developerWorks®

An analysis of the Apache Geronimo PetStore demo

Construct a solid enterprise application platform

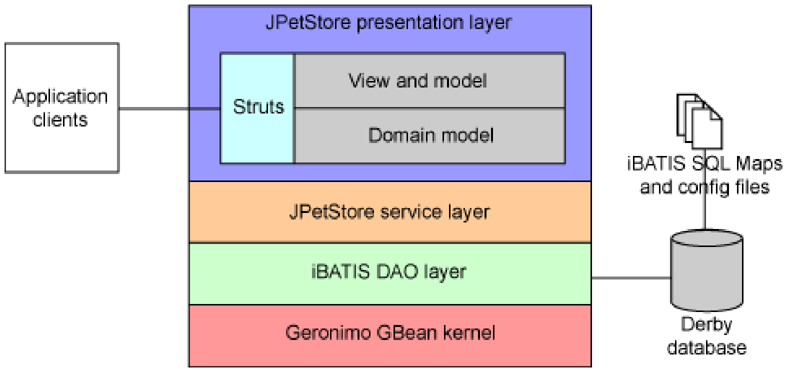
J. Jeffrey Hanson

October 03, 2006

Want practical instructions for building an enterprise application that you can use in your business? The iBATIS PetStore application is an example application that originated from the Sun Java™ BluePrints program. The application illustrates how to use the capabilities of the iBATIS persistence framework, the all-Java Apache Derby database, and Java Platform, Enterprise Edition (Java EE) to develop a simple cross-platform enterprise application. This article provides tips and techniques that you can use to exploit the features of iBATIS, Derby, and Apache Geronimo to construct a flexible and usable implementation of the PetStore application.

Based on the Sun Java BluePrints program's PetStore application, JPetStore, the iBATIS PetStore application supports a wide assortment of databases executed on a variety of application servers. The user interface (UI) framework is implemented using Apache Struts. The persistence layer is composed using the iBATIS SQL Maps framework and the iBATIS Data Access Objects (DAO) framework. The application uses the Derby database for the persistent store. For the purposes of this article, you'll use Geronimo as the application server. Figure 1 shows a high-level conceptual illustration of the relationships of each of these components and frameworks.

Figure 1. The components and frameworks of the JPetStore application



Download and deploy the iBATIS PetStore application

1. First, download the Geronimo 1.1 release with Jetty (see Resources for a link) and extract the files to a directory to be referred to as {GERONIMO_HOME}.

2. Then extract the JPetStoreAPP4Geronimo.zip file into the directory of your choice, to be referred to as {JPETSTORE}.
3. Next, move the following two {JPETSTORE} files to the {GERONIMO_HOME} directory:
   - **jpetstoreAPP.war**: The JPetStore application Web Archive (WAR) file
   - **jpetstoreAPP-geronimo-jetty.xml**: The application deployment descriptor for Geronimo
4. Create a script file, using the {JPETSTORE}/setupM4.bat file or {JPETSTORE}/setupM4.sh file as a guide, to initialize the environment variables for your specific Java EE Java Archive (JAR) file, Geronimo installation, and Java software development kit (JDK).
5. In a command window or shell, execute your customized setupM4 script file, and then start the Geronimo server from {GERONIMO_HOME} using the command `java -jar bin\server.jar`.

Deploy the Derby database for the PetStore application

1. Set up the JPetStoreDB database by unzipping the files from {JPETSTORE}/JPetStoreDB4Geronimo.zip into the {GERONIMO_HOME}/var/derby directory, to be referred to as {Derby_System_Home}.
2. Create a {GERONIMO_HOME}/repository/incubator-derby/jars directory, and then download derby-10.0.2.1.jar and derbytools-10.0.2.1.jar into this directory.
3. Create a file named JPetstoreDBPlan.xml from which you'll construct the Derby database as shown in Listing 1.

Listing 1. JPetstoreDBPlan.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<connector xmlns="http://geronimo.apache.org/xml/ns/j2ee/connector-1.1">
   <dep:environment
      xmlns:dep="http://geronimo.apache.org/xml/ns/deployment-1.1">
      <dep:moduleId>
         <dep:groupId>console.dbpool</dep:groupId>
         <dep:artifactId>JPetstoreDerbyDataSource</dep:artifactId>
         <dep:version>1.0</dep:version>
         <dep:type>rar</dep:type>
      </dep:moduleId>
      <dep:dependencies>
         <dep:dependency>
            <dep:groupId>org.apache.derby</dep:groupId>
            <dep:artifactId>derby</dep:artifactId>
            <dep:version>10.1.1.0</dep:version>
            <dep:type>jar</dep:type>
         </dep:dependency>
      </dep:dependencies>
   </dep:environment>
   <resourceadapter>
      <outbound-resourceadapter>
         <connection-definition>
            <connectionfactory-interface>
               javax.sql.DataSource
            </connectionfactory-interface>
            <connectiondefinition-instance>
               <name>JPetstoreDerbyDataSource</name>
               <config-property-setting name="Password">
                  APPdbpw
               </config-property-setting>
               <config-property-setting name="Driver">
                  org.apache.derby.jdbc.EmbeddedDriver
               </config-property-setting>
               <config-property-setting name="UserName">
                  APP
               </config-property-setting>
               <config-property-setting name="ConnectionURL">
                  jdbc:derby:JPetstoreDB
               </config-property-setting>
               <connectionmanager>
                  <local-transaction/>
                  <single-pool>
                     <max-size>10</max-size>
                     <min-size>0</min-size>
                     <blocking-timeout-milliseconds>
                        5000
                     </blocking-timeout-milliseconds>
                     <idle-timeout-minutes>30</idle-timeout-minutes>
                     <match-one/>
                  </single-pool>
               </connectionmanager>
            </connectiondefinition-instance>
         </connection-definition>
      </outbound-resourceadapter>
   </resourceadapter>
</connector>
```

4. Deploy and initialize the JPetStore data source by opening a new command window, traversing to the {GERONIMO_HOME} directory, running your customized setupM4 script, and then executing the following commands:

```
java -jar bin/deployer.jar --user system --password manager distribute
JPetstoreDBPlan.xml
repository/tranql/rars/tranql-connector-<your_version>.rar
```

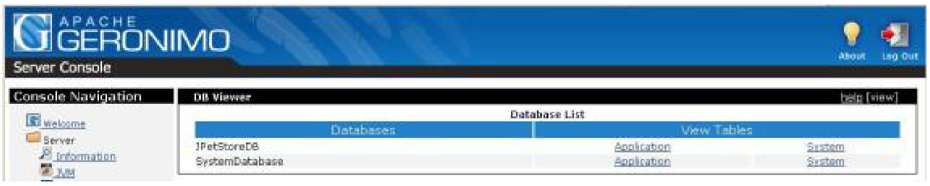
The output should resemble the following:

```
Distributed console.dbpool/JPetstoreDB/1.0/rar
```

At this point, it would be interesting to look at the deployment inside of Geronimo's DB Manager screen available from the Geronimo Server Console.

5. Log in to the Geronimo Server Console in a browser window pointed to http://localhost:8080/console.
6. Select the **Embedded DB/DB Manager** node, and you should see a screen resembling Figure 2.

Figure 2. The JPetStoreDB deployment within the Geronimo Server Console

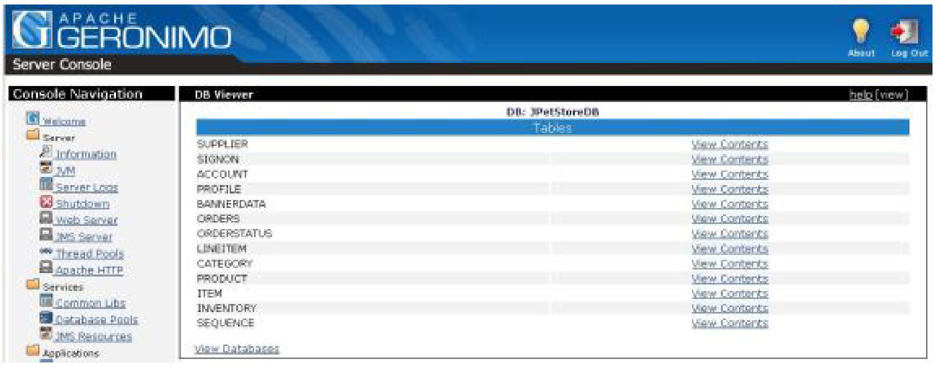


Database List

Databases	View Tables	
JPetStoreDB	Application	System
SystemDatabase	Application	System

7. You can view the application tables of the JPetStoreDB database by selecting the **Application** link under the View Tables column. The JPetStoreDB database tables should look like those illustrated in Figure 3.

Figure 3. The JPetStoreDB table list



DB: JPetStoreDB

Tables	
SUPPLIER	View Contents
SIGNON	View Contents
ACCOUNT	View Contents
PROFILE	View Contents
BANNERDATA	View Contents
ORDERS	View Contents
ORDERSTATUS	View Contents
LINEITEM	View Contents
CATEGORY	View Contents
PRODUCT	View Contents
ITEM	View Contents
INVENTORY	View Contents
SEQUENCE	View Contents

View Databases

8. Next, copy the contents of the jpetstore-derby-dataload.sql file into the SQL Command/s: text area within the Geronimo Server Console, and then click the **Run SQL** button. This populates the JPetStoreDB database with test data.
9. Now you can start the database by selecting the **Applications/J2EE Connectors** node in the Geronimo Server Console and clicking the **Start** link next to the console.dbpool/ JPetstoreDB/1.0/rar component.

Repackage and deploy the PetStore application .war file

1. Now create and save a Web application deployment plan named JPetStoreWebApp-plan.xml with the content shown in Listing 2.

Listing 2. JPetStoreWebApp-plan.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://geronimo.apache.org/xml/ns/j2ee/web-1.1">
  <dep:environment xmlns:dep="http://geronimo.apache.org/xml/ns/deployment-1.1">
    <dep:moduleId>
      <dep:groupId>default</dep:groupId>
      <dep:artifactId>JPetStoreAPP</dep:artifactId>
      <dep:version>1-default</dep:version>
      <dep:type>car</dep:type>
    </dep:moduleId>
    <dep:dependencies>
      <dep:dependency>
        <dep:groupId>geronimo</dep:groupId>
        <dep:artifactId>j2ee-server</dep:artifactId>
        <dep:version>1.1</dep:version>
        <dep:type>car</dep:type>
      </dep:dependency>
      <dep:dependency>
         <dep:groupId>console.dbpool</dep:groupId>
         <dep:artifactId>JPetstoreDerbyDataSource</dep:artifactId>
        <dep:version>1.0</dep:version>
        <dep:type>rar</dep:type>
      </dep:dependency>
    </dep:dependencies>
    <dep:hidden-classes/>
    <dep:non-overridable-classes/>
  </dep:environment>
  <context-root>/jpetstoreAPP</context-root>
  <resource-ref>
    <ref-name>jdbc/JPetstoreDerbyDataSource</ref-name>
    <resource-link>JPetstoreDerbyDataSource</resource-link>
  </resource-ref>
</web-app>
```

At this point, you need to modify the application's .war file for it to deploy in Geronimo 1.1 successfully.

2. Extract the contents of the jpetstoreAPP.war file to an empty directory.
3. Remove the `<security-constraint>` element (including its contents) from the WEB-INF/web-xml file.
4. Change the `<res-ref-name>jdbc/JPetStoreDB</res-ref-name>` element to read `<res-ref-name>jdbc/JPetstoreDerbyDataSource</res-ref-name>`.
5. Add a new file to the WEB-INF directory named geronimo-web.xml, then place the contents shown in Listing 3 in this file and save it.

Listing 3. geronimo-web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
    xmlns="http://geronimo.apache.org/xml/ns/j2ee/web-1.1">
    <environment>
        <moduleId>
            <artifactId>jpetstoreAPP</artifactId>
        </moduleId>
        <dependencies>
            <dependency>
                <groupId>console.dbpool</groupId>
                <artifactId>JPetstoreDerbyDataSource</artifactId>
            </dependency>
        </dependencies>
    </environment>

    <context-root>/jpetstoreAPP</context-root>

    <resource-ref>
        <ref-name>jdbc/JPetstoreDerbyDataSource</ref-name>
        <resource-link>JPetstoreDerbyDataSource</resource-link>
    </resource-ref>
</web-app>
```

6. Make sure that the `dataSource` elements in the WEB-INF\classes\com\ibatis\jpetstore\persistence\sqlmapdao\sql\JNDIsql-map-config.xml file and in the WEB-INF\classes\com\ibatis\jpetstore\persistence\sqlmapdao\sql\sql-map-config.xml file appear as shown in Listing 4.

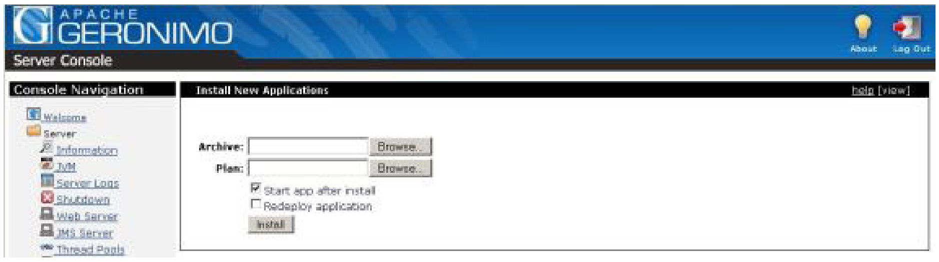
Listing 4. dataSource elements for config files

```
<dataSource type="JNDI">
   <property name="DBJndiContext"
             value="java:comp/env/jdbc/JPetstoreDerbyDataSource"/>
</dataSource>
```

7. Next, repackage the contents of the .war file by creating a .zip file or .jar file of the directory in which you extracted the contents of the .war file and renaming the .zip or .jar file to jpetstoreAPP.war.

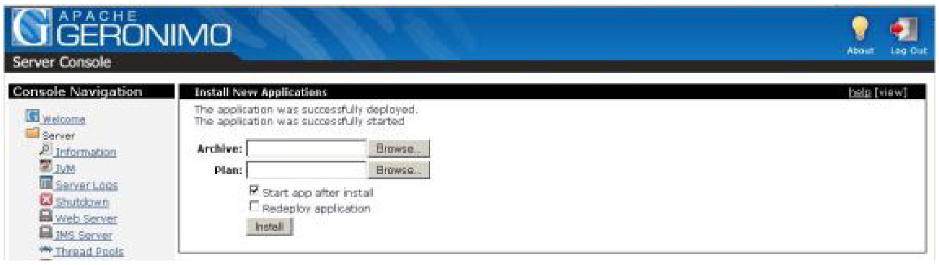
8. Deploy and start the JPetStore application by selecting the **Applications/Deploy New** link in the Geronimo Server Console. This displays a page similar to Figure 4.

Figure 4. The Geronimo Server Console application deployment page



9. Click the **Browse** button to the right of the **Archive:** text box, and navigate to the jpetstoreAPP.war file.
10. Next, click the **Browse** button to the right of the **Plan:** text box, and navigate to the Web application deployment plan (JPetStoreWebApp-plan.xml) created earlier.
11. Make sure the **Start app after install** checkbox is checked, then click the **Install** button. You should see a success page similar to Figure 5.

Figure 5. The Geronimo Server Console application-deployment success page


The application was successfully deployed.
The application was successfully started

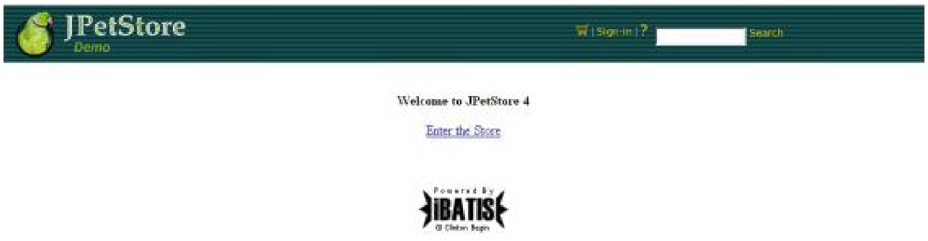
12. Now restart the Geronimo application server, and the PetStore application is ready for use.

Run the iBATIS PetStore application

To run the iBATIS PetStore application, do the following:

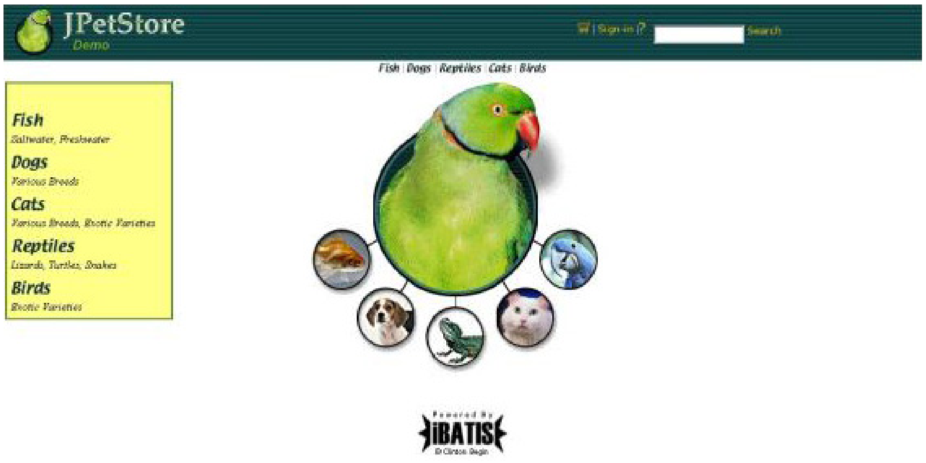
1. Open a browser window and go to http://localhost:8080/jpetstoreAPP/. You should see a page similar to Figure 6.

Figure 6. The iBATIS login page


Welcome to JPetStore 4

2. Press the **Enter the Store** link, which should bring up a page similar to Figure 7.

Figure 7. Main shopping page of the PetStore application



Register a new user

Whenever a user visits the store for the first time, the user should follow the following steps to register as a new user before making any purchases:

1. Click **Sign In** on the home page.
2. Click the **Register Now** button from the sign-in page.
3. Fill out the **User Information**, **Account Information**, and **Profile Information** sections.
4. Click **Submit**.

Place an order

To purchase a Manx cat, for example, follow these steps:

1. On the main shopping page, select **Cats**.
2. Click the link next to the **Manx** item.
3. Click **Add to Cart** in the row for the Manx of your choice.
4. Click the **Proceed to Checkout** button.
5. Click the **Continue** button on the next page.
6. Click the **Submit** button on the page displaying the payment details and billing address.
7. Click the **Continue** button on the confirmation page.

A purchase-receipt page displays. At this point, you can continue shopping or sign out.

Dissecting the iBATIS PetStore application

The iBATIS PetStore application utilizes a multitier design to keep presentation components, business components, and persistence components loosely coupled and clearly separated. The presentation layer is constructed within the Struts framework using `org.apache.struts.action.ActionForm` beans to interact with services that interact with the persistent tier. These services create a model that the presentation tier consumes to create a view to send to a client.

The iBATIS PetStore application consists of the following high-level components:

- **Storefront page**: Presents a browser-based HTTP front end that customers can use to place orders
- **Order-processing page**: Presents a form-based interface to process orders placed by customers
- **`OrderService` class**: Receives requests from the Struts presentation tier, makes requests to the iBATIS persistence tier, and saves and retrieves orders

Figure 8 illustrates the relationships and specializations for the PetStore presentation framework.

Figure 8. Class diagram for the PetStore presentation framework

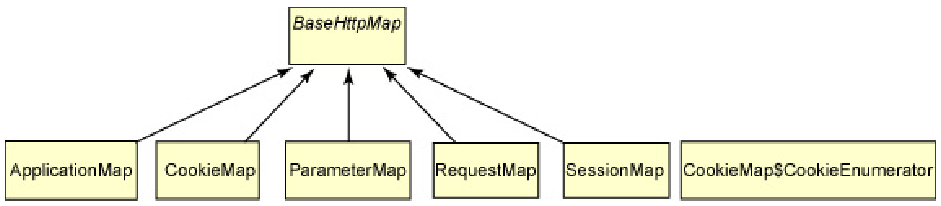
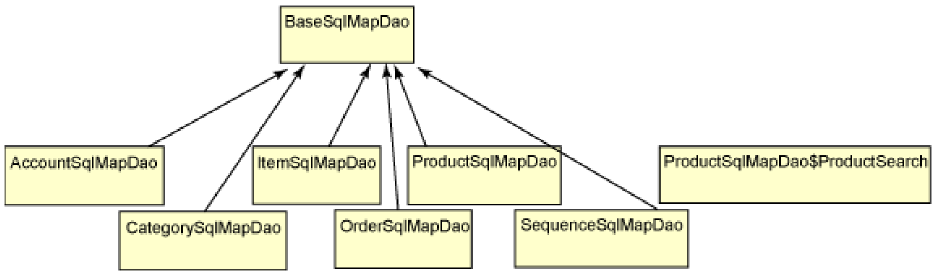


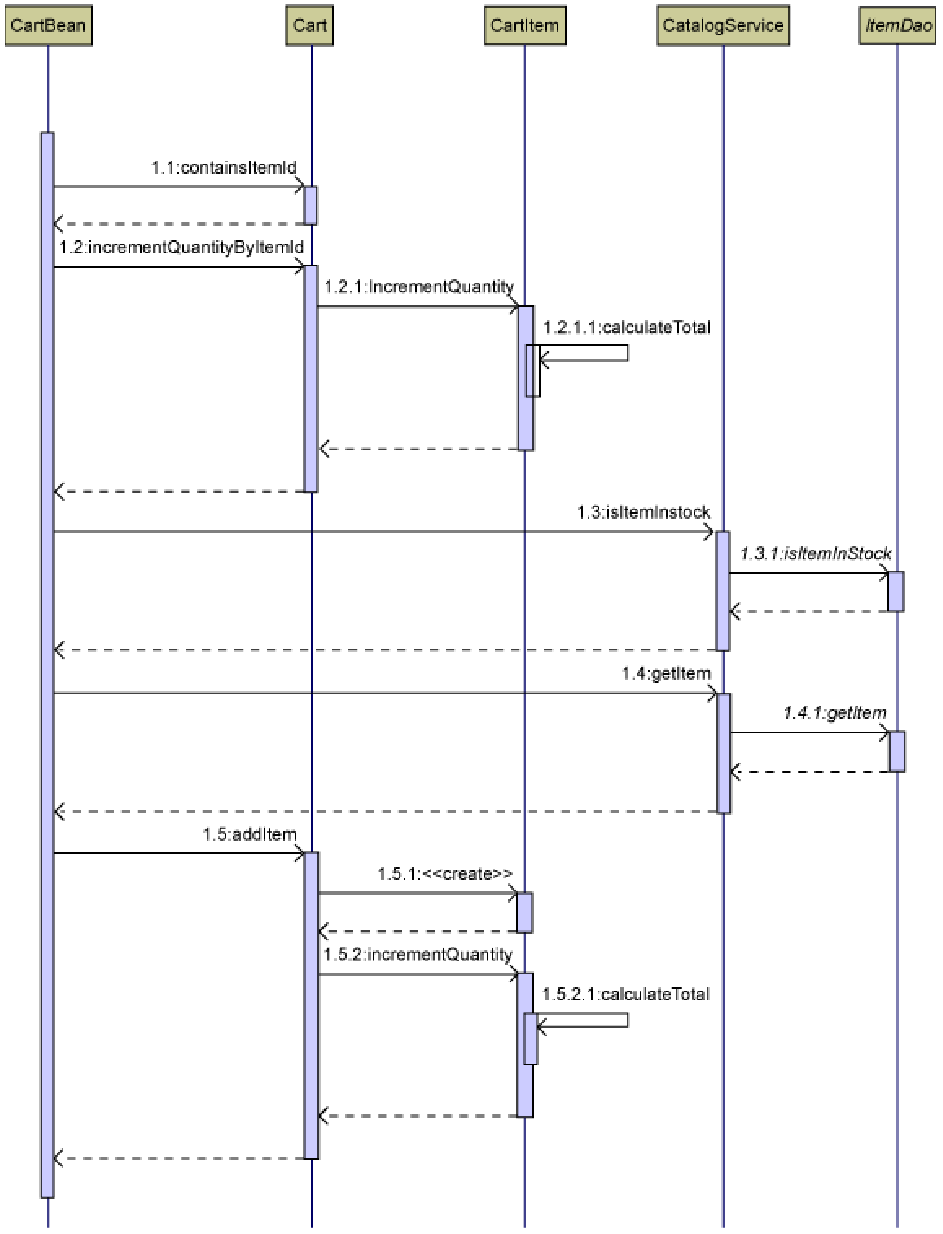
Figure 9 displays the relationships and specializations for the PetStore DAO framework.

Figure 9. Class diagram for the PetStore DAO framework



Adding an item to the virtual shopping cart involves interesting interactions between all tiers of the application framework. Specifically, plain old Java objects (POJOs) from the presentation tier interact with services in the business domain tier, which interact with DAOs in the persistence tier to access and manipulate data. Figure 10 illustrates the flow of sequences during the add-item-to-cart procedure.

Figure 10. The flow of sequences during the add-item-to-cart procedure



The database and persistence framework

This iBATIS PetStore application uses the iBATIS SQL Maps framework and the iBATIS DAO framework to access a fully initialized Derby database provided in the JPetStoreDB4Geronimo.zip file. iBATIS doesn't use its own proprietary query language; rather, it relies on standard SQL used within the SQL Maps framework.

The iBATIS SQL Maps framework provides a simple means of moving data between Java objects and a relational database. The SQL Maps framework maintains loose couplings between your Java classes and SQL using a simple XML-mapping descriptor. The iBATIS DAO framework is an abstraction layer that provides a common API to hide the details of persistence mechanisms from an application.

Using Apache Derby and iBATIS together is truly a trivial solution with a small set of dependencies. As mentioned earlier, the only libraries required are the derby-10.0.2.1.jar file and the derbytools-10.0.2.1.jar file to run Derby, and the ibatis-common-2.jar file, the ibatis-sqlmap-2.jar file, and the ibatis-dao-2.jar file to run iBATIS.

Summary

The iBATIS PetStore application illustrates how to use the capabilities of Struts, iBATIS, and Derby to develop a scalable, flexible, cross-platform enterprise application for the Java EE platform. In this article, you discovered how to integrate Struts, iBATIS, and Derby with Geronimo to construct a simple and usable incarnation of the PetStore application.

You were introduced to the high-level architecture of the iBATIS PetStore application and the associations of each of the frameworks used. Each framework (Struts, iBATIS, Derby, and Geronimo) fills a vital purpose in enterprise application development in a clean and straightforward manner. You learned that the Geronimo framework offers a solid foundation of tools and services (such as a data-source wizard, an application deployer, and a database manager) that complements the functionality of Struts, iBATIS, and Derby to form a complete enterprise application development and management platform.

Downloadable resources

Description	Name	Size
iBATIS PetStore support files	GeronimoPetStoreSrc.zip	83KB

© Copyright IBM Corporation 2006
(www.ibm.com/legal/copytrade.shtml)
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)