

# Oracle and Java Development

Copyright © 2002 by Sams Publishing

International Standard Book Number: 0-672-32117-3

## 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 print number of your book, view the copyright page. The print number is the right-most number on the line below the "First Printing" line. For example, the following indicates that this is the 1st printing of this title and it was printed in August 2001.

**First Printing: August 2001**

04 03 02 01      4 3 2 1

Pg	Error	Correction
50	<p>Last text on the page:</p> <p>The output of the above query is as follows:</p> <pre> ADD_ID  LINE1          LINE2          CITY    ST    ZIP -----  - </pre>	<p>The output of the above query is as follows:</p> <pre> ADD_ID  LINE1          LINE2  CITY    ST  ZIP -----  - -- 1 30 Carlton Avenue # 32D Sunnyvale CA 97848 </pre>
61	<p>Last paragraph on the page, second sentence:</p> <p>...For example, the following <del>packages</del> might need to be imported while using the Oracle extensions to JDBC such as using advanced data types such as BLOB, and so on.</p> <pre> import oracle.jdbc.driver.*; import oracle.sql.*; </pre>	<p>...For example, the following <b>package</b> might need to be imported while using the Oracle extensions to JDBC such as using advanced data types such as BLOB, and so on.</p> <pre> import oracle.sql.*; </pre>
132	<p>Sentence before the numbered list:</p> <p>The following are the steps involved in obtaining the <del>BLOB</del> locator from the database table:</p>	<p>The following are the steps involved in obtaining the <b>CLOB</b> locator from the database table:</p>
133	<p>Line of code after the first paragraph:</p> <p>Alternatively, use the getClob() method of the java.sql.Clob class to retrieve the CLOB locator into a java.sql.Clob object. This is shown here:</p> <pre> <del>Blob</del> clob_data = rset.getClob(1); </pre>	<p>Alternatively, use the getClob() method of the java.sql.Clob class to retrieve the CLOB locator into a java.sql.Clob object. This is shown here:</p> <pre> <b>Clob</b> clob_data = rset.getClob(1); </pre>
135	<p>Section of code at the top of the page, beginning with 5th line:</p> <pre> // outStream.close(); bufReader.close(); // conn.commit(); </pre>	<p>Add a line:</p> <pre> // <b>outStream.flush();</b> outStream.close(); bufReader.close(); // conn.commit(); </pre>
138	<p>Paragraph after the numbered list at top of page:</p> <p>We will use the lob_tab table created earlier to insert BFILE data into the table. Additionally, a directory has to be created to specify the location of the bfiles residing on the O/S. This is done in SQL as follows:</p>	<p>We will use the lob_tab table created earlier to insert BFILE data into the table. Additionally, a directory has to be created to specify the location of the bfiles residing on the O/S. <b>This must be created using a user account with System or DBA privilege.</b> This is done in SQL as follows:</p>
139	<p>15 lines from the bottom of page:</p> <pre> // read nextChunk from the <del>BLOB</del> in a loop </pre>	<pre> // read nextChunk from the <b>BFILE</b> in a loop </pre>
179	<p>Code at the top of the page, lines 1-3, move the } at the end of the 2nd line:</p> <pre> else { System.out.println("The employee with empno 9998 already exists."); } #sql {<del>ROLLBACK</del>};Oracle.close(); </pre>	

	<p>Change to the following:</p> <pre> else { System.out.println("The employee with empno 9998 already exists."); #sql {<b>ROLLBACK</b>};}Oracle.close(); </pre>	
196	<p>2nd paragraph after the tip:</p> <p>We will use the lob_tab table created earlier to insert BFILE data into the table. Additionally, a directory has to be created to specify the location of the bfiles residing on the O/S. This is done in SQL as follows:</p>	<p>We will use the lob_tab table created earlier to insert BFILE data into the table. Additionally, a directory has to be created to specify the location of the bfiles residing on the O/S. <b>This must be done using a user account with System or DBA privilege.</b> This is done in SQL as follows:</p>
281	<p>Line 17 of the code:</p> <pre> String error = "DepartmentBean (" + deptNo + "<del> not found</del>"; </pre>	<pre> String error = "DepartmentBean (" + deptNo + ") <b>not</b> found"; </pre>
391	<p>5th line from the bottom of page:</p> <pre> out.println("(Please enter department details and click on <del>INSERT, UPDATE</del>, or DELETE. )" ); </pre>	<pre> out.println("(Please enter department details and click on <b>ADD, MODIFY</b>, or DELETE. )" ); </pre>
403	<p>8th section of text from the top:</p> <p>The reason for compiling DeptBean.java along with DepartmentS.java is that the DepartmentS servlet uses the DeptBean for instantiation and invokes the Bean methods InsertDept(), UpdateDept(), and DeleteDept() on this object instance...</p>	<p><b>Note that the DeptBean.java file must be modified to obtain a connection to the database using the defaultConnection() method instead of using an explicit connection.</b> The reason for compiling DeptBean.java along with DepartmentS.java is that the DepartmentS servlet uses the DeptBean for instantiation and invokes the Bean methods InsertDept(), UpdateDept(), and DeleteDept() on this object instance...</p>
423	<p>15th line from the bottom:</p> <pre> out.println("(Please enter department details and click on <del>INSERT, UPDATE</del> or DELETE.)" ); </pre>	<pre> out.println("(Please enter department details and click on <b>ADD, MODIFY</b> or DELETE.)" ); </pre>
428	<p>Step 2:</p> <p>2. Run the publishjsp command from the session shell. This automatically translates the .jsp source into a .java source and then compiles it into a .class file. Then it optionally hotloads the page implementation class if the <del>hotload</del> option is specified for publishjsp....</p>	<p>2. Run the publishjsp command from the session shell. This automatically translates the .jsp source into a .java source and then compiles it into a .class file. Then it optionally hotloads the page implementation class if the <del>hotload</del> option is <b>enabled</b> for publishjsp....</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.