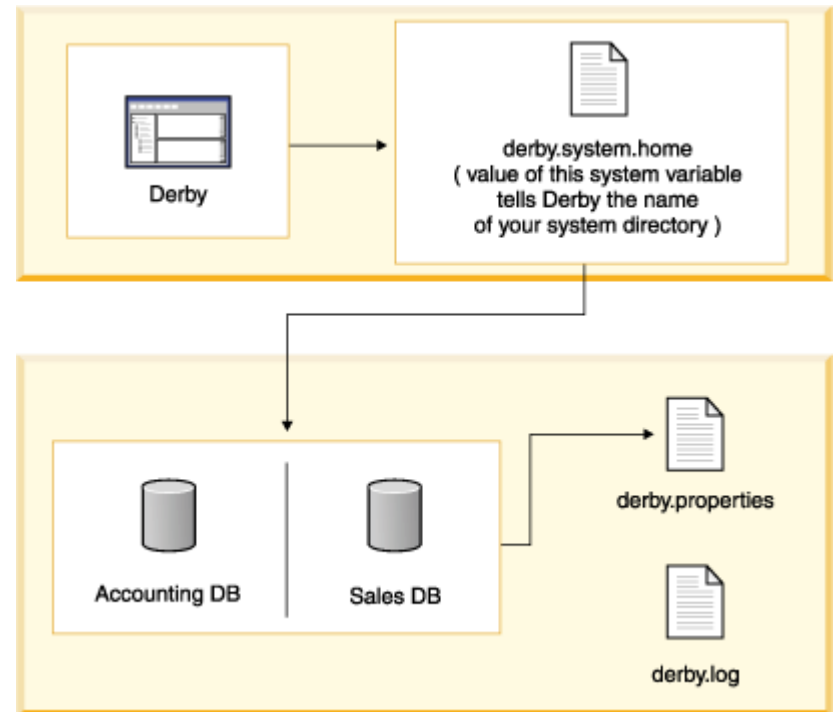
- For the network client driver

```
jdbc:derby://server[:port]/databaseName[;URLAttributes]
```

Default port: 1527

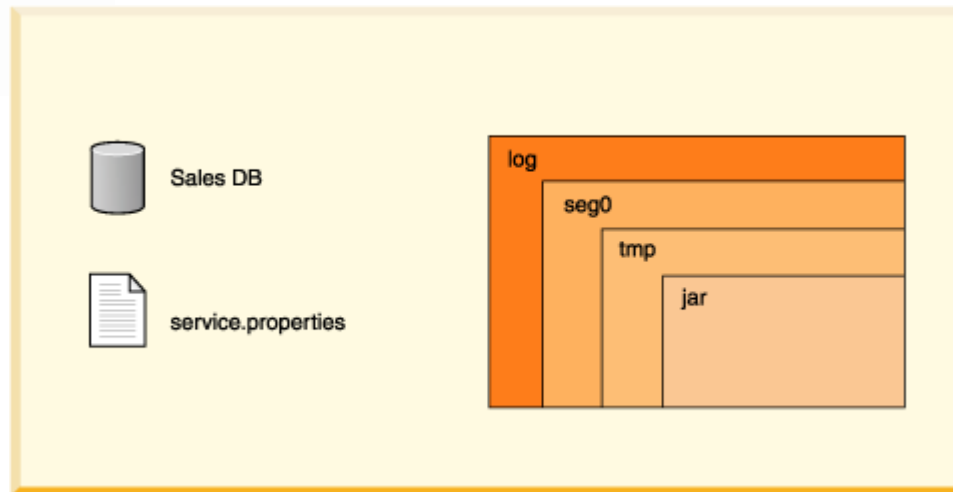
# Derby System

- A Derby database exists within a *system*
- A Derby system is a single instance of the Derby database engine and the environment in which it runs
  - a system directory
    - You define the system directory by specifying *derby.system.home*
    - If you do not specify the system directory, the current directory becomes the system directory
    - Contains a system-wide configuration *derby.properties* and error log *derby.log*
  - zero or more databases
    - A Derby database is stored in files that live in a directory of the same name as the database



# Database Directory

- A Derby database is stored in files that live in a directory of the same name as the database.
- Database directories typically live in *system* directories





# Ij Basics

- `ij` is an interactive SQL scripting tool that comes with Derby.
- It can be used with the Derby Embedded JDBC driver or with a client JDBC driver, such as the Derby Network Client.

- Startup `ij`

```
java -jar %DERBY_LIB%\derbyrun.jar ij
```

- Create a database

```
ij> connect 'jdbc:derby:MyDbTest;create=true';
```

- Connect to a database

```
ij> connect 'jdbc:derby:MyDbTest';
```

- Execute SQL statements

```
ij> create table derbyDB(num int, addr varchar(40));
```

```
ij> insert into derbyDB values (1956, 'Webster St.');
```

```
ij> insert into derbyDB values (1910, 'Union St.');
```

```
ij> update derbyDB set num=180, addr='Grand Ave.' where num=1956;
```

```
ij> select * from derbyDb;
```

- Disconnect from a database

```
ij> disconnect;
```

- Exit

```
ij> exit;
```

- Run SQL scripts

```
ij> run 'mySQLscripts.sql';
```

Embedded mode:

- Database in the current directory

**MyDbTest**

- Database with the directory path:

**/databases/MyDbTest**

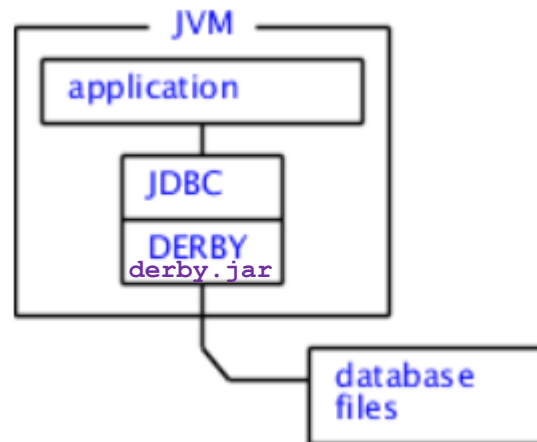
Network server mode:

**//hostname:1527/MyDbTest**

# Embedded Derby

- To use Derby in embedded mode, you only need one library:

`derby.jar`: *Derby engine and Derby Embedded JDBC driver*



**Derby Embedded Architecture**

# Derby Network Server

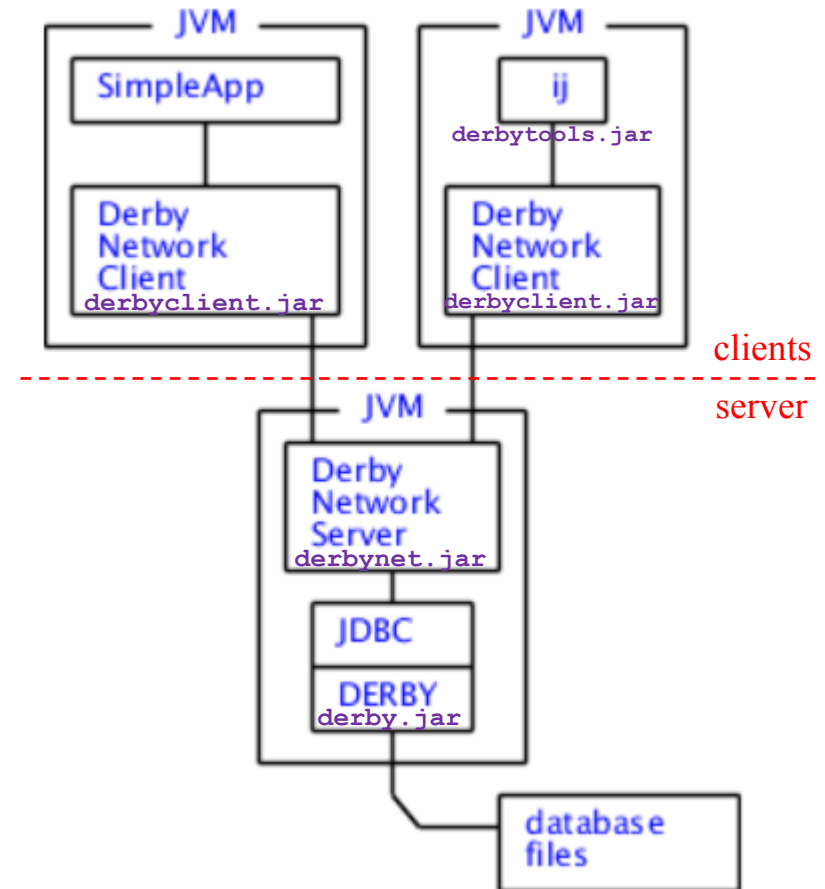
- To run the Derby Network server, you need following files

- On the server side:

<b>derby.jar</b>	Derby engine
<b>derbynet.jar</b>	Derby Network Server

- On the client side:

<b>derbyclient.jar</b>	JDBC driver
------------------------	-------------



Derby Network Server Architecture

# Derby Network Server

---

- Start Network server

- By default, listens on the localhost on port 1527

```
java -jar %DERBY_LIB%\derbyrun.jar server start
```

- Listen on the host machine

```
java -jar %DERBY_LIB%\derbyrun.jar server start -h hostname
```

- Listen on all network interfaces

```
java -jar %DERBY_LIB%\derbyrun.jar server start -h 0.0.0.0
```

- Listen on a specified port

```
java -jar %DERBY_LIB%\derbyrun.jar server start -p port#
```

- Stop Network server

- On the server from another command terminal

```
java -jar %DERBY_LIB%\derbyrun.jar server shutdown
```

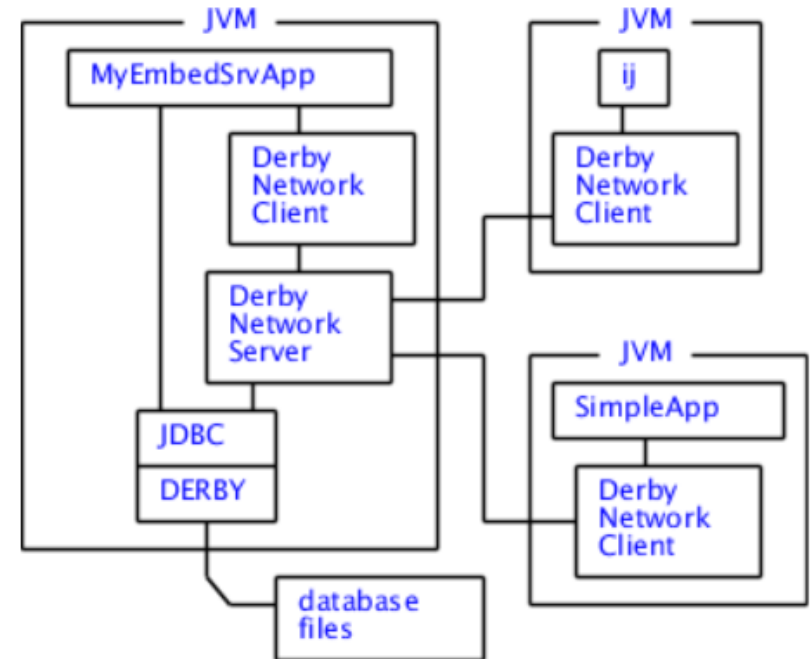
- On the client

```
java -jar %DERBY_LIB%\derbyrun.jar server shutdown -h hostname
```

# Derby Network Server Options

- The Derby Network Server can be run in either of these configurations:
  - **Stand-alone server**  
An independent Java process embedding the Derby database engine
  - **Embedded server**  
Embedded within another Java application, and both the Network Server framework and the Derby database engine are loaded by the Java application.

The Java application that embeds Derby can use either Derby Network Client JDBC driver or the embedded driver; former is preferred for less overhead not to going out through network



**Derby Embedded Server Architecture**

# derby.properties

---

- The file **derby.properties** contains the definition of properties, or configuration parameters
- The **derby.properties** file is not automatically created
- Setting system-wide properties programmatically
  - Setting properties programmatically works only for the application that starts up Derby; **not written to** the **derby.properties** file
  - As a parameter to the JVM command line

```
java -Dderby.system.home=C:\home\Derby\ MyJDBCApp
```
  - Using a Properties object within an application or statement

```
Properties p = System.getProperties();
p.setProperty("derby.system.home", "C:\home\Derby\");
```
- Changing the system-wide properties by using the **derby.properties** file
  - Properties set this way are **persistent** for the system until changed, until the file is removed from the system, or until the system is booted in some other directory (in which case Derby would be looking for **derby.properties** in that new directory)
  - The following is the text of a sample properties file:

```
derby.system.home = C:\home\Derby\
```

# Derby Security

---

- User authentication
  - Derby verifies user names and passwords before permitting them access to the Derby system.  
User authentication is disabled by default
- User authorization
  - A means of granting specific users permission to read a database or to write to a database.  
The default access setting is fullAccess.
- Disk encryption
  - A means of encrypting Derby data stored on disk.  
Attributes on the connection URL:  
`dataEncryption=true; bootPassword=blahblah`
- Validation of certificates for signed jar files
  - Derby validates certificates for classes loaded from signed jar files.
- Network encryption and authentication
  - Derby network traffic may be encrypted with SSL/TLS. SSL/TLS certificate authentication is also supported.

# Resources

---

- **Java DB Technical Documentation**

<http://docs.oracle.com/javadb/>

- Getting Started with Java DB
- Java DB Reference Manual
- Java DB Developer's Guide
- Tuning Java DB
- Java DB Server and Administration Guide
- Java DB Tools and Utilities Guide
- [Java DB API Documentation](#)

- **Apache Derby**

<http://db.apache.org/derby/>