Sams Teach Yourself J2EE 21 Days

International Standard Book Number: 0-672-32384-2

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 or from the use of the CD or programs accompanying it.

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: April 2002

06 05 04 03   10 9 8 7 6 5 4

Misprint	Correction
Page 29 **Installing** J2EE SDK 1.3	**Preparing to Install** J2EE SDK 1.3
Page 29, end of next to last paragraph, remove sentence: If you the JAVA_HOME variable is not defined, you must define it now as follows.	
Page 29, insert new text after next to last paragraph. See end of errata page for new text.	
page 29, insert new section head before last paragraph: In**stalling J2EE SDK 1.3 on Windows NT/2000**	
page 30, step 5 Assuming that you installed the **J2EE SDK 1.3** on the C: drive using the default directory name, you will set the JAVA_HOME variable to C:\j2sdkee1.3	Assuming that you installed the **J2EE SDK 1.3.1** on the C: drive using the default directory name, you will set the JAVA_HOME variable to C:\jdk1.3.1 **If you have installed the J2EE SDK in a different directory, please set the JAVA_HOME variable to that directory.**
page 30, delete text beginning with If you are using Linux or Unix... and ending with ....profile file in your home directory.	
page 30, delete paragraph beining with For Linux/Unix users...	
page 30, next to last paragraph Now downlad the J2EE SDK in **the format appropriate to your system** from http://java.sun.com/j2ee/sdk_1.3/index.html.	Now downlad the J2EE SDK in **for Windows NT/2000** from http://java.sun.com/j2ee/sdk_1.3/index.html.

page 31, remove text beginning with Solaris bullet through 3. Add the J2EE classes to your CLASSPATH.	
page 31, paragraph in middle of page You have already been shown how to define variables and change your path **for your environment (Windows users use the Control Panel and Linux/Unix users add variable definition lines to .profile),** so making these changes will be straightforward.	You have already been shown how to define variables and change your path **Windows,** so making these changes will be straightforward.
page 31, second set of steps, past the middle of the page 1. Define the following environment variable J2EE_HOME=c:\**j2eesdk1**.3	1. Define the following environment variable J2EE_HOME=c:\**j2sdkee1**.3
page 31, second set of steps, second code line for step 3 .;%J2EE_HOME%/lib/j2ee.jar;%J2EE_HOME%/lib/locale	**CLASSPATH=**.;%J2EE_HOME%/lib/j2ee.jar;%J2EE_HOME%/lib/locale
pages 31-32, delete text beginning with Linux/Unix users must add the following... through the line of code right before That is it!... Also delete the Note immediately before "Starting the J2EE Reference Implementation (RI)"	
page 32, insert new text before "Starting the J2EE Reference Implementation (RI)"; see end of errata page for new text	
page 44, the "Components: Message Beans" heading should come before the second paragraph on the page, which begins "A message-driven EJB..."	
page 72, Figure 2.16 The crows foot arrow between the Skill and ApplicantSkill tables should be in the opposite direction, pointing down toward Skill from ApplicantSkill.	
page 104, the paragraph below the second note, first sentence The rest of the sub-section assumes that you have some	The rest of the sub-section assumes that you have some knowledge of Linux and OpenLDAP (at the very least, that

knowledge of Linux and OpenLDAP (at the very least, that you have successfully installed OpenLDAP on a **Unix** server). Also, in the last line of that paragraph there should not be a space before the 2.0.15.	you have successfully installed OpenLDAP on a **Linux** server).
page 108, first two lines of the page Run the following command to check access to the LDAP server. java JNDILookupAny "cn=Manager, o=Agency, **c=us** "	Run the following command to check access to the LDAP server. java JNDILookupAny "cn=Manager, o=Agency, **c=us**" Note there should not be a space before the final quote.
page 116, the paragraph below the Note, ignore/delete the phrase bolded below. With the name registered and the HTTP server running, the client can look up the bound object **without having the Book class file in the search path**.	

page 310, code at the top of the page, semicolon in last line should be a "j"
SELECT OBJECT(;)
FROM Job AS ;
WHERE **;.**location.name = ?1
	SELECT OBJECT(;)
	FROM Job AS ;
	WHERE **j.**location.name = ?1

page 358, last sentence on the page Indeed, if you have used the Adapter design pattern, you'll recognize that the ResourceAdapter is well named, combining two orthogonal interfaces into a single **instance**.	Indeed, if you have used the Adapter design pattern, you'll recognize that the ResourceAdapter is well named, combining two orthogonal interfaces into a single **class**.
page 405, caption to Figure 9.5 Using deploytool to configure **installation**.	Using deploytool to configure **a jms queue**.
page 407, step 3 3. Establish a QueueSession for this connection. In this case, the QueueSession has transactions set to false (transactions are covered later) and **AUTO ACKNOWLEDGE** of receipt of messages...	3. Establish a QueueSession for this connection. In this case, the QueueSession has transactions set to false (transactions are covered later) and **AUTO_ACKNOWLEDGE** of receipt of messages...

page 432, figure 10.1 Add a label to the bottom of the figure, next to the arrow: **ejbRemove()**	
page 445, the paragraph just above Listing 10.3 You can use **the code or** the PTPSender program described in Day 9 to send the message.	You can use the PTPSender program described in Day 9 to send the message.

page 450, step 4 at bottom of page
4. In both AgencyBean.java and RegisterBean.java, add the **following to ejbRemove() to close down the MessageSender.**
try {
messageSender.close();
}
catch (JMSException ex) {
	4. In both AgencyBean.java and RegisterBean.java, add the **following to close down the MessageSender.**
	ejbRemove()
	try {
	messageSender.close();
	}
	catch (JMSException ex) {

page 475, URL halfway down the page http://www.samspublishing.com/images/topnav/logo-bottom.gif	Should be http://www.contentmaster.com/inpublisher/Upload_Images/sams/gif
page 511, 7th entry in the table <FORM> Delimit a form. This is used to send user input.	<FORM> Delimit a form. This is used to send user input **to the server**.
page 527, Table 12.4 servlet-class The servlet's **fully qualified** classname. Same page, third paragraph, which begins "The DTD contains...," last line, Appendix C is not on the CD-ROM; it actually appears in the book.	servlet-class The servlet's classname.
page 530, second line of the code at the top of the page, there should be no space between res. and sendRedirect. **res.sendRedirect**...	

page 578, Note If a bean property has only a getter method, it is read-only: a write-only **method** only has a setter method. A property is read-write if it has both getter and setter methods.	If a bean property has only a getter method, it is read-only: a write-only **property** only has a setter method. A property is read-write if it has both getter and setter methods.
page 609, Table 14.3, first entry TagSupport implements Tag Extend this class for tags that do not have a body or do not interact with the tag body.	TagSupport implements Tag, **IterationTag** Extend this class for tags that do not have a body or do not interact with the tag body.
page 610, third line in paragraph after "The doAfterBody() Method" Delete the phrase in parentheses, "(those that extend BodyTagSupport)"	
page 611, third line in paragraph after "The doInitBody() Method" Delete the phrase in parentheses, "(those that extend BodyTagSupport)"	
622, first paragraph on the page To keep the example code simple, it always stores the bean against the name cust. To be more flexible, you could add an additional parameter (such as **beanName**) to be used to allow the customer bean variable name to specified by the JSP developer. In this case, you would set the TLD <variable> information to use this parameter using the <name-from-attribute>**beanName**<name-from-attribute/> tag.	To keep the example code simple, it always stores the bean against the name cust. To be more flexible, you could add an additional parameter (such as **id**) to be used to allow the customer bean variable name to specified by the JSP developer. In this case, you would set the TLD <variable> information to use this parameter using the <name-from-attribute>**id**<name-from-attribute/> tag.
page 657, first line of last paragraph In real applications, symmetric **algorithms** use sophisticated algorithms...	In real applications, symmetric **encryption** uses sophisticated algorithms...
page 660, second paragraph after "Digital Certificates" heading The official specification for the **X.500 Directory Service** is available...	The official specification for the **X.509 Digital Certificates** is available...

page 707, Table 16.1, entry line 2, the </tag> is in the wrong column. 2 <tag attribute="text"> Attribute and a **text</tag>**body	2 <tag attribute="text"> Attribute and a **text body** **</tag>**
page 710, in the Note Appendix C is not on the CD-ROM; it's in the actual book.	
page 714, last line of the paragraph below the "Namespaces" heading ...application differentiate between these different types of **book** descriptions?	...application differentiate between these different types of **job** descriptions?
page 731, Table 16.10, the third and last entries have text in the wrong columns insertBefore(Node newNode, Inserts the newNode before **Node refNode)**the existing refNode replaceChild(Node newNode, Replaces the oldNode with**Node oldNode)**newNode in the child NodeList.	insertBefore(Node newNode, **Node refNode)** Inserts the newNode before the existing refNode replaceChild(Node newNode, **Node oldNode)** Replaces the oldNode with newNode in the child NodeList.
page 731, Table 16.10 continued, same problem setNodeValue(String Set the value of this node, **nodeValue)**depending on its type; see Table 16.10. setPrefix(**java.lang.**String Set the namespace prefix of **prefix)**this node.	setNodeValue(String **nodeValue)** Set the value of this node, depending on its type; see Table 16.10. setPrefix(String **prefix)** Set the namespace prefix of this node.
page 762, single code line near the bottom of the page, there should be no space between " and child <xsl:template match=" **child**::skill">	<xsl:template match="**child**::skill">
page 780, third set of double code lines at the top of the page </xsl:**when**> </xsl:choose>	</xsl:**otherwise**> </xsl:choose>

page 989, paragraph just above the "Declarations" heading The root element is <jobSummary...</jobSummary>. The <job> element has two attributes and enclosed **elements**. Last sentence on the page, ignore the "preceded" at the end of the first line.	The root element is <jobSummary...</jobSummary>. The <job> element has two attributes and enclosed **subelements**.
page 993, Table C.3, fourth entry down ? Element is optional	? Element is optional **but if present, must appear only once**

New text to be inserted on page 29

Instructions for setting up the JAVA_HOME variable and installing J2EE SDK 1.3 on Windows and Unix are described in the following two sub sections. On each platform you will perform the following steps:

1. Define the JAVA_HOME variable if it is not already defined.
2. Add the JDK bin directory to your program search path.
3. Download the J2EE SDK.
4. Extract and install the J2EE SDK archive.
5. Define the J2EE SDK environment variables
6. Add the J2EE SDK bin directory to your program search path.
7. Add the J2EE classes to your CLASSPATH.

Note: All the documentation for the J2EE utility programs and class files is contained in the J2EE SDK download bundle. You will find the documentation in the docs subdirectory of the J2EE installation directory.

New text to be inserted on page 32

Installing J2EE SDK 1.3 on Linux

If you are using Linux and the JAVA_HOME environment variable does not exist, you can set it with the following command (you must be using Bash or another compatible shell such as the Bourne shell):

```
JAVA_HOME=/usr/local/jdk1.3.1
export JAVA_HOME
```

This example assumes you have installed the Sun Microsystems' JDK 1.3.1 in /usr/local.

Typically, you will add these variable definitions to your login environment by adding the same two lines to the .profile file in your home directory.

Finally, you should ensure that the JDK bin directory is in your search path (again this should already be configured on your workstation).

If your search path does not contain the JDK bin directory, add the following line to your .profile file:

```
PATH=$PATH:$JAVA_HOME/bin
```

Now download the Linux J2EE SDK from http://java.sun.com/j2ee/sdk_1.3/index.html. You should download the J2EE SDK to a temporary directory because the installation process will ask you where to install the SDK.

Change directories to the required parent directory for the J2EE SDK (for example, /usr/local) and extract the download bundle using the following command:

```
        tar - xzvf <download_directory>/j2sdkee-1_3_01-linux.tar.gz
```

You have already been shown how to define variables and change your path setting so making the following changes to your login environment will be straightforward. You must add the following to your .profile (assuming the J2EE SDK was installed in /usr/local/j2eesdk1.3):

1. Define the following environment variable:

   ```
   J2EE_HOME=/usr/local/j2eesdk1.3
   ```

2. Update the PATH variable for J2EE SDK:

   ```
   PATH=$PATH:$J2EE_HOME/bin
   ```

3. Add the J2EE JAR files to the CLASSPATH variable:

   ```
   CLASSPATH=$CLASSPATH:$J2EE_HOME/lib/j2ee.jar:$J2EE_HOME/lib/locale
   ```

If your CLASSPATH variable is not currently defined, you must ensure that it includes the current directory so the full setting will be

```
CLASSPATH=.:$J2EE_HOME/lib/j2ee.jar:$J2EE_HOME/lib/locale
```

That is it! You are now ready to start using the J2EE SDK.

Installing J2EE SDK 1.3 On Solaris

If you are using Solaris and the JAVA_HOME environment variable does not exist, you can set it with the following command (you must be using the Korn shell another compatible shell such as the Bourne shell):

```
JAVA_HOME=/opt/jdk1.3.1
export JAVA_HOME
```

This example assumes you have installed the Sun Microsystems' JDK 1.3.1 in /opt.

Typically, you will add these variable definitions to your login environment by adding the same two lines to the .profile file in your home directory.

Finally, you should ensure that the JDK bin directory is in your search path (again this should already be configured on your workstation).

If your search path does not contain the JDK bin directory, add the following line to your .profile file:

```
PATH=$PATH:$JAVA_HOME/bin
```

Now download the Solaris J2EE SDK from http://java.sun.com/j2ee/sdk_1.3/index.html. You should download the J2EE SDK to a temporary directory because the installation process will ask you where to install the SDK.

Next, issue the following commands to make the download bundle executable and run the installation:

```
chmod a+x ./j2sdkee-1_3__01-solsparc.sh
./j2sdkee-1_3__01-solsparc.sh
```

You have already been shown how to define variables and change your path setting so making the following changes will be straightforward. You must add the following to your .profile (assuming the J2EE SDK was installed in /opt/j2eesdk1.3):

1. Define the following environment variable:

   ```
   J2EE_HOME=/opt/j2eesdk1.3
   ```

2. Update the PATH variable for J2EE SDK:

```
PATH=$PATH:$J2EE_HOME/bin
```

3. Add the J2EE JAR files to the CLASSPATH variable:

   ```
   CLASSPATH=$CLASSPATH:$J2EE_HOME/lib/j2ee.jar:$J2EE_HOME/lib/locale
   ```

   If your CLASSPATH variable is not currently defined, you must ensure that it includes the current directory so the full setting will be

   ```
   CLASSPATH=.:$J2EE_HOME/lib/j2ee.jar:$J2EE_HOME/lib/locale
   ```

That is it! You are now ready to start using the J2EE SDK.

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.