

Getting Started with Derby

Version 10.2

Derby Document build:
October 2, 2006, 8:01:20 AM (PDT)

Contents

Copyright.....	3
Introducing Derby.....	4
Deployment options.....	4
System requirements.....	4
The Derby library.....	4
Installing and working with Derby.....	6
Installing Derby.....	6
Setting up your environment.....	6
Using the Derby tools and startup utilities.....	7
Using the sysinfo tool.....	8
Running ij.....	8
Setting the classpath.....	9
Quick start guide for experienced JDBC users.....	10
Environments in which Derby can run.....	10
Embedded environment.....	10
Client/server environment.....	10
Available drivers.....	10
Database connection URL.....	10
Documentation conventions.....	12
Terminology.....	12
SQL syntax.....	12
Typographical conventions.....	12
Derby libraries and scripts: complete reference	14
Libraries provided by Derby.....	14
Libraries not provided by Derby.....	14
Scripts included with Derby.....	14
Trademarks.....	16

Copyright

Apache Software FoundationGetting Started with DerbyApache Derby

Copyright

Copyright 2004, 2006 The Apache Software Foundation or its licensors, as applicable.

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you 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.

Your deployment options include:

- 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

The Derby library includes the Derby manuals and API reference.

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 *ij*, 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, this section will help you begin using the product.

If you are an experienced JDBC programmer, see [Quick start guide for experienced JDBC users](#).

Installing Derby

To install Derby you must first download the Derby zip or tar archive from the Derby Web site.

Navigate your Web browser to http://db.apache.org/derby/derby_downloads.html. This downloads page provides several distributions of Derby, including prerelease snapshots as well as stable-version binary distributions. You will also find information on how to use the Subversion version control system to access the Derby source code that is currently in development.

The distributions are:

- The *bin* distribution contains scripts, demonstration programs, and documentation. The optimized jar files are available in the *lib* distribution.
- The *lib* distribution contains an optimized, small footprint set of the Derby jar files for deployment.
- The *lib-debug* distribution contains a larger footprint set of the Derby jar files that are useful for debugging or reporting issues.
- The *src* distribution contains the source files that are used to create the *bin*, *lib*, and *lib-debug* distributions.

This guide assumes that you have downloaded the *bin* distribution.

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

- The `demo` subdirectory contains the demonstration programs.
- The `bin` subdirectory contains the scripts for executing utilities and setting up the environment.
- The `javadoc` subdirectory contains the `api` documentation that was generated from source code comments.
- The `docs` subdirectory contains the Derby documentation.
- The `lib` subdirectory contains the Derby jar files.
- The `test` subdirectory contains regression tests for Derby.
- The `frameworks` subdirectory contains older style scripts for executing utilities and setting up the environment. These are provided in this release for backward compatibility. In a next release, they will be deprecated in favor of the scripts in the `bin` directory.

Setting up your environment

You need to set several environment variables so that the scripts that are included with the Derby `bin` distribution run correctly. These scripts need to be run in an appropriate command line environment, i.e. a "Command Prompt" on Windows or a shell on Unix.

To set the environment variables:

1. Set the `DERBY_HOME` environment variable to the location where you extracted the Derby `bin` distribution.

For example, if you installed Derby in the `c:\Derby_10` directory on Windows, or the

/opt/Derby_10 directory on Unix, use the following command to set the DERBY_HOME environment variable:

Operating System	Command
UNIX	<code>export DERBY_HOME=/opt/Derby_10</code>
Windows	<code>set DERBY_HOME=c:\Derby_10</code>

- Set the JAVA_HOME environment variable. The JAVA_HOME environment variable is used by the scripts to locate the JVM and Java applications.

Operating System	Command
UNIX	<code>export JAVA_HOME=/usr/j2se</code>
Windows	<code>set JAVA_HOME=C:\Program Files\Java\j2se1.4.2_05</code>

- Add the DERBY_HOME/bin directory to the PATH environment variable so that you can run the Derby scripts from any directory.

Operating System	Command
UNIX	<code>export PATH="\$DERBY_HOME/bin:\$PATH"</code>
Windows	<code>set PATH=%DERBY_HOME%\bin;%PATH%</code>

When the DERBY_HOME environment variable is set and the underlying bin directory is included in the PATH environment variable, you can use shortened commands to start the Derby tools.

- Verify that the DERBY_HOME and the JAVA_HOME environment variables are set correctly, and that the DERBY_HOME/bin directory is added to the PATH environment variable.

Use the following command to check the settings:

```
sysinfo
```

If the environment variables are set correctly, the sysinfo command displays information about your JVM and the version of Derby you have installed.

For more information on the scripts included in the bin distribution, see [Scripts included with Derby](#).

Using the Derby tools and startup utilities

There are several tools and utilities that you might want to use as you begin to work with Derby.

The tools that are included with Derby are dblook, ij, and sysinfo. You can use the Derby scripts to start the Derby tools.

There are Derby utilities that are system procedures that you can call by using the ij tool. For example, there are system procedures that you can use to import and export external files. Instructions on how to use these system procedures are included in the *Derby Tools and Utilities Guide* and the *Derby Reference Manual*.

The scripts that run these tools are in the `DERBY_HOME/bin` directory. These scripts also help you set up your classpath.

The scripts in `DERBY_HOME/bin` directory are setup to run in the embedded mode. To run these tools with the Network Server, use the following commands:

- For the `dblook` tool, call the script and specify the `-d` option and the full URL to the Network Server database. For example: `dblook -d 'jdbc:derby://localhost/myDB;user=usr'`
- For the `ij` tool, issue the command `set DERBY_OPTS=-Dij.protocol=jdbc:derby://localhost/` and then start `ij` by issuing the command `ij`.
- For the `sysinfo` tool, issue the command `NetworkServerControl sysinfo`.

To show how to use the Derby scripts to launch the Derby tools, the examples in the *Getting Started with Derby* assume that you are using the embedded mode of the Derby database engine.

Using the sysinfo tool

The `Derbysysinfo` tool displays information about your Java environment and your version of Derby.

After you add the `DERBY_HOME/bin` directory to your `PATH` environment variable, run the `sysinfo` tool. The `sysinfo` script sets the appropriate environment variables, including the classpath, and runs the `sysinfo` tool. To run the `sysinfo` tool, issue the following command in a command window:

```
sysinfo
```

Running ij

You can use the `ij` tool to connect to a Derby database.

You must add the `DERBY_HOME/bin` directory in your `PATH` environment variable before you can run the `ij` tool.

- To start the `ij` tool, run the `ij` script from the command line.

For example:

```
ij
```

The `ij` script starts the `ij` tool and sets up the environment variables like `CLASSPATH`.

- To create a database with the `ij` tool, 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 run any SQL statements from the `ij` command line.

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

```
ij> exit;
```

See the *Derby Tools and Utilities Guide* for more information about ij.

Setting the classpath

The classpath is a list of the class libraries that are needed by the JVM and other Java applications to run your program. The scripts that are included with Derby set up their own classpath for running the tools. However, to call the tools directly using Java and not using the scripts, you need to manually set the CLASSPATH environment variable.

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.

In most development environments, it is best to temporarily set the CLASSPATH environment variable in the command line shell where you are entering commands. Derby provides several scripts in the `DERBY_HOME/bin` directory to help you set your classpath quickly. These scripts are:

setEmbeddedCP

The `setEmbeddedCP` script adds the `derby.jar` and `derbytools.jar` files to the classpath, when the database engine is used in embedded mode.

setNetworkServerCP

The `setNetworkServerCP` script adds the `derby.jar` and `derbynet.jar` files to the classpath, so that you can start the network server.

setNetworkClientCP

The `setNetworkClientCP` script adds the `derbyclient.jar` and `derbytools.jar` files to the classpath, so that you can access databases using the network client.

To set the classpath temporarily, run the script that is appropriate for your environment every time that you open a new command window.

To see the classpath that the script sets, issue the following command after you run the script:

- On UNIX, use `echo $CLASSPATH`
- On Windows, use `echo %CLASSPATH%`

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 section 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 embedded environment is 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

A client/server environment is an environment in which multiple applications connect to Derby over the network.

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.

Available drivers

Different 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

You must use a database connection URL when using the Derby-provided embedded

driver to connect to a database.

The format for the database connection URL for connecting to a database is:

```
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 and *databaseName* is the name of the database you want to connect to. 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

This section describes the 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
```

This documentation attempts to be JVM neutral, but any command line examples or syntax using JVM specific references should be verified with your JVM documentation.

Typographical conventions

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

Table 1. Derby typographical conventions

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

This section shows the different libraries used by Derby and their function.

Table 2. Derby libraries and their use

Library name	Use
Engine library 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).
derbyrun.jar	Executable jar file that can be used to start the Derby tools.
The Network Server library	
derbynet.jar	Required to start the Derby Network Server.
The network client library	
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 (Java Development Kit, Version 1.4 already 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 included with Derby

The Derby scripts are located in the `/bin` directory.

The complete list of scripts that are included with Derby are:

- *bin/dblook*
Runs the dblook tool.
- *bin/ij*
Starts the ij tool.
- *bin/NetworkServerControl*
Runs NetworkServerControl.
- *bin/setEmbeddedCP*
Puts all of the Derby libraries for an embedded environment in the classpath.
- *bin/setNetworkClientCP*
Puts the libraries needed to connect to the Derby Network Server into the classpath.
- *bin/setNetworkServerCP*
Puts the libraries needed to start the Derby Network Server into the classpath.
- *bin/startNetworkServer*
Starts the Network Server on the local machine.
- *bin/stopNetworkServer*
Stops the Network Server on the local machine.
- *bin/sysinfo*
Runs the sysinfo tool.

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.