Apache Derby Status Update
OSCON 2005
“DIRBY” or “DAHRBY”?

It depends...
Introductions
What Is Derby?

• Apache's Open Source Database
• 100% Java Relational Database
• Easy to use
• Can run in same VM or as separate server
• Standards-based
  • SQL-92, JDBC 3.0, J2EE, J2ME JSR169
  • No Derby-specific extensions
• Secure
What Is Derby Good For?

• Great Java Developer Database
  • Simple, easy-to-use, standards-compliant

• Great embedded database
  • Can be made “invisible” to end users, no administrative overhead
  • Higher performance when embedded

• Good departmental database
  • Migration to Big Iron DBs easier because of standards compliance
Derby Configurations

- Can be embedded in same VM or client/server
- Depends on which JDBC driver you use

```
App
  ▼
  Derby
  ▼
  Disk
```

```
Network Server
  ▼
  Derby
  ▼
  Disk
```

App 2

App 2

App 3
High-End SQL Features

- Java Stored Procedures
- Triggers
- Two-phase commit (XA support)
- Updatable SQL Views
- Full Transaction Isolation Support
Brief History of Derby

• Founded in 1996 as JBMS, then Cloudscape
• Inspiration of some of the best database engineers from Sybase
• Intention was to take advantage of strengths of Java (portability, robustness, security) and apply them to database architecture
• Acquired by Informix in 1999, which was then acquired by IBM in 2001
More History

- IBM contributed Cloudscape to Apache in August 2004, renamed to Derby
  - Part of Apache DB project
- Community is growing
  - 30 active developers
  - Added four more committers last month
- Graduated from Apache incubation last week
  - Apache recognizes Derby as a viable project
- 10.1 released today as first release out of incubation
Why Sun Likes Derby

- Enables productive Java development with a good solid, easy-to-use Java database
- Great for use in our own products
  - Embeddable, solid Java product, free to redistribute under Apache license
- It has a strong architecture
  - Already is a very good developer and embedded database
  - Committed to working with community to make even better
- A good alternative for users who need a lightweight, free, easy-to-use 100% Java DB
How Can You Get Involved?

• Download Derby, see what you think
  • Let us know what's missing
  • Join derby-user@db.apache.org
• Get involved as a developer
  • Lots to do
  • You can actually get stuff committed
  • Easy to build, test
  • Strong community support
• http://db.apache.org/derby
What Next?

- Just finished the 10.1 release, what should we do next?
- Focus on embedded or client/server or J2ME?
- Application server/web server support?
- Disconnected support?
- High-end features (replication, clustering)?
Derby Architecture
Derby Architecture

- Module-based architecture
- A module is a discrete set of functionality (lock manager, indexing method, etc.)
- Module interface defined by a set of Java interfaces
- Which modules are loaded is defined at runtime
  - Allows for a very flexible architecture and control of what is loaded for a given configuration
Derby Architecture

• Four key areas: JDBC, SQL, Store and Services

```
+-----------------+    +-----------------+    +-----------------+
| Services        | →  | JDBC             | →  | SQL              |
|                 |    | Store            |    |                  |
```
JDBC

• This *is* the Derby API for applications
  • No Derby-specific API (ala Oracle OCI)
• JDBC 2.1, JDBC 3.0, JSR-169 (J2ME)
• JDBC 4 work in progress
SQL

- Parses, compiles, executes SQL statements
- SQL statements compiled into Java bytecode
  - Executed by running the Java interpreter
- Accesses data through the Store
Store

- Two main areas: *access* and *raw*
- Access layer provides table/row view of data
  - Handles table scans, index scans, index lookups, transactions, etc.
- Raw layer provides interfaces for raw access to data
  - Can provide different implementations, e.g. on disk, in memory, etc.
Services

• General services for other components
• Lock management, cache management, error logging, internationalization, daemon threads, etc.
Network Server

- Server engine that embeds Derby
- Allows client/server access via Derby's network client JDBC driver
- Communicates via DRDA protocol
Tools

• Very basic tool support
  • Interactive SQL
  • Backup/Restore
  • Import/Export

• Intend most tool support to come from third-party tools