Online Lender Cashes in with Improved J2EE Application Management

"The lowest mortgage rates in 40 years mean a boom business in the online lending business — if the systems can act fast enough and work reliable. With that promise of a high-payback, one of the nation’s fastest growing online mortgage services firms found a way to help their front-line Java developers overcome the complexity of J2EE management.

The result: Developers can now gain tons more visibility and control over their complex extranet applications, without having to write complex management codesets in SNMP or JMX (Java Management Extension)."

Vance McCarthy

Nexstar Financial Corp., based in St. Louis, Mo., provides relocation mortgage services to many of the country's Fortune 500 firms, as well as mortgage portals to financial services companies that don't have their own mortgage units, such as Banc One and TD Waterhouse. At the heart of their competitive edge as a mortgage service provider is a J2EE-based custom gateway Nexstar developers built to bridge their online forms applications and data with the backend mortgage approval and underwriting system owned and operated by Fannie Mae.

This architecture at first blush may appear to be a typical B2B connection. But, the gateway’s service levels are extraordinarily crucial to Nexstar's success, Mark Carlson, Nexstar's vice president for technology told Integration Developer News. Hence, Carlson's search for a better way to implement Java Management Extensions (JMX) technologies.

"Monitoring and managing performance is a key differentiator for us," Carlson told IDN, "Much of our business is based on being a mortgage service provider to other firms, like a Banc One and TD Waterhouse, for example." When a client of one of those firms clicks on the mortgage section of those web sites, the traffic is actually being run to a Nexstar site that takes the information, processes it, transfers it to computers at Fannie Mae [the nation's largest source of financing for home mortgages]

"The better the customer experience, meaning the easier it is to fill out a form and the quicker we can process an application, the better it is for our business," Carlson said.

Manage for Speed

One of the biggest, and most popular differentiators, Carlson said, is speed. "We can approve an online mortgage in a couple of minutes or less." The key is a custom gateway that Nexstar built that links his company's computers to Fannie Mae. The gateway uses a J2EE-based bridging application that aligns data formats between the Nexstar web system and Fannie Mae requirements. Further, it maps the fields of the Nexstar web-based forms to Fannie Mae's data requirements.

This gateway, which was built several years ago, is not a 100% web-based e-commerce link, Carlson concedes. One potentially performance-dragging element, for instance, is that the systems communicate using Fannie Mae's
proprietary protocol. This set-up makes it impossible for Nexstar to get its performance data back from Fannie Mae as part of the “return trip” on the transaction, so Carlson said, “that meant we needed to provide that instrumentation on our end.

The mission was not to rebuild the gateway or convert to new protocols, but simply, as Carlton put it, “to get more visibility into our velocity — how fast; and volume — how many, applications we can process and send through to Fannie Mae. We needed to know more about how our distributed components were performing.”

The Good, the Slow and the Ugly

Carlson had several options for getting better visibility inside their gateway -- and into the end-to-end process, but few were very appealing.

First, Nexstar's IT department could get some of that information by running queries, " Carlson said "but it wouldn't be in real-time, and we wouldn't have any way to really see how well the applications are performing" Carlson said. "Besides, if we're too slow or worse, if we're about to hit capacity, I'm sure one of our customers would tell us before we'd find out on our own."

Second, Nexstar could use SNMP to create some custom MIB (Management Information Base) extensions to look past the network layer into the higher-level application data. "SNMP was our first thought," but Carlson said that would have required bringing in the operations and/or sysadmin resources, and they didn't always understand how best to hook into the details of the application that Carlson's team might be looking for. "Our development team was who had the knowledge of the application, and the information we wanted," he said.

That led Carlson to JMX. The JMX framework sounded good, but Carlson said he had trouble with applying the Sun reference implementation to his needs. "It was too bare bones, and there were a couple of key things we were looking for in terms of instrumentation that didn't seem to be there," he said. Further, to customize JMX to Nexstar's situation looked like it might be more work than they bargained for. "We were looking for a way to expose attributes of our application without having to rewrite the application," and based on the available reference implementations for both JMX and SNMP, Carlson had the sinking feeling that such a re-write might be required.

Advantages of a 'Bundled' JMX

Then Carlson came across the AdventNet ManageEngine framework, and word that the latest edition (Suite 5) would bundle a pre-written implementation of JMX. After a review of the pre-release version of the product, Carlson agreed to join the beta program with an eye toward making his Nexstar-to-Fannie Mae application more manageable.

"AdventNet let us do some initial set-ups without needing to rewrite the underlying application code from scratch to support the JMX extensions"  
Mark Carlson

"AdventNet let us do some initial set-ups without needing to rewrite the underlying application code from scratch to support the JMX extensions," Carlson said. "We had some attributes we wanted to expose and using the AdventNet GUI we could
simply chose which ones we wanted, set a GET method for that attribute." All that was needed was a couple of lines of code to return the value of that attribute. AdventNet's Manage Engine lets a developer point at a JAR (Java Archive) file or a Java class file, and obtain management information on performance, throughput, etc from those files.

AdventNet's Manage Engine lets a developer point at a JAR (Java Archive) file or a Java class file, and obtain management information on performance, throughput, etc from those files.

Best of all, Carlson said, "A general overview of JMX is helpful, but not required to accomplish what we needed." In fact, Carlson said the developer who configured the JMX features on AdventNet had no JMX experience, "I only bought him a book 2 or 3 days before, and that was all he needed."

"SNMP is still the most dominant technique for adding application management"

Tony Thomas, co-founder
AdventNet Inc.

"Thanks to this implementation of JMX, we've taken the developer perspective" AdventNet's co-founder and Tony Thomas told IDN. "The developers know more about the application than anyone else on a site, yet the tools available to now really don't make it easy for them to leverage that knowledge."

To illustrate the point, Thomas said that even with JMX specifications available to java developers for some two years, "SNMP is still the most dominant technique for adding application management" to Java applications. JMX and ARM (Application Response Measurement) is gaining some use, Thomas added, but for the most part the technique is still quite complex and requires too much hand-coding for many developers to get interested.

AdventNet's approach does not require developers to create custom MIBs or to use MIB editors.

AdventNet works by automating the process of creating the Mbeans (Management Beans) that JMX needs to manage an application and to interact with the JAR or Java class files. "We can take a Java class, EJB. Or other application component and load that in, read the meta data and list all the key elements of that component," Thomas said. Once the developer has a generated list, the developer can use the AdventNet point-and-click console interface to simply define which attributes he wants to monitor and/or manage. "Our goal is to define and expose the most-commonly needed attributes, so that developers get access to as much as 80% of the information and instrumentation they want without having to write any code.

To enable developers to manage applications that run from Java to non-Java environments, such as links to legacy and Microsoft ASP/.NET resources, AdventNet will also work with simple log files, Thomas said.

There are components to ManageEngine Suite 5:

**JMX Studio**: Auto-instruments for application performance and business-level management

**Applications Manager**: An integrated console for systems, middleware and custom applications.