Informix
SOFTWARE
The database company

INFORMIX JDBC Driver

DATA SHEET

Version 2.20 A platform-independent, industry-standard type 4 JDBC driver

These are the sections outlined in this Technical Brief:

With more and more application developers using the Java programming language, the need to access databases from Java continues to grow. JDBC is a platform-independent application programming interface (API) that enables application developers to write database applications using a pure Java API. Based on JavaSoft's JDBC architecture, the Informix' JDBC Driver provides an industry-standard means of connecting to Informix databases. The Informix JDBC Driver achieves optimal performance and robustness because it is a pure Java implementation and uses a native interface to connect to the database server.

What Is JDBC?

JDBC is the JavaSoft specification of a standard API that allows Java programs to access database management systems (DBMSs). The JDBC API consists of a set of interfaces and classes written in the Java programming language. Using these standard interfaces and classes, programmers can write applications and applets that connect to databases, send queries written in Structured Query Language (SQL), and process the results.

JavaSoft's JDBC specification is based on the X/Open SQL CLI (Call Level Interface). The current JDBC specification supports only relational databases; however, with the version 2.0 release, JavaSoft's JDBC specification will introduce SQL3 features to support object-relational databases.

Since the JDBC API is a standard specification, a Java program that uses this API can connect to any DBMS, as long as a driver exists for that particular DBMS. A Java program uses the JDBC driver manager, distributed with the Java Development Kit (JDK), to connect to the correct JDBC driver. The JDBC driver manager can connect multiple drivers to different databases.

The diagram in Figure 1 illustrates JDBC architecture.

When a database connection is made from a Java application, the Java Virtual Machine (JVM) finds the JDBC class libraries using the Java CLASSPATH environment variable to point to the driver files.

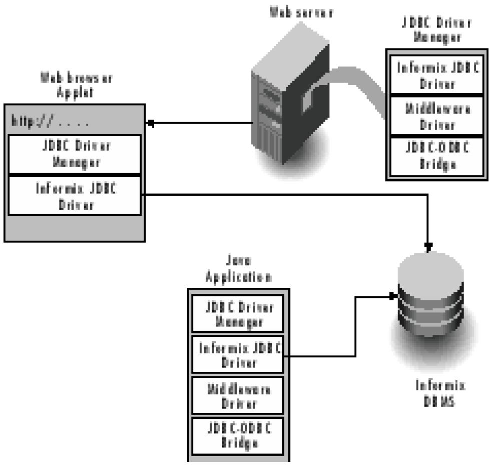

Figure 1: JDBC architecture.

When a database connection is made from an applet, the driver is downloaded from the Web server to make a connection to the database from a Web browser.

Types of JDBC Drivers

JDBC drivers can either be entirely written in Java so that they can be downloaded as part of an applet, or implemented using native methods to bridge to existing database access libraries. There are four categories of JDBC drivers, all of which implement the same database connectivity functionality:

- A type 1 driver provides JDBC access using a JDBC-ODBC bridge. This bridge provides JDBC access to most ODBC drivers. Disadvantages of this type of JDBC driver include additional performance overhead of the ODBC layer, and the requirement to load client code on each client machine.
- A type 2 driver is a partly Java driver that converts JDBC calls into the native client database API. Like the type 1 driver, this driver requires some client code to be loaded on each client machine.
- A type 3 driver is a pure Java driver that translates JDBC calls into a database-independent network protocol. The database-independent protocol is implemented using a middleware server. The middleware server then translates the database-independent protocol into the native database server protocol. A type 3 driver is typically offered by middleware vendors. Because the driver is written purely in Java, it requires no configuration on the client machine other than telling the application the location of the driver.
- A type 4 driver is a pure Java driver that uses a native protocol to convert JDBC calls into the database server network protocol. Using this type of driver, the application can make direct calls from a Java client to the database. A type 4 driver, such as Informix JDBC Driver, is typically offered by the database vendor. Because the driver is written purely in Java, it requires no configuration on the client machine other than telling the application where to find the driver.

The first three types of drivers are based on *bridge* architectures. The bridge is implemented as a process between the Java-based client and C-based database server, and translates Java methods and objects into the C functions and data types understood by database servers. Thus, any of the first three types of drivers turn a traditional 2-tier (client/server) architecture into a 3-tier architecture, and an N-tier architecture into an N+1-tier architecture, resulting in additional translation overhead and an additional process to administer. Because a type 4 driver such as Informix JDBC Driver uses the database server's native network protocol, it eliminates the additional translation and administration overhead of the bridge process.

Using a Java Language Interface

Microsoft's Open DataBase Connectivity (ODBC) API is a popular C programming language interface used to access relational databases. It offers the ability to connect to almost all databases on almost all platforms from C database applications.

If you are using the Java programming language to create applications, using the JDBC Java language API to access relational databases has the following benefits:

- ODBC is a C interface. Calls from Java to native C code cannot take full advantage of the security and portability features of Java language applications.
- The JDBC API is similar to the ODBC API, but translated into an object-oriented interface that is optimized specifically for Java language features.
- The ODBC API requires the ODBC driver manager and drivers to be installed on every client machine. JDBC applets are automatically installable, portable, and secure on any Java platform when the JDBC driver is written purely in Java.

Both the JDBC and ODBC interfaces are based on the X/Open SQL CLI (Call Level Interface), so programmers familiar with the ODBC API will find it easy to learn the JDBC API. Additionally, you can use the ODBC API from Java, by using a JDBC-ODBC bridge, described in the previous section.

SQL Conformance

Structured Query Language (SQL) is the standard language for querying relational and object-relational databases. While relational databases conform to SQL standards for basic functionality, each vendor has defined extensions that are specific to their DBMS. The goal of SQL3 is to merge these extensions into an updated standard. JavaSoft plans to introduce SQL3 features to support these extensions. Until then, the JDBC API supports the use of database-specific extensions when necessary, in the following three ways:

- The JDBC API allows any query string to be *passed through* to an underlying DBMS driver, without any syntax checking.
- Similarly to ODBC, the JDBC API implements escape clauses to enable the use of proprietary SQL syntax when required. The escape clauses provide a standard JDBC syntax for some common areas of SQL divergence.
- The JDBC API also provides descriptive information about the DBMS by means of the **DatabaseMetaData** interface so that applications can be adapted to the requirements and capabilities of each DBMS.

Benefits of Informix JDBC Driver

The Informix JDBC Driver 2.20 successfully addresses the key market needs of accessibility, openness, performance and scalability, and extensibility. This makes Informix JDBC Driver 2.20 an ideal Java connectivity technology for multi-tier Internet and Intranet application development and deployment.

The Informix JDBC Driver 2.20 targets a very broad market segment. Market segments for the JDBC 2.20 Driver include:

- Traditional Java-based applications connecting to Informix databases

- Multi-tier Internet and Intranet applications
- J2SE and J2EE applications
- Enterprise-wide and in-house Web applications
- Business-To-Business (B2B) Web applications
- Business-To-Client (B2C) Web applications

Accessibility:

The JDBC 2.20 sqlhosts/LDAP lookup facility in proxy servlet

- Makes it easy for client applications to access target database that are located behind the firewall.
- Makes it easy for applets to access a database that is running on a different host.

This provides wide accessibility of target Informix database servers from client Java applications.

Openness:

XML is widely acclaimed as the data format of the Web that brings together data from diverse sources into a uniform format. The ability of Java client applications to store and retrieve XML documents into and from the Informix database server because of the XML support in the Informix JDBC Driver 2.20 is thus ideal for Business-to-Business (B2B) Web applications.

The XML Support feature

- Makes it easy to store and retrieve XML documents to and from an Informix database using Apache's JAXP API.
- Ensures that XML documents are valid and well formed because of XML data parsing using SAX protocol.
- Uses Apache's Xerces parser (Xerces) by default, but provides ability to use any parser.
- Supports the JAXP way of creation of DOM objects.

Performance and Scalability:

By providing tuning parameters to DBA and users for connection pooling, the connection pool manager feature of the Informix JDBC Driver 2.20 provides better performance and scalability for Java client applications that use the driver to connect to Informix database servers.

Extensibility:

The user-defined datatype/routine (UDT/UDR) manager feature in JDBC 2.20 is an Informix extension of the JDBC API. Informix Dynamic Server v 9.x gives users the ability to create their own datatypes, although this is often a cumbersome process because users must code in C and understand how to use DBDK. The UDT manager feature simplifies the creation and use of user-defined datatypes by providing the facility to write them in a universal language, such as Java, from a client application that uses the JDBC API. The UDT manager API provides an infrastructure for mapping client side Java classes as data types and store their instances in the database. This feature is a big leap for Java-based DataBlade module development.

Availability:

Informix wants to make the Informix JDBC Driver available to a wide audience. A license agreement is being put in place to give rights to customers to bundle the Informix JDBC Driver with their Java applications for redistribution to end users. This means that more Java applications having the capability to connect to Informix database servers will be written.

SQLJ - Embedded SQL in Java v 1.0.1

The Informix implementation of SQLJ Embedded SQL in Java is bundled as part of the JDBC 2.20 product bundle. As with the Informix JDBC 2.20 driver Informix SQLJ 1.0.1 was developed for use with the Java 2 Platform (JDK 1.2.x or 1.3.x), supports all Informix servers on all platforms and is has no software license cost.

SQLJ has 3 parts, and the Informix JDBC Driver implements the Part 0 - Embedding SQL statements in Java Methods. Normally done in the client but it has been tested using server side JDBC also. The .sqlj file gets converted to java and the for the embedded SQL - JDBC code is generated. The benefit is in ease of use/programming.

System Requirements

Informix has developed a single, platform independent version of the Informix JDBC 2.20 and Embedded SQLJ bundle that will run on any standard Java 2 platform. If the customer requires a driver for the JDK 1.1.x platform, then the Informix JDBC 1.50 driver is the appropriate driver to use.

The JDBC 2.20 bundle is tested and certified with all these major vendor products:

Platforms:

- Sun SPARC 32-bit Solaris 2.5.1, 2.6
- Sun SPARC 32 and 64-bit Solaris 7, 8
- Microsoft NT Server - 4.0 + Service Pack 4, 5, 6
- Sun JDK - 1.2, 1.3
- Microsoft Windows 95/98
- Microsoft Windows 2000 Professional - Released Version
- Microsoft 2000 Server Family - Released Version
- HP 32-bit HP-UX 11.00
- Linux Kernel 2.2.5, Redhat 6.0
- IBM - AIX 4.3.3

XML Parsers:

- Apache - Xerces 1.1.12
- Sun - JAXP 1.0
- IBM - XML4J 3.0.1

LDAP Servers:

- OpenLDAP.org - OpenLDAP Directory Server

NOTE: Netscape and Sun Alliance - IPlanet (Netscape) Directory Server works during internal testing with the JDBC 2.20 driver, but has not been formally certified.

Informix Database Servers:

- Informix Standard Engine (SE) 5.x
- Informix OnLine 5.x
- Informix Dynamic Server (IDS) 7.x, 9.x
- Extended Parallel Server (XPS) 8.x
- Informix Internet Foundation 9.x

Migrating from earlier Informix JDBC Driver versions

Customers using the older Informix JDBC Driver 2.x versions, can seamlessly upgrade without making any changes to their existing applications or their Java envrionemnt.

Customers using the Informix JDBC Driver 1.x need to upgrade their JDK version to JDK 1.2 or higher before starting to use the JDBC 2.20 Driver from their Java applications.

About Informix

Informix Software, The database company, is a leading provider of database management systems for data warehousing, transaction processing and eBusiness applications. With more than 100,000 customers worldwide, Informix Software delivers high-performance database systems in markets including retail, financial services, government, health care, manufacturing, media and publishing, and telecommunications. For more information, visit the Informix Web site at www.informix.comInformix Corporation, based in Menlo Park, California, provides innovative database products that help the world''s major corporations to attain competitive advantage. Informix is widely recognized as the technology leader for corporate computing environments ranging from small workgroups to very large parallel processing applications. Informix''s database server, application development tools, superior customer service, and strong partnerships enable the company to be at the forefront of major information technology solution areas including data warehousing, high performance OLTP, and Web/content management. For more information, contact the sales office nearest you or visit our Web site at http://www.informix.com.

Informix Regional Sales Offices:

- Asia/Pacific 65 298 1716
- Canada(Toronto) 1 416 730 9009
- Europe/Middle East/Africa 44 181 818 1000
- Federal (USA) 1 703 847 2900
- Japan 81 3 5562 4500
- Latin America 1 305 265 7545
- North America 1 800 331 1763

1 650 926 6300