

[vertical list of authors]

© Copyright ,.

[cover art/text goes here]

Contents

Copyright

Apache Software FoundationGetting Started with DerbyApache Derby

Copyright

Copyright 1997, 2006 The Apache Software Foundation or its licensors, as applicable.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Introducing Derby

Welcome to Derby! Derby is a relational database management system (RDBMS) that is based on Java™ and SQL. This section describes Derby.

Deployment options

You can deploy Derby in a number of different ways:

- Embedded in a single-user Java application. Derby can be almost invisible to the end user because it requires no administration and runs in the same Java virtual machine (JVM) as the application.
- Embedded in a multiuser application such as a web server, an application server, or a shared development environment.
- Embedded in a server framework. You can use the Network Server with the Network Client driver or a server of your own choice.

System requirements

Derby is a database engine written completely in Java; it will run in any certified Java Virtual Machine (JVM).

The Derby library

Derby Developer's Guide

Describes the functionality and features of Derby common to all deployments, such as Derby's JDBC and SQL specifics, deploying Derby applications, security, and other advanced features.

Derby Reference Manual

Documents Derby's implementation of the SQL language. Also provides reference information for Derby's JDBC and JTA implementations, keywords, system tables, properties, and *SQLExceptions*.

Tuning Derby

Explains how to configure and tune Derby through properties and provides reference information on properties. It also offers performance tips, an in-depth discussion of performance, and information about the Derby optimizer.

Derby Tools and Utilities Guide

A guide for working with the Derby tools such as *i j*, and more advanced utilities such as import/export and the database class loader.

Derby Server and Administration Guide

Part One of this guide discusses configuration of servers, how to program client applications, and database administration.

In addition, some systems might require administrative tasks such as backing up databases. These tasks are independent of any server framework but are unique to multi-user or large systems.

Part Two of this guide discusses administrative issues such as backups and debugging deadlocks.

Derby API javadoc

Automatically generated for all public Derby classes. (No javadoc is provided for the JDBC API, which is part of the Java™ 2 Platform, Standard Edition) For more

information about the classes in the API, see the *Derby Reference Manual* .

Installing and working with Derby

If you are new to Derby and JDBC programming, the following topics will help you begin using the product. If you are an experienced JDBC programmer, see [Quick start guide for experienced JDBC users](#).

Installing Derby

In order to install Derby you must first download the Derby zip or tar archive from the Derby web site at http://db.apache.org/derby/derby_downloads.html. This downloads page includes provides several distributions of Derby, including snapshot packages as well as stable-version binary distributions. You will also find information on accessing the active source distributions for Derby. The distributions come in *bin*, *lib* and *src* packages. This guide assumes that you have downloaded the *bin* distribution package.

Extract the downloaded package. The extracted installation contains several subdirectories:

- The `demo` subdirectory contains demonstration programs.
- The `frameworks` subdirectory contains scripts for executing utilities and setting up the environment.
- The `javadoc` subdirectory contains the `api` documentation that was generated from source code comments.
- The `doc` subdirectory contains the Derby documentation.
- The `lib` subdirectory contains the Derby jar files.

Setting up your Java environment

You need to set your PATH environment variable so the JVM and Java applications run correctly. This is extremely important for the success of your installation because the PATH variable enables your operating system to find the appropriate programs from any directory. If you have more than one JVM installed, the JVM you want to use must appear before any of the others in the PATH variable.

To check the PATH environment variable:

1. In a command window, type the following:

```
java -version
```

If the path is set correctly, you will see a printout telling you the version of your JVM.

2. If the command does not return the correct version of your JVM, set the PATH variable by adding the `bin` subdirectory of the JVM directory to the beginning of the path.

For example, if the directory is `C:\JDK1.4`, add `C:\JDK1.4\bin` to the beginning of the path.

3. Repeat step 1 to make sure the PATH environment variable is set correctly.

Using the tools and startup utilities

Derby tools include `dblook`, `ij`, and `sysinfo`. Derby utilities include the import, export and database class utilities. The `/frameworks/embedded/bin` directory contains scripts for running some of the Derby tools and utilities in embedded mode. You can find similar scripts for running tools and utilities for the Network Server in the `/frameworks/NetworkServer/bin` directory. In discussion of these tools and utilities in embedded mode, the `/frameworks/embedded/bin` directory is referred to as the `/bin`

directory unless stated otherwise.

The scripts have descriptive names such as `sysinfo.bat` or `ij.ksh`. Like the classpath scripts, these scripts end with different extensions depending on your environment. Windows scripts have a `.bat` extension, while Unix scripts have a `.ksh` extension. You might need to modify these scripts in order for the tools and utilities to run properly on Windows and Unix platforms.

The following information applies *only* to UNIX platforms.

In order to use the script files to run the tools and utilities, do the following:

1. Turn on the execute bit for the script files. For example:

```
chmod +x fileName
```

where *fileName* is any script file with a `.ksh` extension. This tells the system to execute the commands in the script every time you use it.

2. Add the `/bin` directory to your `PATH` variable. This allows you to use shortened commands to start the Derby tools.

For more information on scripts, see [Scripts provided by Derby](#).

Using sysinfo

Derby's `sysinfo` tool displays information about your Java environment and your version of Derby. The `sysinfo` script sets the appropriate environment variables, including the classpath, and executes the `sysinfo` program. Once you have the `/bin` directory in your `PATH`, run `sysinfo` by entering the following in a command window:

```
sysinfo
```

Running ij

You can use the `ij` tool to connect to a Derby database. You must include the `/bin` directory in your `PATH` environment variable to run `ij`.

- You can run `ij` by typing the following command:

```
ij
```

The `ij` script executes the `ij` program and sets up environment variables like `CLASSPATH`.

- To create a database with `ij`, type the following command:

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

This command creates a database called `testdb` in the current directory, populates the system tables, and connects to the database. You can then execute any SQL statements from the `ij` command line.

- When you are ready to leave `ij`, type:

```
ij> exit;
```

See the *Derby Tools and Utilities Guide* for more information on running `ij`.

Manually setting environment variables and paths

If you cannot run the scripts for the Derby tools and utilities, you must complete certain steps manually. The following topics show how to set your environment and run the tools manually.

Set the DERBY_INSTALL environment variable

During installation, a base directory was created where the software was installed. This document assumes that directory is named Derby_10. This document refers to that directory as the Derby base directory.

If you plan to use the scripts in the `/frameworks/embedded/bin` directory and if your operating system supports it, create an environment variable called `DERBY_INSTALL` and set its value to the path of the Derby base directory. This will allow you to use shortened commands to run the Derby tools and utilities.

For example, if you installed the product in `c:\Derby_10`, set `DERBY_INSTALL` to `c:\Derby_10`:

```
set DERBY_INSTALL=c:\Derby_10
```

Set the classpath

The JVM needs to know the path of every class file needed by your application. The classpath is a list of the class libraries needed by the JVM and other Java applications to run your program. You can set the `CLASSPATH` environment variable in the operating system either temporarily, permanently, or at run time when you start your Java application and the JVM. If you set the environment variable temporarily, you must set it each time that you open a new command window.

In most development environments, it is best to set the operating system's `CLASSPATH` environment variable temporarily. Derby provides some scripts to help you set your classpath this way; they are found in the `/frameworks/embedded/bin` directory or the `/frameworks/NetworkServer/bin` directory. To set the class path temporarily, run a script every time you open a new command window.

- Set your `CLASSPATH` so that it includes the `derby.jar` and `derbytools.jar` files.

For example:

```
set CLASSPATH=%DERBY_INSTALL%\lib\derby.jar;  
%DERBY_INSTALL%\lib\derbytools.jar;%CLASSPATH%
```

- To manually run the `sysinfo` utility, enter the following in a command window or shell:

```
java org.apache.derby.tools.sysinfo
```

- To manually run the `ij` utility and connect to the database:
 - a. Enter the following in a command window or shell:

```
java org.apache.derby.tools.ij  
ij> connect 'jdbc:derby:<dbname>;create=true';
```

where `<dbname>` is the name of the database you are connecting to.

- b. To exit the `ij` utility, enter:

```
ij> exit;
```

For more information on running the ij and sysinfo utilities, see the *Derby Tools and Utilities Guide*

Quick start guide for experienced JDBC users

This chapter is for experienced JDBC programmers who already know how to set the classpath, how to run a Java program, and how to use a JDBC driver.

For more detailed information on the topics covered in this section, see the *Derby Developer's Guide* and the *Derby Tools and Utilities Guide*.

Environments in which Derby can run

Before you configure your system to run Derby, it is useful to understand something about the different environments in which Derby can run, because these environments affect the classpath, driver name, and database connection URL.

See the *Derby Developer's Guide* for more information on Derby environments.

Embedded environment

An environment in which only a single application can access a database at one time, and no network access occurs. When an application starts an instance of Derby within its JVM, the application runs in an embedded environment. Loading the embedded driver starts Derby.

Client/server environment

An environment in which multiple applications connect to Derby over the network. These applications run in a client/server environment. Derby runs embedded in a server framework that allows multiple network connections. (The framework itself starts an instance of Derby and runs in an embedded environment. However, the client applications do not run in the embedded environment.)

You can also embed Derby in any Java server framework.

See the *Derby Server and Administration Guide* for more information on how to run Derby on a server.

Libraries and classpath

See [Derby libraries and scripts: complete reference](#) in the appendix for a complete description of the Derby libraries.

Available drivers

The following JDBC drivers are available depending on the environment you choose for Derby:

- `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 Derby-provided embedded driver, the format for the database connection URL for connecting to a database is this:

```
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 *;locale=ll_CC* or *;create=true*.

For more information, see the *Derby Developer's Guide* .

For the Derby-provided network client driver, the format for the database connection URL for connecting to a database is this:

```
jdbc:derby://<server>[:<port>]/  
databaseName[;URLAttributes=<value> [;...]]
```

where the <server> and <port> specify the host name (or IP address) and port number where the server is listening for requests. The *URLAttributes* can be either Derby embedded or network client attributes. See the *Derby Server and Administration Guide* for more information on accessing the Network Server by using the network client.

Documentation conventions

Terminology, syntax, and typographical conventions of the Derby documentation.

Terminology

The Derby documentation uses the specialized term *environment* to describe the method your application uses to interact with Derby. An environment is sometimes referred to as a *framework*. The two types of environments are the *embedded environment* and the *client/server environment*.

SQL syntax

SQL syntax is presented in modified BNF notation. The meta-symbols of BNF are:

Symbol	Meaning
	“or.” Choose one of the items
[]	Enclose optional items.
*	Flags items that you can repeat 0 or more times. Has a special meaning in some SQL statements.
{ }	Groups items so that they can be marked with one of the other symbols, i.e. [], , or *.
() . ,	Other punctuation that is part of the syntax.

An example of how SQL syntax is presented:

```
CREATE [ UNIQUE ] INDEX
  IndexName
  ON
  TableName
  (
    SimpleColumnName
    [ ,
      SimpleColumnName
    ] * )
```

Command-line syntax for running Java programs and utilities (as well as examples) always begins with the word *java*:

```
java org.apache.derby.tools.ij
```

In addition, this documentation uses the IBM Software Development Kit style for setting JVM arguments and properties. If you use another Java Virtual Machine, the way you set JVM arguments and properties might be different.

Typographical conventions

This documentation uses some typographical conventions to highlight elements of the SQL language, operating system commands, and the Java programming language.

Usage	Typeface	Examples
New terms	Italic	defined by <i>keys</i>

Usage	Typeface	Examples
File and directory names	Italic	<i>C:\derby</i>
Dictionary objects	Italic	The <i>Employees</i> table
In syntax, items that you do not type exactly as they appear, but replace with the appropriate name	Italic	CREATE TABLE <i>tableName</i>
SQL examples	Bold and/or fixed-width	SELECT city_name FROM Cities
Java application examples	Bold and/or fixed-width	Connection conn = DriverManager.getConnection ("jdbc:derby:Sample")
Things you type in a command prompt	Bold and/or fixed-width	java org.apache.derby.tools.ij
Comments within examples	Bold and/or fixed-width	--This line ignored
SQL keywords (commands)	All caps	You can use a CREATE TABLE statement

Derby libraries and scripts: complete reference

This appendix describes Derby libraries and scripts.

Libraries provided by Derby

Library name	Use
Engine libraries You always need this library for embedded environments. For client/server environments, you only need this library on the server.	
derby.jar	For embedded databases
Tools libraries For embedded environments, you need a library in the classpath to use a tool. For a client/server environment, you need a library on the client only.	
derbytools.jar	Required for running all the Derby tools (such as ij, dblook, and import/export).
The Network Server libraries	
derbynet.jar	Required to start the Derby Network Server.
The network client libraries	
derbyclient.jar	Required to use the Derby network client driver.
The locale libraries	
<ul style="list-style-type: none"> derbyLocale_es.jar derbyLocale_de_DE.jar derbyLocale_fr.jar derbyLocale_it.jar derbyLocale_ja_JP.jar derbyLocale_ko_KR.jar derbyLocale_pt_BR.jar derbyLocale_zh_CN.jar derbyLocale_zh_TW.jar 	Required to provide translated messages for the indicated locales.

Libraries not provided by Derby

In a Java Development Kit Version 1.3 environment, some special Derby features require that you install additional libraries and place them in your class path (Environments using Java Development Kit, Version 1.4 includes all of these libraries):

- LDAP (see the *Derby Developer's Guide*)
- JTA (see the *Derby Reference Manual*)
- JDBC 2.0 Extensions (see the *Derby Reference Manual*)

Scripts provided by Derby

Derby provides scripts in the `/bin` directory. Each script comes in two flavors, one ending in `.bat` and one ending in `.ksh`. Here is a complete listing:

- `frameworks/embedded/bin/dblook`

Runs dblook.

- *frameworks/embedded/bin/ij*
Starts ij.
- *frameworks/embedded/bin/setEmbeddedCP*
Puts all the Derby libraries for an embedded environment in the classpath.
- *frameworks/embedded/bin/sysinfo*
Runs sysinfo.
- *frameworks/NetworkServer/bin/dblook*
Runs dblook in a Network Server client context.
- *frameworks/NetworkServer/bin/ij*
Runs ij in a Network Server client context.
- *frameworks/NetworkServer/bin/NetworkServerControl*
Runs NetworkServerControl.
- *frameworks/NetworkServer/bin/setNetworkClientCP*
Puts the libraries needed to connect to Derby Network Server into the classpath.
- *frameworks/NetworkServer/bin/setNetworkServerCP*
Puts the libraries needed for Derby Network Server into the classpath.
- *frameworks/NetworkServer/bin/startNetworkServer*
Starts the Network Server on the local machine.
- *frameworks/NetworkServer/bin/stopNetworkServer*
Stops the Network Server on the local machine.
- *frameworks/NetworkServer/bin/sysinfo*
Gets the system information from a running Derby Network Server.

Trademarks

The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the documents in the Apache Derby documentation library:

Cloudscape, DB2, DB2 Universal Database, DRDA, and IBM are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.