

Sams Teach Yourself EJB in 21 Days

Copyright © 2003 by Sams Publishing

International Standard Book Number: 0-672-32423-7

(d)Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

When reviewing corrections, always check the print number of your book. Corrections are made to printed books with each subsequent printing. To determine the printing of your book, view the copyright page. The print number is right-most number on the line below the "First Printing" line. For example, the following indicates the 4th printing of a title.

First Printing: June 2002

05 04 03 02 4 3 2 1

Misprint	Correction
<p>Page xxix, Ragae Ghaly bio</p> <p>He is an independent consultant, and he...</p>	<p>He is a senior architect for Digital Evolution, a Web Services company. He...</p>
<p>Page 15, paragraph immediately following first note</p> <p>The most common examples of a view beans are in designing a GUI.</p>	<p>The most common examples of a view bean are in GUI design.</p>
<p>Page 17, paragraph following first bulleted list</p> <p>If the containers in which EJB components are deployed assume responsibility for managing runtime services such as persistence, transactions, and concurrent database access, bean developers are free to focus on developing software components that encapsulate business logic.</p>	<p>Bean developers are free to focus on developing software components that encapsulate business logic, as the containers in which EJB components are deployed assume responsibility for managing runtime services such as persistence, transactions, and concurrent database access.</p>
<p>Page 33, paragraph in middle of page</p> <p>Entity beans typically contain data-related logic, such as inserting, updating, and removing a customer record in the database.</p>	<p>Entity beans typically contain data-related logic, such as inserting, updating, and removing a record in the database.</p>
<p>Page 51, first bullet</p> <p>A remote home interface extends the java.ejb.EJBHome interface, whereas a local home interface extends javax.ejb.EJBLocalObject.</p>	<p>A remote home interface extends the java.ejb.EJBHome interface, whereas a local home interface extends javax.ejb.EJBLocalHome.</p>
<p>Page 51, third bullet</p> <p>The bean class implements the javax.ejb.SessionBean interface. In addition, it implements all business methods listed in the remote interface.</p>	<p>The bean class implements the javax.ejb.SessionBean interface. In addition, it implements all business methods listed in the component interface.</p>
<p>Page 55, last code snippet on page</p> <pre>/* ctx and cart constitute the conversational state */</pre>	<pre>/* The member variable cart constitutes the enterprise bean's conversational state */</pre>

<p>Page 57, third paragraph</p> <p>After serializing the enterprise bean state, the EJB container calls the <code>ejbPassivate()</code> method on the instance.</p>	<p>Before serializing the enterprise bean state, the EJB container calls the <code>ejbPassivate()</code> method on the instance.</p>
<p>Page 59, "What You Need to Do" column for <code>ejbRemove()</code> entry</p> <p>Close resources, if any, and assign corresponding instance fields to null. For example, close sockets, JDBC connections, and so on. For a stateless session bean, this is empty because the container never passivates the bean instance.</p>	<p>Close resources, if any, and assign corresponding instance fields to null. For example, close sockets, JDBC connections, and so on. The instance typically releases the same resources that it releases in the <code>ejbPassivate()</code> method.</p>
<p>Page 60, second bullet</p> <p>The container instantiates the bean using the <code>newInstance</code> method and then calls the methods <code>setSessionContext</code> and <code>ejbCreate</code>. The container also sets the transaction context and security attributes (as set in the deployment descriptor). Now the bean is ready to serve any client.</p>	<p>The container instantiates the bean using the <code>newInstance</code> method and then calls the methods <code>setSessionContext</code> and <code>ejbCreate</code>.</p>
<p>Page 68, first paragraph</p> <p>JNDI is a unified Java API designed to standardized access to a variety of naming and directory services.</p>	<p>JNDI is a unified Java API designed to standardize access to a variety of naming and directory services.</p>
<p>Page 68, first bullet</p> <p>An application-level programming interface (API). APIs are used by the application components to access naming and directory services.</p>	<p>An application-level programming interface (API). This is used by the application components to access naming and directory services.</p>
<p>Page 69, paragraph preceding "Context Operations"</p> <p>The tree-like structure of JNDI is a natural to support subcontexts.</p>	<p>The tree-like structure of JNDI inherently supports subcontexts.</p>

<p>Page 73, last item in Table 4.1</p> <p>Disconnects from the JNDI service, and is used to free resources used by a context.</p>	<p>Disconnects from the JNDI service and frees resources used by a context.</p>
<p>Page 75, replace code snippet on bottom half of page</p>	<pre>Day 4: Demonstration of the usage of JNDI... Connecting to a JNDI service... Connected successfully. Initial context created. Getting Environment Properties... Properties: {java.naming.provider.url=t3://localhost:7001, java.naming.factory.initial=weblogic.jndi.WLInitialContextFactory} Binding a new name: mary to an object: mary@hotmail.com... Object: mary@hotmail.com is bound to name: mary Looking up the name... Found Name= mary, with email= mary@hotmail.com Unbinding the name... Name is unbound successfully! Spanning JNDI context bindings... javax: weblogic.jndi.internal.WLContextImpl: WLContext (javax) jms: weblogic.jndi.internal.WLContextImpl: WLContext (javax.jms) transaction: weblogic.jndi.internal.WLContextImpl: WLContext (javax.transaction)</pre>

	<pre> weblogic: weblogic.jndi.internal.WLContextImpl: WLContext (weblogic) drs: weblogic.jndi.internal.WLContextImpl: WLContext (weblogic.drs) jms: weblogic.jndi.internal.WLContextImpl: WLContext (weblogic.jms) transaction: weblogic.jndi.internal.WLContextImpl: WLContext (weblogic.transaction) rmi: weblogic.jndi.internal.WLContextImpl: WLContext (weblogic.rmi) common: weblogic.jndi.internal.WLContextImpl: WLContext (weblogic.common) jdbc: weblogic.jndi.internal.WLContextImpl: WLContext (weblogic.jdbc) j2ee: weblogic.jndi.internal.WLContextImpl: WLContext (weblogic.j2ee) management: weblogic.jndi.internal.WLContextImpl: WLContext (weblogic.management) security: weblogic.jndi.internal.WLContextImpl: WLContext (weblogic.security) fileSystem: weblogic.jndi.internal.WLContextImpl: WLContext (weblogic.fileSystem) . . . Lookup for the unbound name...error expected **ERROR: An unexpected exception occurred...javax.naming.NameNotFoundException: Unable to 'mary' Resolved: '' Unresolved:'mary' ; remaining name 'mary' Connection to JNDI is closed successfully. </pre>
<p>Page 76, paragraph after "Programmatic Method"</p> <p>The following sample code illustrates the access to a JNDI service from inside your code:</p>	<p>The following sample code illustrates the connect to a JNDI service from inside your code:</p>

<p>Page 80, first paragraph</p> <p>Clients can be calling from the Web tier to the business-tier, or from the client tier to either the Web tier or to the EJB tier.</p>	<p>Clients can be calling from the Web tier to the business-tier, or from the client tier to either the Web tier or to the business-tier.</p>
<p>Page 83, end of first paragraph</p>	<p>(See the "Service Locator Design Pattern" section in Day 15.)</p>
<p>Page 84, last paragraph</p> <p>Try to use composite names such as the DNS name www.sampublishing.com and the Windows/DOS name file://c:/projects/myfile.doc.</p>	<p>Try to use compound names such as the DNS name www.sampublishing.com and the Windows/DOS name file://c:/projects/myfile.doc.</p>
<p>Page 94, headline for Listing 5.7</p> <p>The Full Text of C:\styejb\examples\day05\jboss.xml</p>	<p>The Full Text of day05/jboss.xml</p>
<p>Page 96, code in step 1</p> <p>C:\styejb>setEnvWebLogic.bat</p> <p>C:\styejb>startWebLogic.bat</p>	<p>C:\styejb\examples>setEnvWebLogic.bat</p> <p>C:\styejb\examples>startWebLogic.bat</p>
<p>Page 97, Note at top of page</p> <p>When prompted, enter system as the username and administrator as the password.</p>	<p>When prompted, enter the username (system) and password you chose when you installed WebLogic Server (see Appendix A, "WebLogic Application Server 7.0").</p>
<p>Page 117, Listing 6.7</p> <pre>for(int i = 0; i < 10; i++)</pre>	<pre>for(int i = 0; i < 15; i++)</pre>
<p>Page 124, first paragraph after "Understanding Web Applications"</p> <p>These requests can be in the form of either HTTP PUT or GET actions that are processed by the Web components to generate results back to the client.</p>	<p>These requests can be in the form of either HTTP POST or HTTP GET actions that are processed by the Web components to generate results back to the client.</p>
<p>Page 128, Figure 7.2</p> <p>manifest.xml should be manifest.mf</p>	

<p>Page 149, last paragraph</p> <p>...parameters to the appropriate components, which handle the business logic. Finally, the servlet takes the result back and uses it to provide a response to the user. The servlet usually forwards the response to a JSP to perform a presentation task (see Figure 7.9). Servlets are the link between the client request and the model (EJB and JavaBeans).</p>	<p>...parameters to the appropriate handler, which calls the business logic. Finally, the handler takes the result back and uses it to provide a response to the user. The handler usually forwards the response to a JSP to perform a presentation task (see Figure 7.9). Servlets are the link between the client request and the model (EJB and JavaBeans). The Struts framework of Apache's Jakarta Project is an open source Java framework, which is based on the MVC design pattern and the ideas presented in this section.</p>
<p>Page 158, third paragraph</p> <p>Entity beans typically contain data-related logic that performs a task such as inserting, updating, or removing a customer record in the database.</p>	<p>Entity beans typically contain data-related logic that performs a task such as inserting, updating, or removing a record in the database.</p>
<p>Page 160, caption for Figure 8.3</p> <p>Bean-managed persistence contains the calls that access the database.</p>	<p>Bean-managed persistence bean contains the calls that access the database.</p>
<p>Page 161, paragraph following "Container-Managed Persistence"</p> <p>If your bean has container-managed persistence (CMP),...</p>	<p>With container-managed persistence,...</p>
<p>Page 165, paragraph following "Instance Pool and Instance Cache"</p> <p>Just as with stateless session beans, and EJB container may maintain and instance pool of each type of entity bean.</p>	<p>Just as with stateless session beans, and EJB container may maintain and instance pool for each type of entity bean.</p>
<p>page 172, first entry in Table 8.2</p> <p>SetEntityContext</p>	<p>setEntityContext</p>
<p>Pages 175 and 176, replace "Best Practices section"</p>	<p>Before the introduction of the EJB 2.0 specification, developers often used BMP rather than CMP. The EJB 2.0 specification enabled developers to develop portable applications using CMP that are database independent and free</p>

	<p>of database access code.</p> <p>In addition, containers typically optimize the data access code generated for CMP entity beans. The container can monitor for any change to the in-memory buffer of the bean's data and avoid writing this data to the database in case it did not change. Also, the container can optimize CMP finder methods. To load an entity bean from a database, a BMP makes two calls to the database: The first call loads the primary key via a finder method and the second call loads the entity bean data using the <code>ejbLoad()</code> method. On the other hand, with CMP the container can load the entity data in a single call to the database. Thus the entity beans can give better performance than BMP entity beans because the container has a good hold on CMP entity beans.</p>
Page 176, replace "Summary"	<p>Today you examined entity beans and learned that they are server-side components that represent back-store data in the middle tier. By design, entity beans are persistent and survive any server crashes. They are transactional and share their state with multiple clients.</p> <p>You also learned about both types of persistence: bean-managed and container-managed. With bean-managed persistence, the entity bean contains the calls that access the database. With container-managed persistence, the EJB container is responsible for generating the necessary data access code.</p> <p>The EJB container may maintain an instance pool for each type of entity bean. This saves the precious time of creating and destroying objects. To save the resources such as memory, the EJB container can have an instance cache for each type of entity bean.</p> <p>You also examined the various entity bean methods. The create methods allow clients to create entity beans. The finder methods allow clients to locate entity beans. The container invokes the <code>ejbLoad()</code> and <code>ejbStore()</code> methods to synchronize the</p>

	entity bean's state with the corresponding values in the database. The home methods contain business logic that is not specific to an entity bean instance. Over the next few days, you will examine entity beans in more detail with complete examples.
Page 182, Figure 9.2 Applet box under EJB tier should be named EJB	
Page 198, first paragraph To change this implicit behavior , JDBC provides the method <code>setAutoCommit(false)</code> to set the transaction mode on the Connection object.	To change the implicit behavior of "auto commit," JDBC provides the method <code>setAutoCommit(false)</code> to set the transaction mode on the Connection object.
Page 205, code in Listing 9.3, remove following lines <code>if (stmt != null)</code> <code> stmt.close();</code>	
Page 208, step 2 <code>c:\styejb\examples>startPointbase.bat</code> <code>c:\styejb\examples>setEnvWebLogic.bat</code>	<code>c:\styejb\examples>setEnvWebLogic.bat</code> <code>c:\styejb\examples>startPointbase.bat</code>
Page 209, step 5 In the left pane, expand Services > JDBC.	In the left pane, expand mydomain > Services > JDBC.
Page 209, step 6 In the left pane, expand Services > JDBC.	In the left pane, expand mydomain > Services > JDBC.
Page 209, step 6 For Pool Name, enter styejbPool Figure 9.8 shows the corresponding screen shot.	For Pool Name, enter styejbPool Select Emulate Two-Phase Commit for non-XA driver. Figure 9.8 shows the corresponding screen shot.

<p>Pages 211 and 212, code at bottom and top of pages</p> <p>Trying to drop table STUDENTS...</p> <p>Table STUDENTS is created...</p>	<p>Trying to drop table DAY09_STUDENTS...</p> <p>Table DAY09_STUDENTS is created...</p>
<p>Page 216, paragraph in "Exercises" section</p> <p>Modify the day's example to perform few batch updates on the Student table.</p>	<p>Modify the day's example to perform few batch updates on the DAY09_STUDENTS table.</p>
<p>Page 221, code snippet near top of page</p> <pre>create table students (student_id varchar(64); first_name varchar(64); last_name varchar(64); address varchar(64));</pre>	<pre>create table students (student_id varchar(12) NOT NULL; first_name varchar(15); last_name varchar(15); address varchar(164));</pre>
<p>Page 221, paragraph following code snippet</p> <p>In the case of the JBoss server, we'll use the HyperSonic database.</p>	<p>In the case of the JBoss server, we'll use the Hypersonic database. Note that the database tables were already created in Day 9.</p>
<p>Page 242, remove step 1 at top of page and renumber subsequent steps</p>	<p>Insert text before new step 2</p> <p>The preceding step starts the PointBase database server as well.</p>
<p>Page 242, insert text before second step 2</p> <p>The preceding step starts the default Hypersonic database server as well.</p>	
<p>Page 253, first paragraph after Listing 11.3</p> <p>Note that the ejbCreate() method returns NULL.</p>	<p>Note that the ejbCreate() method returns null.</p>

<p>Page 256, "The WHERE Clause" section</p> <p>The WHERE clause consists of a conditional expression that is used to select or values that satisfy the expression. Therefore, the WHERE clause restricts the results of the query.</p>	<p>The WHERE clause consists of a conditional expression that is used to select objects or values that satisfy the expression. Therefore, the WHERE clause restricts the results of the query. The following sections discuss the various language constructs that can be used in the WHERE clause.</p>
<p>Page 266, first set of steps in "Running the Example" section, remove step 1 and renumber subsequent steps</p> <p>1. Start WebLogic Server in a new command window as follows.</p>	<p>1. Start WebLogic Server and PointBase database server in a new command window as follows.</p>
<p>Page 292, second paragraph in note</p> <p>Instance b is composed of instance c and instance d is composed of c.</p>	<p>Instance b is composed of instance c and instance c is composed of d.</p>
<p>Page 298, Listing 12.15</p> <pre>ClientLineItem item =(ClientLineItem)havax.rmi.PortableRemoteObject.narrow</pre>	<pre>ClientLineItem item =(ClientLineItem)it.next();</pre>
<p>Page 299, last line before "Running the Example" section</p> <p>C:\styejb\examples\day 11>buildJboss.bat</p>	<p>C:\styejb\examples\day 12>buildJboss.bat</p>
<p>Page 299, first set of steps in "Running the Example" section, remove step 1 and renumber subsequent steps</p> <p>1. Start WebLogic Server in a new command window as follows:</p>	<p>1. Start WebLogic Server and PointBase database server in a new command window as follows.</p>
<p>Page 299, last paragraph on page</p> <p>The JBoss server automatically starts the default HyperSonic database.</p>	<p>The JBoss server automatically starts the default Hypersonic database.</p>

<p>Page 309, sentence immediately before Table 13.1</p> <p>Table 13.1 summarizes the JMS common interfaces for both the PTP model and Pub/Sub model interfaces.</p>	<p>Table 13.1 summarizes the JMS common interfaces for both the PTP model and Pub/Sub model interfaces, and whether they support concurrent use.</p>
<p>Page 309, sentence immediate before Table 13.2</p> <p>Table 13.2 provides a brief definition of these JMS interfaces, and whether they support concurrent use.</p>	<p>Table 13.2 provides a brief definition of these JMS interfaces.</p>
<p>Page 315, code line in "Step 7: Close the Connection"</p> <p>QConn.close();</p>	<p>qConn.close();</p>
<p>Page 318, code snippet in "Step 7: Close the Connection"</p> <p>tCon.close();</p>	<p>tConn.close();</p>
<p>Pages 320 and 323, swap Figures 13.5 and 13.6</p>	
<p>Page 343, remove step 1 and renumber subsequent steps</p> <p>1. Start WebLogic Server in a new command window as follows:</p>	<p>1. Start WebLogic Server and PointBase database server in a new command window as follows.</p>
<p>Page 391, code in Note</p> <p>UserTransaction utx = ctx.getUserTransssaction();</p>	<p>UserTransaction utx = ctx.getUserTranssaction();</p>
<p>Pages 400 and 401, replace code in Listing 16.6</p>	<pre><?xml version="1.0"?> <!DOCTYPE ejb-jar PUBLIC '-//Sun Microsystems, Inc.//DTD Enterprise JavaBeans 2.0//EN' 'tyyp://java.sun.com/dtd/ejb-jar_2_0.dtd'> <ejb-jar> <enterprise-beans></pre>

```
<session>
  <ejb-name>UserManager</ejb-name>
  <home>day16.UserManagerHome</home>
  <remote>day16.UserManagerBean</remote>
  <ejb-class>day16.UserManagerBean</ejb-class>
  <session-type>Stateful</session-type>
  <transaction-type>Bean</transaction-type>
  <resource-ref>
    <res-ref-name>jms/ConnectionFactory</res-ref-name>
    <res-type>javax.jms.QueueConnectionFactory</res-type>
    <res-auth>Container</res-auth>
    <res-sharing-scope>Shareable</res-sharing-scope>
  </resources-ref>
  <resource-env-ref>
    <resource-env-ref-name>jdbc/styejbDB</resource-env-ref-name>
    <resource-env-ref-type>javax.sql.DataSource</resource-env-ref-
type>
  </resource-env-ref>
  <resource-env-ref>
    <resource-env-ref-name>jms/RegistrarQ</resource-env-ref-name>
    <resource-env-ref-type>javax.jms.Queue</resource-env-ref-type>
  </resource-env-ref>
</session>
<message-driven>
  <ejb-name>RegistrarMDB</ejb-name>
  <ejb-class>day16.RegistrarMDB</ejb-class>
```

	<pre> <resource-env-ref-name>jdbc/styejbDB</resource-env-ref-name> <jndi-name>java:/DefaultDS</jndi-name> </resource-env-ref> <resource-env-ref> <resource-env-ref-name>jms/RegistrarQ</resource-env-ref-name> <jndi-name>queue/RegistrarQ</jndi-name> </resource-env-ref> <resource-ref> <res-ref-name>jms/ConnectionFactory</res-ref-name> <jndi-name>ConnectionFactory</jndi-name> </resource-ref> </session> <message-drive> <ejb-name>RegistrarMDB</ejb-name> <destination-jndi-name>queue/RegistrarQ</destination-jndi-name> </message-drive> </enterprise-beans> </jboss> </pre>
<p>Page 401, replace code in Listing 16.7</p>	<pre> <?xml version="1.0"?> <!DOCTYPE weblogic-ejb-jar PUBLIC '-//BEA Systems, Inc.//DTD WebLogic 6.0.0 EJB//EN' 'http://www.bea.com/servers/wls600/dtd/weblogic-ejb-jar.dtd'> <weblogic-ejb-jar> </pre>

```
<weblogic-enterprise-bean>
  <ejb-name>UserManager</ejb-name>
  <reference-descriptor>
    <resource-description>
      <res-ref.name>jms/ConnectionFactory</res-ref.name>
      <jndi-name>javax.jms.QueueConnectionFactory</jndi-name>
    </resource-description>
    <resource-env-description>
      <res-env-ref.name>jdbc/styejbDB</res-env-ref.name>
      <jndi-name>jdbc.styejbDB</jndi-name>
    </resource-env-description>
    <resource-env-description>
      <res-env-ref.name>jms/RegistrarQ</res-env-ref.name>
      <jndi-name>RegistrarQ</jndi-name>
    </resource-env-description>
  </reference-descriptor>
  <jndi-name>day16/UserManagerHome</jndi-name>
</weblogic-enterprise-bean>
<weblogic-enterprise-bean>
  <ejb-name>RegistrarMDB</ejb-name>
  <message-driven-descriptor>
    <pool>
      <max-beans-in-free-pool>5</max-beans-in-free-pool>
      <initial-bean-in-free-pool>1</initial-beans-in-free-pool>
    </pool?>
  </message-driven-descriptor>
</weblogic-enterprise-bean>
```

	<pre><destination-jndi-name>RegistrarQ</destination-jndi-name> </message-drive-descriptor> <jndi-name>day16.RegistrarMDB</jndi-name> </weblogic-enterprise-bean> </weblogic-ejb-jar></pre>
Page 402, replace Listing 16/8	<pre><?xml version="1.0" encoding="UTF-8"?> <jboss> <enterprise-beans> <session> <ejb-name>UserManager</ejb-name> <jndi-name>day16/UserManagerHome</jndi-name> <resource-env-ref> <resource-env-ref-name>jdbc/styejbDB</resource-env-ref-name> <jndi-name?java:/DefaultDS</jndi-name> </resource-env-ref> <resource-env-ref> <resource-env-ref-name>jms/RegistrarQ</resource-env-ref-name> <jndi-name>queue/RegistrarQ</jndi-name> </resource-env-ref> <resource-ref> <res-ref-name>jms/ConnectionFactory</res-ref-name> <jndi-name>ConnectionFactory</jndi-name> </resource-ref> </session></pre>

	<pre> <message-driven> <ejb-name>RegistrarMDB</ejb-name> <destination-jndi-name>queue/RegistrarQ</destination-jndi-name> </message-driven> </enterprise-beans> </jboss> </pre>
Page 412, header for second column in Table 17.1 EJB1.method	EJB.methodA
Page 419, swap columns of third row All methods javax.transaction.UserTransaction	javax.transaction.UserTransaction All methods
Page 433, last sentence Where applicable, exercise solutions are offered on the book's Web site at www.ejb21days.com .	Where applicable, exercise solutions are offered on the book's Web site at www.samspublishing.com .
Page 466, first bullet Identity: Who is supposed to grant access to resources?	Identity: Who is supposed to be granted access to resources?
Page 498, second code line Address address[1] = address1;	Address address[1] = address2;
Page 530, second paragraph For example, the enrollments table consists of the columns enrollment_id (primary key), student_id (foreign key to student_id in the students table), and course_id (foreign key to course_id in the courses table).	or example, the day21_enrollments table consists of the columns enrollment_id (primary key), student_id (foreign key to student)id in the students table), and course_id (foreign key to course_id in the courses table).
Page 532, first paragraph in "Packaging and Deploying the Application" These steps assume that you configured OrderVerifierTopic as discussed on Day 14, "Developing Message-Driven Beans," and urMailSession as discussed on Day 20, Implementing JavaMail in EJB Applications."	You need to configure Day21_OrderVerifierTopic, similar to the one discussed in Day 14, "Developing Message-Driven Beans," and urMailSession as discussed on Day 20, Implementing JavaMail in EJB Applications." Please refer to README.PDF in

	the examples/day21 folder for detailed instructions.
Page 533, remove step 1 and renumber subsequent steps 1. Start WebLogic server and PointBase server in a new command window as follows:	
Page 536, step 2 The remaining steps are similar to those listed for WebLogic Server's steps 4 through 7.	The remaining steps are similar to those listed for WebLogic Server's steps 3 through 6.
Page 538 In addition, add a Collection <code>getStudentEntrollments()</code> method to the Student component. This method invokes the method <code>Collection findStudentEnrollment(String studentId)</code> of the Enrollment component.	In addition, add a Collection <code>getStudentEntrollments()</code> method to the StudentFacade component. This method invokes the method <code>Collection findStudentEnrollments(String studentId)</code> of the Enrollment component.
Page 541, insert new text immediately following chapter title	Before you start, we assume you already downloaded the examples from the Web site http://www.samspublishing.com and unzipped the downloaded file into the <code>c:\</code> directory. This will create the examples under the directory <code>C:\styejb/examples</code> . For detailed and up-to-date information, please refer to the document <code>Setup Weblogic.pdf</code> in the examples directory.
Page 541, paragraph following "Downloading and Installing WebLogic 7.0" BEA provides you with a fully functional trial version of this server for 30 days (from the installation date), which provides ample time for running the examples within the book at the suggested pace of 21 days.	BEA provides you with a fully functional trial version of this server for 90 days (from the installation date), which provides ample time for running the examples within the book at the suggested pace of 21 days.
Page 542, replace numbered steps	<ol style="list-style-type: none"> 1. Double-click on the downloaded file <code>net_platform701_win32.exe</code>; the Installer will prompt you with the Welcome window. Click on the Next button to proceed. 2. When the BEA License Agreement window appears, it will ask you if you agree with the terms of the License. Please read the license and select the Yes radio button if you agree with

the terms of the license. Then click the Next button.

3. Now you will be presented with a window to select where you want to choose your BEA home directory. Because this installation can be used to install multiple products from BEA, the BEA home directory is used as a high level directory for all other products. Go ahead and select the default directory C:\bea and click on the Next button. If you wish to select another directory, feel free to do so, but remember that you have enough disk space on the selected drive.

4. Now you can choose the installation type. Because BEA offers multiple products to try within the same installation, the typical installation will install all these products, which requires about 430 MB. Choose Custom Installation and click the Next button.

5. In the Choose Components window (see Figure A.1), select the first WebLogic Server product, which includes the first three components. Unselect all other components. Click on the Next button to proceed.

*****Insert Figure A.1 Afig01*****

Figure A.1

Components selection dialog.

6. Now you need to tell the Installer where you want the temporary files to be stored (in c:\temp) and if you want the Installer to get rid of them after the installation is complete. If you want The Installer to remove the downloaded temporary files, choose this option. Otherwise, you can choose to delete the files after the installation is complete. Click the Next button.

7. Now the installation process will start downloading the files from BEA's site. Make sure that you are still connected to the Internet at this time. Be patient: This process will take 45 minutes if you are using a DSL connection (45 KB/sec).

	<p>8. The next screen will automatically appear (see Figure A.2). Now you need to choose the WebLogic server home directory. This directory is of prime importance in order to successfully run the examples in the book. Enter c:\bea\weblogic701 and then click the Next button. If you have decided to change the BEA home directory in step 3, you need to use subdirectory weblogic701. (Notice the Installer suggests choosing weblogic700 as the default.)</p> <p>***Insert Figure A.2 Afig02***</p> <p>Figure A.2 Product Directory dialog.</p> <p>9. Now the Installer will install the binary files from the downloaded archives in the temporary directory. This should be faster than the previous step.</p> <p>10. After the installation is complete, select the Yes, Run Configuration Wizard to Create My Application Domain option, then click the Next button.</p>
<p>Page 542, first paragraph after "Configuring WebLogic 7.0"</p> <p>When the installation is complete, you must configure the WebLogic Server environment. Front the Start menu, select Programs, BEA WebLogic Platform 7.0, and choose the Configuration Wizard application. You'll be presented with the screen in Figure A.1.</p>	<p>Now you must configure the Weblogic server. Scroll through the template selections until you find WLS Domain, then select it. Choose mydomain, then click Next to proceed. You'll be presented with the screen in Figure A.3.</p>
<p>Pages 543 through 545, renumber figures and figure references by two</p>	

<p>Page 544, last sentence</p> <p>Now you can choose End Configuration Wizard, and click the Next button to finish.</p>	<p>Now you can choose End Configuration Wizard, and click the Done button to finish.</p>
<p>Page 545, first paragraph</p> <p>After you've entered your server's configuration parameters in the setEnvWeblogic.bat file located in the examples' root directory (as per the instructions contained in README.txt located in the same directory), you can start running the examples provided with this book.</p>	<p>After you've entered your server's configuration parameters in the Setup_WebLogic.bat file located in the examples' root directory (as per the instructions contained in examples located in the same directory), you can start running the examples provided with this book.</p>
<p>Page 547, insert new text immediately following chapter title</p>	<p>Before you start, we assume you already downloaded the examples from the Web site http://www.samspublishing.com and unzipped the downloaded file into the c:\ directory. This will create the examples under the directory c:\styejb\examples. For detailed and up-to-date information, please refer to the Setup_JBoss.pdf document in the examples directory.</p>
<p>Page 547, first paragraph</p> <p>JBoss is an open-source, J2EE-compliant application server, which is currently distributed for free.</p>	<p>JBoss is an open-source, J2EE application server, which is currently distributed under LGPL license.</p>
<p>Page 548, add to end of second Note</p>	<p>After downloading, install it under the directory C:\java131. This directory will be your JAVA_HOME.</p>
<p>Page 548, new text for "Configuring JBoss 3.0" section</p>	<p>After the installation is complete, you must edit a few configuration parameters before you start running the JBoss server. In order to run the examples successfully, you need to open setEnvJBoss.bat, in directory C:\styejb/examples, by using a text editor (such as Notepad). Edit the first few lines with the following information (shown in bold):</p> <pre> set JBOSS_HOME=c:\jboss3.0.1 set JAVA_HOME=c:java131 set STYEJB_HOME=c:\styejb\examples </pre> <p>After making these changes, save the file and exit the editor. If you have already followed our recommendation in</p>

	<p>the preceding steps, you probably don't need to change anything in these settings. If you have chosen different installation directories, you need to set appropriate values for the JBOSS_HOME, JAVA_HOME, and STYEJB_HOME environment variables.</p>
<p>Page 549, replace text in "Running and Testing JBoss 3.0"</p>	<p>To run and test JBoss, open a command window and execute the following commands:</p> <pre>c:\>cd styejb\examples c:\styejb\examples>setEnvJBoss.bat c:\styejb\examples>startJBoss.bat</pre> <p>When JBoss server starts, it will print a message such as</p> <pre>14:23:51,532 INFO [Server] JBoss (MX MicroKernel) [3.0.1 Date:200208122231] Started in 0m:23s:3ms</pre> <p>Now open your browser and enter the URL http://localhost:8080/jmx-console. This should display something similar to the JBoss Console in Figure B.1.</p> <p>Congratulations! Now the setup is complete and you are ready to run the examples in the book.</p>
<p>Page 560, second paragraph</p> <p>The class also contains the operations addCourses and getCourses.</p>	<p>The class also contains the operations addCourses, getCourses, and empty.</p>

This errata sheet is intended to provide updated technical information. Spelling and grammar misprints are updated during the reprint process, but are not listed on this errata sheet.