

Building With Java

LEADING TOOLS & SOLUTIONS FOR WEB & APPLICATION DEVELOPMENT



A SUPPLEMENT TO
SDTimes
The Industry Newspaper for Software Development Managers

APRIL 15, 2006

INTERSYSTEMS

InterSystems' Caché Jalapeño Spices Up Persistence

While a number of tools are available for mapping Java classes to underlying relational structures, maintaining persistence without mapping is tricky. Cambridge, Mass.-based InterSystems, known for bridging the gap between object and relational databases, is now offering Caché Jalapeño Persistence Library for Java, which helps developers store Java objects within an object-oriented database while providing standard SQL access to the same data.

"We've been providing Java connectivity to our Caché database for some time, but over the past year we've built a new generation of the technology that we think is pretty exciting," says Joe DeSantis, vice president of software development for InterSystems (www.intersystems.com). "The basic idea behind Jalapeño is to take Plain Old Java Objects that have



JOE DeSANTIS

Vice President, Software Development
InterSystems

been developed for an application and run them through a process that uses Java introspection to find out what's in them."

From that information, Jalapeño creates an object database schema automatically. The library can be particularly useful early on in a project when the model is undergoing a lot of change. The fact that all of Jalapeño's operations take place automatically allows users to experiment and make changes. Jalapeño then changes the model rather than requiring developers to create and remap the database

every time something is altered.

DeSantis uses an employee management system to illustrate how Jalapeño works. Using Java, a developer might create a class called Employee and use a name, job title and Social Security number as properties. Through a GUI that plugs into the developer's Java editor, Jalapeño analyzes the data to generate a persistence object called Employee that contains those properties. The database would then contain an object definition that corresponds to the Employee Java class

along with a runtime engine that can save and retrieve the object. While the object maintains simple persistence, it can be queried through SQL and JDBC using the same connection.

"Java itself is not really rich enough to describe everything you might want in a

INTERSYSTEMS

database, but Java 1.5 has a nice concept called annotations," DeSantis says. "Annotations let you add additional bits of information to a class. For example, users could go back to their Employee class and add an annotation in the database that lets them look up employees by name. That indexing really increases search speed. Having the Caché object database in the back end rather than a relational database gives users the ability to find relationships between objects or make use of inheritance."

While DeSantis believes in the strength of object databases, he also sees the conundrum of simultaneously wanting the features of a relational database. For Java programmers, leaping from objects to rows and columns and back can be difficult. "Any large application is going to have a lot of pieces that have to be put into a database somewhere, whether they're customer, patient or financial records," he says. "Often, it's best to use an object model because that's how the rest of the application works."

While the object model might be the best one, to many developers a traditional relational structure can look more appropriate for this type of data. The Jalapeño library lets users lay out their Java classes however they like, whether they're using a UML modeler, Eclipse or something else.

"They can start up our tool, point at their .JAR file or set of class definitions, and it creates an object database schema that maps to the correct structure," DeSantis explains. "It even supports things like schema evolution. It's very straightforward and it's very simple. You can use it in everything from a very lightweight Java application to a full-blown J2EE server."

DeSantis concludes, "We used to have users start from the object database and create Java for them. Now, they can start in Java, and we can create the database for them. It's a friendlier approach for someone who doesn't know much about InterSystems' products but knows a lot about Java." ◀