JNbridge®
SPANNING JAVA & .NET

.NET-to-Java and Java-to-.NET Cross-platform Transactions with 2-Phase Commit

Transaction: A sequence of information that is treated as an individual unit and follows the "ACID" (atomicity, consistency, isolation and durability) test. A transaction must succeed or fail as a unit, following the atomic rule of "all or nothing." A transaction must be consistent, leaving both sides in a valid state. A transaction is isolated, unaware of or not seen by other concurrently executing transactions. And a transaction must be durable, once it succeeds it must persist.

Feature

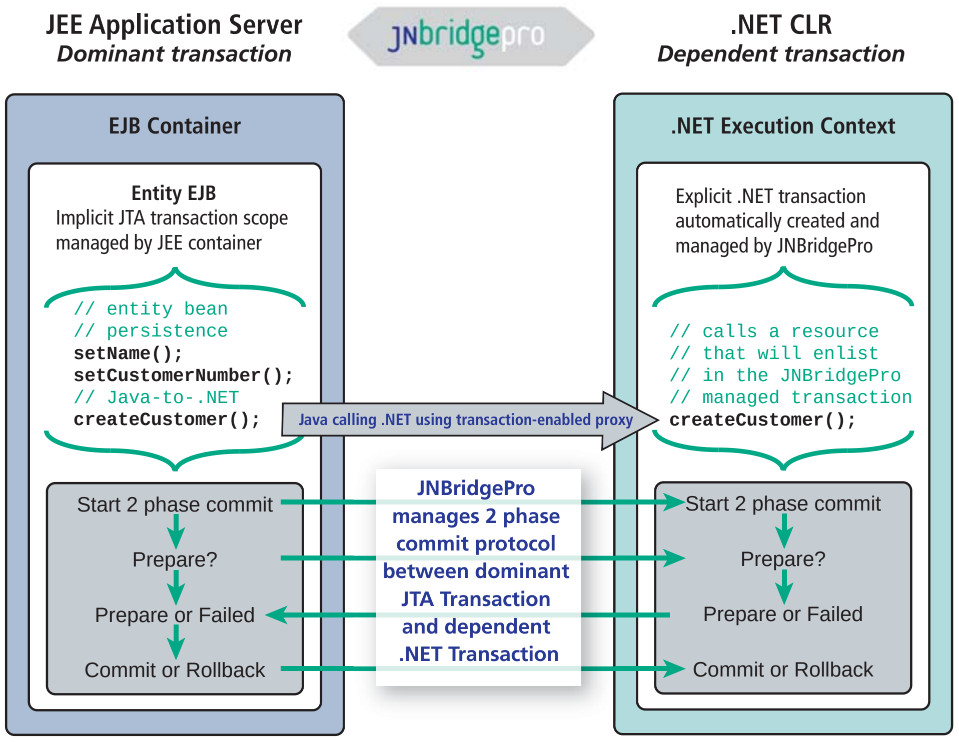
Support for cross-platform transactions that seamlessly extend container-managed Java transactions to .NET or .NET transaction scopes to Java.

Benefit

Enable cross-platform transactions quickly and easily, with near-negligible amounts of development time and effort.

Electronic transactions are the heart of our commercial world. Economic systems require a significant amount of confidence and trust between customers, shopkeepers, businesses, banks, governments and national economies. Transaction processing insures that trust by maintaining the integrity and consistency of financial data. In the data center, support for global transaction processing is provided by IT systems that use implementations that conform to accepted standards. However, transactions are not truly global as different IT frameworks implement standards differently, leading to incompatible transactions.

The incompatibilities between Enterprise Java and Microsoft transactions and transaction managers has been a barrier to integration between the business entities involved in electronic commerce. There have been industry-driven interoperability solutions like TIP (Transaction Internet Protocol) and WS-AT (Web Services Atomic Transaction). TIP is now obsolete. WS-AT requires a significant investment in additional implementation, configuration, training and other costs in order to expose the cross-platform transactions as Web services. Moreover, WS-AT also requires investment in Enterprise Java Transaction Managers that implement


JEE Application Server
Dominant transaction
JNbridgepro
.NET CLR
Dependent transaction
EJB Container
Entity EJB
Implicit JTA transaction scope managed by JEE container
.NET Execution Context
Explicit .NET transaction automatically created and managed by JNBridgePro
Java calling .NET using transaction-enabled proxy
JNBridgePro manages 2 phase commit protocol between dominant JTA Transaction and dependent .NET Transaction
Start 2 phase commit
Prepare?
Prepare or Failed
Commit or Rollback

```
// entity bean
// persistence
setName();
setCustomerNumber();
// Java-to-.NET
createCustomer();
```

```
// calls a resource
// that will enlist
// in the JNBridgePro
// managed transaction
createCustomer();
```

JNBridgePro-enabled transaction architecture in the Java-calling-.NET direction. The .NET-calling-Java direction is similar.

an additional web service protocol, WS-Coor (Web Services Coordination), to allow interoperability with Microsoft Distributed Transaction Coordinators.

JNBridge now provides an alternative to WS-AT that integrates Enterprise Java and .NET distributed transactions by making them truly cross-platform. JNBridgePro, the Java and .NET interoperability tool, has always been an integration alternative to Web services that allows Java to create and call anything in .NET and .NET to create and call anything in Java. **JNBridgePro 5.0 now supports cross-platform transactions that seamlessly extend container-managed Java transactions to .NET or .NET transaction scopes to Java.** This eliminates the overhead that's associated with loosely coupled SOAP messages implementing web services, WS-AT and WS-Coor between platforms, the JEE (Java Enterprise Edition) transaction manager and MSDTC (Microsoft's Distributed Transaction Coordinator).

In the Java-to-.NET direction, JNBridgePro supports cross-platform transactions by extending an existing implicit transaction on the JEE side to a managed explicit transaction on the .NET side. JNBridgePro automatically creates and manages a .NET "CommittableTransaction." The CommittableTransaction is associated with the thread in which Java-to-.NET calls execute on the .NET side. This transaction is dependent on the dominant Java-side transaction that's calling it, and it participates in the two-phase commit protocol that's managed by the JEE transaction manager or monitor. JNBridgePro provides a similar mechanism for .NET-to-Java calls that map the dominant .NET transaction scope to a dependent JEE "UserTransaction."

For example, say you needed to integrate a .NET-based customer billing system into a Java-based CRM system. A simple architecture would be an Entity EJB (Enterprise Java Bean) that executes in a JEE container. The JEE container manages the implicit transaction context in which the EJB executes. In this example, the Entity Bean provides the persistence mechanism for the customer data, and we'd need to integrate the .NET-based customer billing application into the Entity Bean. Using JNBridgePro, we can create transaction-enabled Java proxies of the billing application's API. The Entity Bean can then create and update customer data on both the Java–based CRM system and the .NET-based billing application. Data integrity is ensured on the Java side by the JEE container-managed transaction. Data integrity is ensured on the .NET side by a JNBridgePro-managed .NET transaction. If either side throws an exception during data processing, both transactions are rolled back. If no exception occurs and the two-phase commit protocol begins, then if during the prepare phase either the Java Transaction Manager or the MSDTC force a rollback, both transactions are rolled back.

Using JNBridgePro to enable cross-platform transactions costs a negligible amount of development time and effort—almost zero—when compared to the work entailed in implementing actions via Web services plus the investment in setting up and configuring WS-AT and WS-Coor enabled transaction managers on both sides.

JNBridge provides Java and .NET interoperability tools that simplify joining any type of program element together across the platform boundary. Unlike the single narrow view that Web services provide, JNBridgePro allows developers to access the entire object-oriented API from the other side, providing a high degree of performance, reliability, and programmatic control. JNBridge's customers include 5 of the top 10 U.S. financial services firms and over 30% of the top 25 global financial services firms.

JNbridge®

SPANNING JAVA & .NET

JNBridge, LLC
3024 Jefferson St.
Boulder, CO 80304 USA
Tel: (+1) 303.545.9371
info@jnbridge.com
www.jnbridge.com