HowItWorks

How Derby Works

- How Derby Works
  - Where to Look
  - General
    - Common Development Tasks
  - JDBC
  - Network Server
  - SQL
  - Store
  - Services

Where to Look

Information on Derby's internals is scattered over a number of locations:

1. The code, describes exactly how it works.
2. Comments in the code, usually match the code, and hopefully will add value as to why the code is as it is, rather than just describing the code exactly. The complete javadoc of all the classes, methods & fields is generated nightly. If more comments could be added then please submit a patch.
3. Discussions in the mail list archives for derby-dev at db.apache.org. The web-site has a collection of useful links to past discussions.
4. Papers and presentations on web site.
5. Functional and design specs attached to Jira issues, encouraged when a feature is implemented.
6. Wiki, please add as much information as you can.

The page attempts to provides links to all the information about how Derby works.

General

| Architecture Overview | Version Numbering Scheme | Internals of Derby | JavaPackageConventions |

Also the presentations (click presentations in the left tab) on the web-site's papers page provide a useful overview.

Common Development Tasks

- How to add a new error message

JDBC

Derby supports an embedded type 4 JDBC driver and a client type 4 JDBC driver. The goal is for the two drivers to have the same level of api support and identical behaviour.

Network Server

Derby's network server adds the traditional client/server mode to the emebedded database engine. The supported protocol is DRDA over TCP/IP using Java sockets.

SQL

Derby's SQL layer sits below the embedded JDBC driver and above the store. It handles parsing, compilation and execution of SQL.

Store

Derby's store layer sits below the SQL layer and above the Java i/o filesystem, though the file system layer is abstracted allowing alternate storage such as Jar files.

Services

Derby's services are generic areas of code that are not specific to JDBC, SQL or the store layer.