CONNX & J2EE™, a complete solution

by Larry McGhaw, Director of Product Development

The J2EE architecture allows for the creation and distribution of enterprise applications on any platform. The CONNX component that fits into the J2EE architecture is a pure Java JDBC driver that provides access to relational and non-relational data alike. CONNX technology makes non-relational data sources, including flat, text, dbase, C-ISAM, VSAM, RMS, and Dataflex files, appear as true relational databases.

A significant component of the J2EE is the Connector. The Connector is a method of wrapping data access and business rules into a single component. This component is the front door to the underlying business object. The creation of Connector components eventually requires access to the actual data. CONNX provides a pure Java method of accessing the actual data through JDBC. Most JDBC drivers provide access to relational databases, such as SQL Server, Oracle, and DB2. CONNX not only provides access to those common databases, but also to non-relational databases.

Unifying SQL

A key feature in CONNX is its unifying SQL technology. Even though most relational databases support ANSI SQL, there are subtle (and, sometimes, not so subtle) differences in the SQL supported between major relational databases, such as Oracle, DB2, and SQL Server. CONNX provides a unified ANSI SQL syntax that works across all databases, and even across those data sources that do not support SQL at all. This enables J2EE developers true database-independent data access. CONNX automatically translates the standard ANSI SQL Syntax to the native database syntax. This has the advantage of portability, but provides the performance of using native SQL. Additionally, CONNX enables you to pass true native SQL directly through to the relational database, providing the best of both worlds.

Connection Pooling and Connection Sharing

When creating enterprise applications, scalability is a key factor in the performance and success of the application. CONNX provides two technologies through JDBC that lead to increased scalability: Connection Pooling and Connection Sharing.

Connection Pooling keeps previously used connections in a "pool" for a configurable period of time. This eliminates the overhead of creating a new connection every time one is needed.
Connection Sharing enables multiple users to "share" a single physical connection resource. CONNX provides configuration options to control the number of logical users that share a single physical connection. By using these two options together, it is possible to create J2EE applications that support thousands of users with excellent performance.

Summary

CONNX and J2EE empower developers to create high performance enterprise-wide applications. By taking advantage of the CONNX portability and scalability features, developers spend more time on addressing business needs and less time on the technical details of implementation and rollout.

CONNX®
Solutions™
Simplified Data Access™