

DEVELOPING THREE TIER WEB APPLICATIONS with an OBJECT-ORIENTED DATABASE and JAVA TECHNOLOGY

Mark B. Jones

**School of Science and Computer Engineering
University of Houston – Clear Lake
Houston, TX 77058 US
(281) 467-9872
Fax: (281) 226-7596
Mark.B.Jones@uth.tmc.edu**

Morris M. Liaw, Ph.D.

**School of Science and Computer Engineering
University of Houston – Clear Lake
Houston, TX 77058 US
(281) 283-3876
Fax: (281) 226-7596
liaw@uhcl.edu**

Abstract

This paper details the use of the ObjectStore 6.2 object-oriented database and Java API, Apache Tomcat Servlet/JSP container, J2EE, and the NetBeans 5.0 Java IDE (Integrated Development Environment) to develop a three tier Web application. The major contribution of this paper is to show how a three tier web application can be easily developed and take advantage of an object-oriented database instead of using ADO.Net or JDBC libraries to access traditional relational databases, such as Microsoft Access, MS SQL Server, or Oracle. The NetBeans 5.0 IDE is used to run the ObjectStore postprocessor and a simple class is developed to illustrate that ObjectStore can be used as the backend of three tier Web applications.

Keywords

Object-Oriented Database, J2EE, ObjectStore, NetBeans, Web Application, Java Technology

1 INTRODUCTION

One of the essential tools in application development is an Integrated Development Environment (IDE). Also key is the existence of Application Programming Interface (API) libraries that abstract common programming tasks such as making database connections. This paper shows how the popular NetBeans 5.0 Java IDE can be used to develop a standard three tier Web application that uses ObjectStore 6.2 for the backend database and the use of simple Java classes to abstract communication with the database. This includes connecting to ObjectStore, writing and retrieving data, and configuring NetBeans to run the ObjectStore postprocessor on “persistence-capable” classes with the built in NetBeans ‘Build Project’ function.

NetBeans is an open source Java IDE that is sponsored by Sun Microsystems [4]. It features an integrated Apache Tomcat server which is an open source J2EE JSP/Servlet container and uses Apache Ant, an open source build tool, to build and run projects.

ObjectStore 6.2 is an Object Database Management System that is designed to complement object oriented languages such as C++ and Java.

This project differs from existing projects such as the “Online Grade Book” application [3] in three ways. While both projects are Java based the previous work did not make use of an IDE. This project uses NetBeans to facilitate the building and packaging of the application into a standard Web Archive (war) file that can be deployed on most J2EE application servers. NetBeans was also configured to handle running the ObjectStore postprocessor. And finally, the previous work relied on custom client and server classes to talk to the database. In this project the client and server classes were eliminated such that the application connects directly to the database which is a more portable solution.

A Web application was partially implemented using this approach. The Online Quiz Application can be tested at the following link using id ‘admin’ and password ‘admin’: <http://oodb-8.cl.uh.edu:8080/Quiz/>.

2 THE ONLINE QUIZ APPLICATION

The Online Quiz application is an online learning tool designed to allow instructors to build a bank of questions organized by book, chapter, and section that can be grouped into quizzes for students. The idea is that the instructor can organize quizzes around the structure of a particular text book or set of text books. Students can use it to take quizzes created by the instructor and track their scores.

This application was built using the J2EE JSP/Servlet technology [1] and the Model-View-Controller design pattern [2]. It has a student interface and an instructor or administrator interface. Students can take quizzes and view grades. Administrators can view/update the question bank, create quizzes, add students, and view reports.

The page shown in Figure 1 is the student home page. From here students can take quizzes or view their grades. Instructors see the home page shown in figure 2. It lists some statistics about the contents of the Quiz database and shows options for viewing/updating the question bank, creating quizzes, adding students, and viewing reports.



Figure 1. Student view home page.



Figure 2. Instructor view home page.

Class Name	Persistence Capable	Description
Controller.java		Controller servlet for the application
ODB.java		Manages database connections and database queries
Person.java	Yes	Stores student and instructor information
Question.java	Yes	Super class for question classes
TrueFalseQuestion.java	Yes	Subclass for true/false questions
MultipleChoiceQuestion.java	Yes	Subclass for Multiple Choice questions
Quiz.java	Yes	Stores information about specific quizzes
QuizResponse.java	Yes	Associates specific students with specific quizzes

Figure 3. Online Quiz Application middle tier Java classes

The Quiz application stores six types of objects (Shown in figure 3) in four root objects (figure 4). The

six Java classes that are stored in the database are run through the ObjectStore postprocessor, after they are compiled, in order for them to be made persistence-capable.

The database stores four types of objects in collections using the ObjectStore class `com.odi.util.OSHashMap`. These collections are named objects called 'roots'. The four roots are shown in figure 4.

Root Name	Objects stored
"allUsers"	Person: name, ID, password
"allQuestions"	Question: Book, chapter, section, question text TrueFalseQuestion: Correct Answer, Student Response MultipleChoiceQuestion: Correct Answers, Student Responses
"allQuizzes"	Quiz: Name, description, Questions
"assignedQuizzes"	QuizResponse: a Person, a Quiz

Figure 4. Database Root objects

3 USING NETBEANS 5.0

One of the challenges to using ObjectStore is the need to run the postprocessor on classes that will be stored in the database. This can be overcome and made quite transparent using the NetBeans IDE.

The first task is installing ObjectStore 6.2. ObjectStore was installed using the default install configuration on a PC running Microsoft Windows XP Professional SP 2. NetBeans 5.0 was installed using the JDK 5.0/NetBeans 5.0 bundle available from 'java.sun.com'. Next the CLASSPATH environment variable was added to make the ObjectStore Java libraries available to the ObjectStore postprocessor when it is invoked by NetBeans; and the PATH environment variable was edited to include paths to the Java runtime and the ObjectStore postprocessor executable.

Required Environment Variables (assuming the default ObjectStore install location):
 CLASSPATH-c:\ODI\OSJI\tools.jar;c:\ODI\OSJI\osji.jar
 PATH-C:\j2sdk1.4.2_12\bin;C:\ODI\OSJI\bin;

The next step is to create a project in NetBeans. Once a project is created all of the jar libraries from the ... \ODI\OSJI\ directory should be added to the project.

NetBeans uses generated Apache Ant scripts to compile and build projects and it gives the programmer a

method to insert custom script into these generated scripts. This is how NetBeans can be configured to run the ObjectStore postprocessor.

In the 'nbproject' folder of the NetBeans project there is a file named 'build-impl.xml'. In this file there are several 'target' tags (or elements) like the following.

```
<target name="-post-compile">
  <!-- Empty placeholder for easier customization. -->
  <!-- You can override this target in the ../build.xml file. -
  ->
</target>
```

These 'target' tags allow for instructions that are added to the 'build.xml' file to be inserted into the 'build-impl.xml' script when it is invoked by NetBeans. Since the ObjectStore postprocessor must be run after the class has been compiled this is where we will need to run the postprocessor. So the following lines of Ant script were added to the 'build.xml' file.

```
<target name="-post-compile">
  <exec dir="${build.classes.dir}/quiz/"
  executable="osjcfp.bat">
    <arg line="@cfpargs"/>
  </exec>
</target>
```

This will invoke the ObjectStore postprocessor (osjcfp.bat) using the command line options specified in the file 'cfpargs'. The 'cfpargs' file is saved in the NetBeans project along with the Java source files and contains the following two lines.

```
-inplace -dest .
-pc Person.class Question.class TrueFalseQuestion.class
MultipleChoiceQuestion.class Quiz.class
QuizResponse.class
```

With the NetBeans project configured this way, each time the NetBeans 'build' feature is invoked the ObjectStore postprocessor is run on the classes specified in the 'cfpargs' file with out any extra effort.

4 CONNECTING TO THE DATABASE

In the Quiz application the ODB.java class is used to handle connection to and all communication with ObjectStore. This approach requires that ObjectStore and Tomcat both be installed on the same server. This is also the case when using Microsoft Access. Connecting to the database and issuing queries is quite simple. Using the ODB.java class connecting to the database only requires the following simple statement:

```
ODB odb = new ODB(path);
```

The ODB class uses code from the ObjectStore API to open the database connection.

```
//Open the database or create a new one if necessary.
try {
  database = Database.open(DATABASE,
ObjectStore.UPDATE);
} catch (DatabaseNotFoundException e) {
  database = Database.create(DATABASE,
ObjectStore.ALL_READ | ObjectStore.ALL_WRITE);
}
```

The 'path' is the path and file name where the database file exists or will be created. This file can exist anywhere on the server similar to MS Access MDB files.

All other database communication is abstracted through the ODB.java class making updating the database very easy. For example, the following are example calls to add, retrieve, and delete objects from the database respectively.

```
boolean addSuccessful = odb.addPerson(new
Person(firstName,lastName,stuId,stuId));
Person thisUser = odb.getPerson(userName);
boolean deleteSuccessful = odb.deleteQuestion(qText)
```

The ODB.java class manages the details of working with ObjectStore sessions, transactions, and queries using the ObjectStore API classes.

5 CONCLUSION

The development of the Online Quiz application shows that the NetBeans Java IDE can be used to develop standard, portable Web applications that take advantage of an object-oriented database. It further shows that communication with a database such as ObjectStore can be accomplished using NetBeans to run the ObjectStore postprocessor and a simple Java class instead of using JDBC libraries to access traditional relational databases.

6 REFERENCES

[1] Hall, M., Core Servlets and Java Server Pages. Upper Saddle River, NJ: Prentice Hall PTR, 2000.

[2] Alur, D., Crupi, J., Malks, D., Core J2EE Patterns: Best Practices and Design Strategies, 2nd edition. Upper Saddle River, NJ: Prentice Hall PTR, 2003.

[3] Moen, Charles R., Liaw, Morris M., “Using an Object-Oriented Database to Support a Web Application Built with Java Technologies”

[4] NetBeans.org, “Welcome to the NetBeans Community”, NetBeans IDE, May 2007
<<http://www.netbeans.org/about/index.html>>