

Challenge:

To connect a .NET-based report generation component to an existing J2EE-based trading system, with minimal effort and significant performance requirements.

Solution:

Use JNBridgePro to handle the interoperability between the JMS messaging service and the .NET component.

Benefits:

- Rapid development: initial implementation took less than one day.
- Callbacks allowed the Java side to implicitly call the .NET service without having to alter the Java code in any way.
- Callbacks enabled the .NET service to behave like a black box.
- Communicating via the JNBridge binary protocol avoided the network latency issues inherent in other solutions.

"The use of JNBridgePro enabled E-Crossnet to implement a cost effective bridge between JAVA and .NET expediently, which has given us a good return on our investment."

***--- Alan Dean
CTO, E-Crossnet***

JNBRIDGEPRO[™] SUCCESS STORY

E-Crossnet Selects JNBridgePro for Java-.NET Interoperability in a New Trading Application

E-Crossnet, a London-based financial services firm, was developing a translation tool to enable automated trading that consisted of adding a .NET reporting component in front of an existing Java-based back-end system, and needed a way to quickly and easily connect Java servlets and JMS messages to the .NET service. After exploring a number of interoperability options, E-Crossnet selected JNBridgePro to solve their Java-.NET interoperability requirements. This interview was conducted with Mujahid Shariff, the lead analyst programmer on the team.

Q: What does E-Crossnet do?

A: E-Crossnet is an Electronic Crossing Network for European equities. E-Crossnet's objective is to use technology to cut our clients' transaction costs, thereby improving investment performance. Our clients trade directly with each other, or "cross" in a wholly electronic environment. E-Crossnet is unique in that all participants in the network are investors rather than intermediaries.

Q: What kind of application are you building, and what business need does it solve?

A: Our IT team is currently developing software to enhance buy-side clients' ability to cross their trades. A client connects to E-Crossnet's central processing center via either a private network connection or the Internet. Their program and list trades are sent in proprietary formats, and translated by our system into the FIX (Financial Information eXchange) protocol. These messages are then forwarded to the crossing network with all other order traffic.

Q: How is the application architected?

A: The main components of the system are a J2EE application server (Sybase EA Server 4.1.3) on the back end, a Financial Fusion FIX server, which is Java-based, VB.NET windows service on the front end, and a JMS messaging service for communication.

Q: So, it sounds like you were faced with an interoperability challenge to get the JMS messaging service to communicate with a .NET service?

A: Correct. We needed the .NET component to both publish and subscribe to JMS.

“Other interoperability solutions available to us took too much time and effort to implement, plus the results were far from satisfactory with respect to speed and performance. JNBridgePro got us up and running very quickly, and without the latency issues of other solutions.”

***--- Mujahid Shariff
E-Crossnet Analyst
Programmer***



JNBridge is a leading provider of Java/.NET interoperability tools.

JNBridge, LLC
3024 Jefferson St.
Boulder, CO 80304
USA
Tel: (+01) 303.545.9371
Fax: (+01) 303.938.0594
info@jnbridge.com
www.jnbridge.com



E-Crossnet is a confidential electronic network that facilitates crossings between investors in pan European equities.

E-Crossnet Limited
48 Gracechurch Street
London EC3V 0EJ
Tel: +44 (0) 20 7382 1900
Fax: +44 (0) 20 7382 1901
info@ecrossnet.com
www.ecrossnet.com

Copyright © 2003 JNBridge, LLC. All rights reserved.

JNBridge and JNBridgePro are trademarks of JNBridge, LLC. E-Crossnet is a trademark of E-Crossnet Limited. All other products are the trademarks or registered trademarks of their respective owners.
(Rev. 04/03)

Q: What other interoperability solutions did you explore?

A: We first looked at generating COM components for the Java-based pieces. Both Sun and Microsoft provide COM tools, but the results were far from satisfactory.

We then investigated using Web Services. Our major problem was that different vendors provided different WSDL formats, which meant using a manual process to either edit the WSDL files and generate client proxy code, or write the client side proxies ourselves. We tested Apache, IIS, and Tomcat. We were also looking for better speed and performance than Web Services could provide.

We also tried some free components, such as ActiveJMS, but making them work was too cumbersome.

Q: What features and functions of JNBridgePro were particularly important for your application?

A: Our .NET service receives JMS messages from Java servlets. These messages are used to trigger off some processes, then, once they finished, the .NET service needs to send a JMS message back to the FIX server. Since the .NET service needs to publish and subscribe to the JMS queue, we found the callbacks in JNBridgePro to be vital for the success of this model.

The callbacks feature was the determining factor in using JNBridgePro, as it meant that the windows service could behave like a black box and be notified of activities and act upon them.

Additionally, the option of communicating over a binary protocol suited us very well, as speed and performance were also a major concern.

Q: How long did it take you to implement the bridge?

A: We originally set aside a couple of days to evaluate your product. We were absolutely delighted when we completed our initial tests in almost no time and with a minimum of fuss or effort. We were sending and receiving JMS messages from .NET clients before the end of the day.

Q: In conclusion, how would you summarize your experience?

A: Other interoperability solutions available to us took too much time and effort to implement, plus the results were far from satisfactory with respect to speed and performance. JNBridgePro got us up and running very quickly, and without the latency issues of other solutions.