

## DB2 and Java Application Development

DB2 Quickstart Education

Maintained by Paul Yip (ypaul@ca.ibm.com)

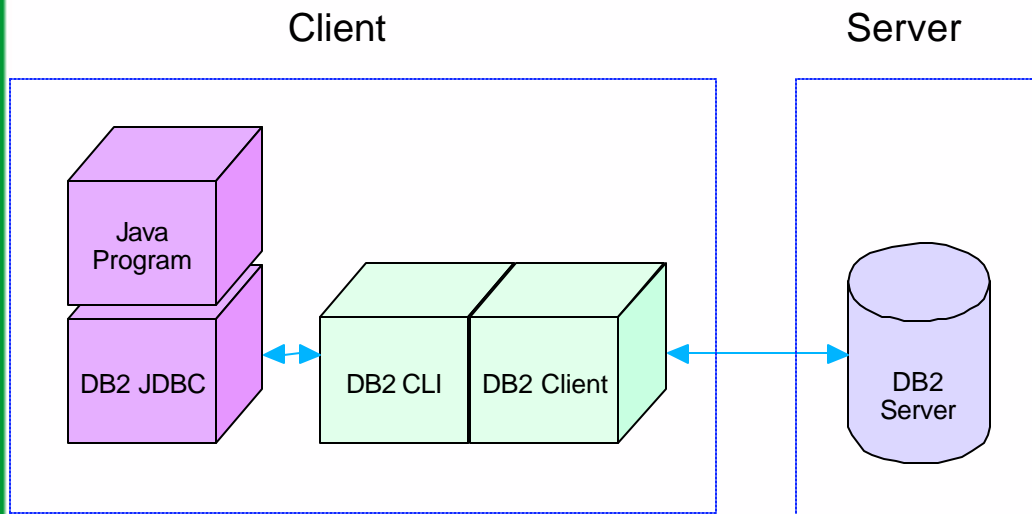
February 2003

IBM Software Group

### DB2 JDBC TYPE 2 Application Driver

- Driver class: COM.ibm.db2.jdbc.app.DB2Driver
- db2java.zip must be in the classpath
  - ▶ Windows: C:\.. \SQLLIB\java\db2java.zip
  - ▶ UNIX: <instance\_home>/sqllib/java/db2java.zip
  - ▶ Built on top of native DB2 Call Level Interface (CLI)
- DB2 Client must be installed at database client
  - ▶ Examples:
    - ─ App server in 3-tier
    - ─ Desktop in 2-tier

## DB2 JDBC Type 2 Driver Model



DB2 Data Management Software



## DB2 JDBC Type 2 Driver

### JDBC 2.1 Core API Restrictions

- ▶ Updatable ResultSets
- ▶ New SQL types (Array, Ref, Java Object, Struct)
- ▶ Customized SQL type mapping
- ▶ Scrollable sensitive ResultSets (scroll type of `ResultSet.TYPE_SCROLL_SENSITIVE`)

DB2 Data Management Software



## DB2 JDBC Type 2 Driver

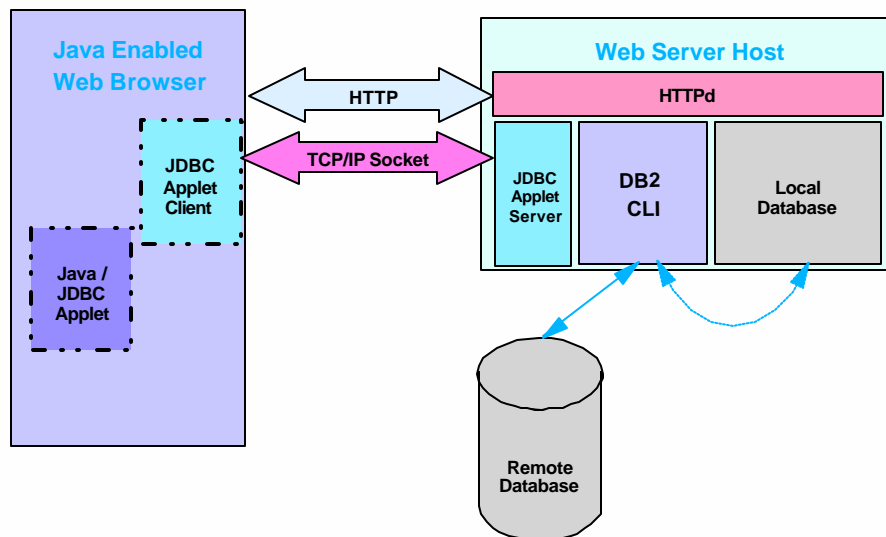
### JDBC 2.1 Optional Package API Support

- ▶ Java Naming and Directory Interface™ (JNDI) for Naming Databases
  - javax.naming.Context
  - javax.sql.DataSource
  - javax.naming.InitialContextFactory
- ▶ Connection Pooling
  - javax.sql.ConnectionPoolDataSource
  - javax.sql.PooledConnection
- ▶ Java Transaction APIs (JTA)
  - javax.sql.XAConnection
  - javax.sql.XADataSource
  - javax.transaction.xa.XAResource
  - javax.transaction.xa.Xid

## DB2 JDBC TYPE 3 Net Driver

- Type 3 Driver
- Driver class: COM.ibm.db2.jdbc.net.DB2Driver
- Layer on top of DB2 Call Level Interface
- No code required at client
- Deprecated in V8, migration to Type 4 driver is recommended
- Not as scalable - JDBC listener required

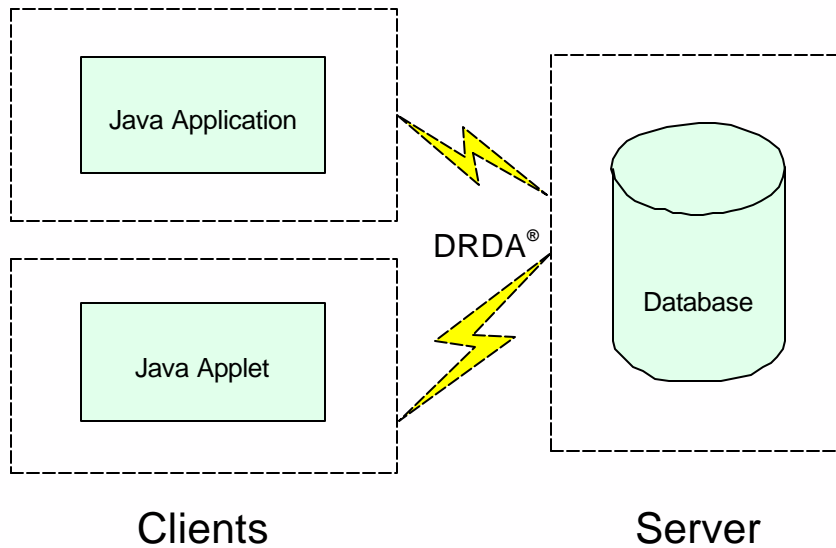
## DB2 JDBC Type 3 Driver Model



## DB2 JDBC Native Driver

- Type 4 Driver, [new in V8](#)
- Driver class: `com.ibm.db2.jcc.DB2Driver`
- `db2jcc.jar` must be in the classpath of the client
- New two-tier architecture, short code path, better performance and stability
- Java client communicates directly with DB2 server
- Supports both application and applet

## DB2 JDBC Type 4 Driver Model



## DB2 JDBC Type 4 Driver

### JDBC 2.1 Core API Restrictions

- ▶ Updatable ResultSets
- ▶ New SQL types (Array, Ref, Java Object, Struct)
- ▶ Customized SQL type mapping

## DB2 JDBC Type 4 Driver

### JDBC 2.1 Optional Package API Support

- ▶ Java Naming and Directory Interface (JNDI) for Naming Databases
  - `javax.sql.DataSource`

## Summary

- **TYPE 2** - "app" driver (`COM.ibm.db2.jdbc.app.DB2Driver`)
  - ▶ `db2java.zip`
  - ▶ J2EE 1.3 Certified & recommended for WebSphere Application Server
  - ▶ DB2 client required
  - ▶ example URL: `"jdbc:db2:sample"`
- **TYPE 3** - "net" driver (`COM.ibm.db2.jdbc.net.DB2Driver`)
  - ▶ `db2java.zip`
  - ▶ Requires listener process at server & DB2 Client not required
  - ▶ example URL: `"jdbc:db2://host.mydomain.com:5678/sample"`
- **TYPE 4** (`com.ibm.db2.jcc.DB2Driver`)
  - ▶ `db2jcc.jar`
  - ▶ Native DRDA
  - ▶ DB2 Client not required
  - ▶ example URL: `"jdbc:db2://host.mydomain.com:50000/sample"`

listener's port

instance port

## JDBC Performance Fundamentals

IBM Software Group

### JVM Heap Size

- Vital to performance
- Defaults are not sufficient
- Recommend to set both initial and maximum to an equal large value
- Minimize garbage collection scanning

## Table Spaces

- Database Managed Storage (DMS) vs System Managed Storage (SMS)
- Real Life Example: Java Batch Inserts
  - DMS
  - SMS with Multi-page File Allocation Enabled

## JDBC Programming Tips

- Making connections is expensive
  - Use Connection Pooling
- Committing transactions is expensive
  - Turn autocommit off
    - `Conn.setAutoCommit(false)`**
  - Commit frequent enough to free up locks



## JDBC Programming Tips

- Use of DatabaseMetaData methods is costly
  - ❌ `ResultSet rs=dbmd.getTables(null, null, "tablename", arrayTypes);`
  - ❌ `ResultSet rs=dbmd.getTables("SYSTABLES", "DB2ADMIN", "T1", arrayTypes);`
- VARCHAR(32K) FOR BIT DATA vs BLOB

## JDBC Programming Tips

- Reduce size of results
- Use host variables for literals in SQL statements
- Avoid data type conversion
  - ▶ Use `getInt`, `getDouble`, `getDecimal` vs `getString`
  - ▶ Use Date, Time and Timestamp objects vs Strings

## JDBC Programming Tips

- Close ResultSets
  - ▶ Garbage collection can reclaim the objects
- Close PreparedStatements
  - ▶ Release the underlying cursor resource
- Close CallableStatements
  - ▶ Recycle call sections.
- Close Connections

## DB2 Programming Tips

- FOR READ or FETCH ONLY clause
  - ▶ Less lock contention
  - ▶ Blocking
- FOR UPDATE clause
  - ▶ Avoid potential deadlocks
- Append On clause
  - ▶ May improve performance for large insert
  - ▶ Can not be used with clustered index



## Declared Global Temporary Tables

---

- No lock contention
- Minimal undo logging or NOT LOGGED
  - ▶ Minimal or no logging contention
- Index and Statistics support - [New in V8](#)
  - ▶ Better performance
- Session context
  - ▶ Using with Connection Pooling
  - ▶ WITH REPLACE