JDBC
JDBC Overview

Contents

The JDBC API is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases. The JDBC API provides a call-level API for SQL-based database access. JDBC technology allows you to use the Java programming language to exploit "Write Once, Run Anywhere" capabilities for applications that require access to enterprise data.

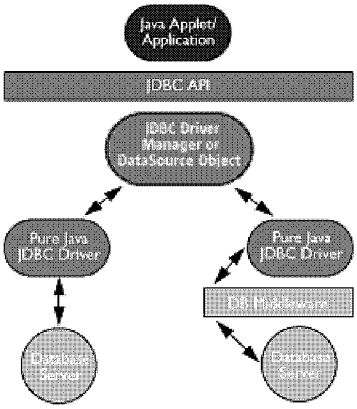
JDBC API Overview

The JDBC API makes it possible to do three things:

- Establish a connection with a database or access any tabular data source
- Send SQL statements
- Process the results

JDBC Architecture

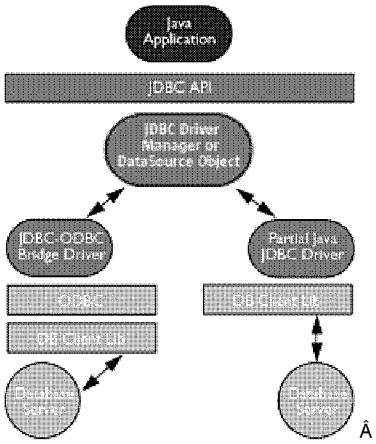
The JDBC API contains two major sets of interfaces: the first is the JDBC API for application writers, and the second is the lower-level JDBC driver API for driver writers. JDBC technology drivers fit into one of four categories. Applications and applets can access databases via the JDBC API using pure Java JDBC technology-based drivers, as shown in this figure:



Left side, Type 4: Direct-to-Database Pure Java Driver
This style of driver converts JDBC calls into the network protocol used directly by DBMSs, allowing a direct call from the client machine to the DBMS server and providing a practical solution for intranet access.

Right side, Type 3: Pure Java Driver for Database Middleware
This style of driver translates JDBC calls into the middleware vendor's protocol, which is then translated to a DBMS protocol by a middleware server. The middleware provides connectivity to many different databases.

The graphic below illustrates JDBC connectivity using ODBC drivers and existing database client libraries.



Â

Left side, Type 1: JDBC-ODBC Bridge plus ODBC Driver
This combination provides JDBC access via ODBC drivers. ODBC binary code -- and in many cases, database client code -- must be loaded on each client machine that uses a JDBC-ODBC Bridge. Sun provides a JDBC-ODBC Bridge driver, which is appropriate for experimental use and for situations in which no other driver is available.

Right side, Type 2: A native API partly Java technology-enabled driver
This type of driver converts JDBC calls into calls on the client API for Oracle, Sybase, Informix, DB2, or other DBMS. Note that, like the bridge driver, this style of driver requires that some binary code be loaded on each client machine.

For comparison of driver types, please see the article published in Computerworld.

Partnering for Progress

Sun worked with an array of companies in the industry to create and rapidly establish the JDBC API as the industry-standard, open interface for Java applications to access databases.

Industry Momentum

Leading database, middleware and tool vendors have been building support for JDBC technology into many new products. This ensures that customers can build portable Java applications while choosing from a wide range of competitive products for the solution best suited to their needs. See the Industry Support page for a list of companies that are shipping products with support for JDBC technology.

Advantages of JDBC Technology

Leverage Existing Enterprise Data
With JDBC technology, businesses are not locked in any proprietary architecture, and can continue to use their installed databases and access information easily -- even if it is stored on different database management systems.

Simplified Enterprise Development
The combination of the Java API and the JDBC API makes application development easy and economical. JDBC hides the complexity of many data access tasks, doing most of the "heavy lifting"for the programmer behind the scenes. The JDBC API is simple to learn, easy to deploy, and inexpensive to maintain.

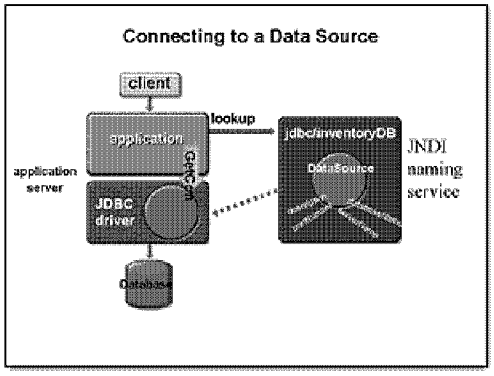
Zero Configuration for Network Computers
With the JDBC API, no configuration is required on the client side. With a driver written in the Java programming language, all the information needed to make a connection is completely defined by the JDBC URL or by a DataSource object registered with a Java Naming and Directory Interface (JNDI) naming service. Zero configuration for clients supports the network computing paradigm and centralizes software maintenance.

Key Features

Full Access to Metadata
The JDBC API provides metadata access that enables the development of sophisticated applications that need to understand the underlying facilities and capabilities of a specific database connection.

No Installation
A pure JDBC technology-based driver does not require special installation; it is automatically downloaded as part of the applet that makes the JDBC calls.

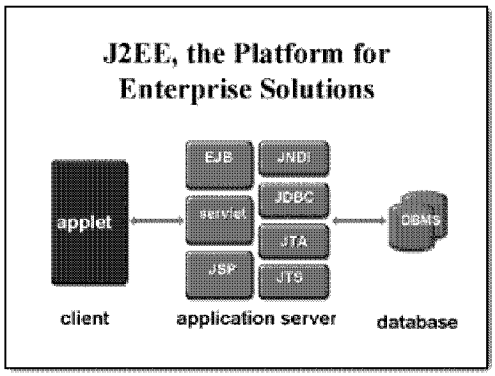
Database Connection Identified by URL
JDBC technology exploits the advantages of Internet-standard URLs to identify database connections. The JDBC API includes an even better way to identify and connect to a data source, using a DataSource object, that makes code even more portable and easier to maintain.



In addition to this important advantage, DataSource objects can provide connection pooling and distributed transactions, essential for enterprise database computing. This functionality is provided transparently to the programmer.

Included in the Java Platform
As a core part of the Java 2 Platform, the JDBC API is available anywhere that the platform is. This means that your applications can truly write database applications once and access data anywhere. The JDBC API is included in both the Java 2 Platform, Standard Edition (J2SE) and the Java 2 Platform, Enterprise Edition (J2EE), providing server-side functionality for industrial strength scalability.

An example of a J2EE based architecture that includes a JDBC implementation:



Requirements

- Software: The Java 2 Platform (either the Java 2 SDK, Standard Edition, or the Java 2 SDK, Enterprise Edition), an SQL database, and a JDBC technology-based driver for that database.
- Hardware: Same as for the Java 2 Platform.

Availability
Download the JDBC API specification, documentation, and interface classes. The JDBC API and a reference implementation of the JDBC-ODBC Bridge and related documentation are shipping now with the Java 2 SDK, Standard Edition, and the Java 2 SDK, Enterprise Edition.

Related Documents

Features

Getting Started with JDBC API