BEA WebLogic Enterprise

Using the WebLogic EJB Deployer

WebLogic Enterprise 5.0
Document Edition 5.0
December 1999
Using the WebLogic EJB Deployer

<table>
<thead>
<tr>
<th>Document Edition</th>
<th>Date</th>
<th>Software Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.0</td>
<td>December 1999</td>
<td>BEA WebLogic Enterprise 5.0</td>
</tr>
</tbody>
</table>
About This Document

What You Need to Know ................................................................. i
e-docs Web Site ............................................................................. ii
How to Print the Document.......................................................... ii
Related Information ........................................................................ ii
Contact Us! .................................................................................... iii
Documentation Conventions ....................................................... iii

1. What is the WebLogic EJB Deployer?

Key Capabilities ........................................................................... 1-1
Supported File Formats ............................................................... 1-2

2. Using the WebLogic EJB Deployer

Navigating the WebLogic EJB Deployer .................................. 2-2
Starting the WebLogic EJB Deployer .......................................... 2-2
Setting the CLASSPATH .............................................................. 2-2
Running the Startup Command ................................................. 2-3
Navigating Menus ....................................................................... 2-3
File menu ..................................................................................... 2-3
Edit menu ...................................................................................... 2-4
Help menu .................................................................................... 2-4
Using the Toolbar ........................................................................ 2-4

Configuring the WebLogic EJB Deployer ............................... 2-6
Configuring Environment Settings ............................................ 2-6
Configuring EJBC Settings .......................................................... 2-6
Exiting the WebLogic EJB Deployer .......................................... 2-7

EJB Deployment Tasks ................................................................. 2-9
Summary of Deployment Steps ................................................................. 2-9
Opening ejb-jar Files ............................................................................... 2-10
Navigating the ejb-jar Contents List ......................................................... 2-11
Navigating a Deployment Descriptor .................................................... 2-11
Changing Deployment Descriptor Properties ......................................... 2-12
Saving Changes to ejb-jar Files ............................................................. 2-13
    Saving the Current ejb-jar File .......................................................... 2-14
    Saving All Open ejb-jar Files ......................................................... 2-14
    Saving the Current ejb-jar File to a New Name ............................... 2-14
Deploying ejb-jar Files ......................................................................... 2-15
Closing ejb-jar Files ............................................................................. 2-16
    Closing the Current ejb-jar File...................................................... 2-16
    Closing All Open ejb-jar Files ....................................................... 2-16
Clearing Messages.................................................................................. 2-17

3. Panels

    ejb-jar panel ...................................................................................... 3-2
    Description tab ................................................................................ 3-2
    Security tab .................................................................................... 3-3
    Module Initialization tab ................................................................ 3-3
Bean Properties Panel ........................................................................... 3-4
EJB panel .............................................................................................. 3-5
Method panel ........................................................................................ 3-5
Persistence panel .................................................................................. 3-6
Security Role References panel ........................................................... 3-7
Environment Entries panel ................................................................... 3-7
EJB Reference panel ............................................................................. 3-8
Resource Factory Reference panel ....................................................... 3-9
JDBC panel .......................................................................................... 3-9
    Configuration tab ........................................................................... 3-10
    Field-Column Map tab ................................................................. 3-10
    Finders tab ................................................................................... 3-11
File panel ............................................................................................. 3-12
4. **EJB Deployment Descriptor Reference**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Deployment Descriptors</td>
<td>4-2</td>
</tr>
<tr>
<td>EJB Roles and Deployment Descriptors</td>
<td>4-2</td>
</tr>
<tr>
<td>File Formats</td>
<td>4-3</td>
</tr>
<tr>
<td>DOCTYPE Declaration</td>
<td>4-3</td>
</tr>
<tr>
<td>Summary of Deployment Descriptor Elements</td>
<td>4-3</td>
</tr>
<tr>
<td>Description of Deployment Descriptor Elements</td>
<td>4-7</td>
</tr>
<tr>
<td>assembly-descriptor</td>
<td>4-7</td>
</tr>
<tr>
<td>cmp-field</td>
<td>4-7</td>
</tr>
<tr>
<td>container-transaction</td>
<td>4-7</td>
</tr>
<tr>
<td>description</td>
<td>4-7</td>
</tr>
<tr>
<td>display-name</td>
<td>4-8</td>
</tr>
<tr>
<td>ejb-class</td>
<td>4-8</td>
</tr>
<tr>
<td>ejb-jar</td>
<td>4-9</td>
</tr>
<tr>
<td>ejb-link</td>
<td>4-9</td>
</tr>
<tr>
<td>ejb-name</td>
<td>4-9</td>
</tr>
<tr>
<td>ejb-ref</td>
<td>4-10</td>
</tr>
<tr>
<td>ejb-ref-name</td>
<td>4-10</td>
</tr>
<tr>
<td>ejb-ref-type</td>
<td>4-10</td>
</tr>
<tr>
<td>enterprise-beans</td>
<td>4-10</td>
</tr>
<tr>
<td>entity</td>
<td>4-11</td>
</tr>
<tr>
<td>env-entry</td>
<td>4-11</td>
</tr>
<tr>
<td>env-entry-name</td>
<td>4-11</td>
</tr>
<tr>
<td>env-entry-type</td>
<td>4-12</td>
</tr>
<tr>
<td>env-entry-value</td>
<td>4-12</td>
</tr>
<tr>
<td>field-name</td>
<td>4-12</td>
</tr>
<tr>
<td>home</td>
<td>4-13</td>
</tr>
<tr>
<td>large-icon</td>
<td>4-13</td>
</tr>
<tr>
<td>method</td>
<td>4-13</td>
</tr>
<tr>
<td>method-intf</td>
<td>4-14</td>
</tr>
<tr>
<td>method-name</td>
<td>4-14</td>
</tr>
<tr>
<td>method-param</td>
<td>4-14</td>
</tr>
<tr>
<td>method-params</td>
<td>4-15</td>
</tr>
<tr>
<td>method-permission</td>
<td>4-15</td>
</tr>
<tr>
<td>persistence-type</td>
<td>4-15</td>
</tr>
</tbody>
</table>
5. **WebLogic EJB Extensions Reference**

   About the WebLogic EJB Extensions .......................................................... 5-2
   Summary of WebLogic EJB Extensions ................................................... 5-3
   Description of WebLogic EJB Extension Elements .............................. 5-5
   attribute-map .......................................................................................... 5-5
   attribute-map-entry ................................................................................ 5-5
   bean-field-name ....................................................................................... 5-5
   finder-descriptor ..................................................................................... 5-6
   jndi-name .................................................................................................. 5-7
   module-initializer-class ........................................................................ 5-8
   module-initializer-class-name ............................................................... 5-8
   persistence-store-descriptor ............................................................... 5-8
   persistence-store-directory-root ........................................................... 5-9
   persistence-store-jdbc ........................................................................... 5-10
   persistence-store-file ............................................................................ 5-11
   pool-name ................................................................................................ 5-11
   session ....................................................................................................... 5-12
   session-type ............................................................................................ 5-12
   small-icon .................................................................................................. 5-13
   transaction-type ....................................................................................... 5-13
   trans-attribute .......................................................................................... 5-13
   ID .............................................................................................................. 5-13

...
Index

query-grammar ................................................................. 5-12
table-column-name ....................................................... 5-14
table-name ................................................................. 5-14
trans-timeout-seconds .................................................. 5-14
weblogic-deployment-params ...................................... 5-14
weblogic.ejb-extensions ............................................. 5-15
weblogic.enterprise-bean .......................................... 5-15
weblogic.version ......................................................... 5-15
About This Document

This document explains how to use the WebLogic EJB Deployer to examine, configure, and deploy Enterprise JavaBeans (EJBs) for use with the WebLogic Enterprise (WLE) server.

This document covers the following topics:

- Chapter 1, “What is the WebLogic EJB Deployer?,” introduces the WebLogic EJB Deployer.
- Chapter 2, “Using the WebLogic EJB Deployer,” describes how to use the WebLogic EJB Deployer.
- Chapter 3, “Panels,” provides a reference to each panel in the WebLogic EJB Deployer.
- Chapter 4, “EJB Deployment Descriptor Reference,” provides a reference to Sun Microsystem’s standard XML elements in the EJB deployment descriptor.

What You Need to Know

This document is intended mainly for EJB deployers who need to examine, configure, and deploy EJBs in the WLE environment. It assumes a familiarity with EJB applications and Sun Microsystem’s Enterprise JavaBeans Specification 1.1 (Public Release 2, dated October 18, 1999).
e-docs Web Site

The BEA WebLogic Enterprise product documentation is available on the BEA corporate Web site. From the BEA Home page, click the Product Documentation button or go directly to the “e-docs” Product Documentation page at http://e-docs.beasys.com.

How to Print the Document

You can print a copy of this document from a Web browser, one file at a time, by using the File—>Print option on your Web browser.

A PDF version of this document is available on the WebLogic Enterprise documentation Home page on the e-docs Web site (and also on the documentation CD). You can open the PDF in Adobe Acrobat Reader and print the entire document (or a portion of it) in book format. To access the PDFs, open the WebLogic Enterprise documentation Home page, click the PDF Files button, and select the document you want to print.

If you do not have the Adobe Acrobat Reader, you can get it for free from the Adobe Web site at http://www.adobe.com/.

Related Information

For more information about CORBA, Java 2 Enterprise Edition (J2EE), BEA TUXEDO, distributed object computing, transaction processing, C++ programming, and Java programming, see the WLE Bibliography in the WebLogic Enterprise online documentation.
Contact Us!

Your feedback on the BEA WebLogic Enterprise documentation is important to us. Send us e-mail at docsupport@beasys.com if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the WebLogic Enterprise documentation.

In your e-mail message, please indicate that you are using the documentation for the BEA WebLogic Enterprise 5.0 release.

If you have any questions about this version of BEA WebLogic Enterprise, or if you have problems installing and running BEA WebLogic Enterprise, contact BEA Customer Support through BEA WebSupport at www.beasys.com. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface text</strong></td>
<td>Indicates terms defined in the glossary.</td>
</tr>
<tr>
<td>Ctrl+Tab</td>
<td>Indicates that you must press two or more keys simultaneously.</td>
</tr>
<tr>
<td>Convention</td>
<td>Item</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
</tr>
<tr>
<td><em>italics</em></td>
<td>Indicates emphasis or book titles.</td>
</tr>
</tbody>
</table>
| monospace text | Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.  
*Examples:*

```c
#include <iostream.h> void main ( ) the pointer psz 
chmod u+w * 
\tux\data\ap 
.doc 
tux.doc 
BITMAP 
float
```

| monospace boldface text | Identifies significant words in code.  
*Example:*

```c
void commit ( )
```

| monospace italic text | Identifies variables in code.  
*Example:*

```c
String expr
```

| UPPERCASE TEXT | Indicates device names, environment variables, and logical operators.  
*Examples:*

```c
LPT1
SIGNON
OR
```

| () | Indicates a set of choices in a syntax line. The braces themselves should never be typed. |
|[-v] [-o name ] [-f file-list]... [-l file-list]... | Indicates optional items in a syntax line. The brackets themselves should never be typed.  
*Example:*

```c
buildobjclient [-v] [-o name ] [-f file-list]...
[-l file-list]...
```
<table>
<thead>
<tr>
<th>Convention Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.</td>
</tr>
<tr>
<td>... Indicates one of the following in a command line:</td>
</tr>
<tr>
<td>- That an argument can be repeated several times in a command line</td>
</tr>
<tr>
<td>- That the statement omits additional optional arguments</td>
</tr>
<tr>
<td>- That you can enter additional parameters, values, or other information</td>
</tr>
<tr>
<td>The ellipsis itself should never be typed.</td>
</tr>
<tr>
<td>Example:</td>
</tr>
<tr>
<td>buildobjcclient [-v] [-o name ] [-f file-list]... [-l file-list]...</td>
</tr>
<tr>
<td>Indicates the omission of items from a code example or from a syntax line.</td>
</tr>
<tr>
<td>The vertical ellipsis itself should never be typed.</td>
</tr>
</tbody>
</table>
Using the WebLogic EJB Deployer
1 What is the WebLogic EJB Deployer?

The WebLogic EJB Deployer is a graphical tool that Deployers can use to examine, configure, and deploy Enterprise JavaBeans (EJBs) for use with the WebLogic Enterprise (WLE) server. The WebLogic EJB Deployer fully supports Sun’s Enterprise JavaBeans Specification 1.1 (Public Release 2 dated October 18, 1999). For more information about deploying EJBs in the WLE environment, see Building and Deploying Enterprise JavaBeans (EJBs).

Key Capabilities

Using the WebLogic EJB Deployer, you can:

- Open ejb-jar files containing one or more EJBs. Bean Providers must separately create the ejb-jar files that the WebLogic EJB Deployer uses for deployment.
- View the contents of ejb-jar files.
- Examine deployment descriptors of ejb-jar files, which include:
  - **Standard XML elements** as defined in the Enterprise JavaBeans Specification, including assembly descriptor information, bean references, and security information.
  - **Optional WLE XML extension elements** that provide additional support for container-managed persistence, module initialization classes, and JNDI registration.
What is the WebLogic EJB Deployer?

- Optionally, configure the deployment descriptor settings that Deployers are allowed to modify, in compliance with Sun’s *Enterprise JavaBeans Specification*. In addition, Deployers can modify WLE XML extension elements to configure persistence, specify resource verification, and so on.

- Optionally, create a deployable ejb-jar file for subsequent deployment to the WLE EJB Container.

- Save changes in the same ejb-jar file or create a new one.

### Supported File Formats

The WebLogic EJB Deployer supports the ejb-jar formats listed in Table 1-1.

**Table 1-1  Supported File Formats**

<table>
<thead>
<tr>
<th>ejb-jar Format</th>
<th>Contains</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard</td>
<td>Bean binaries, Sun’s standard XML elements (for EJB version 1.1 compliance), and optional assembly descriptor XML elements.</td>
</tr>
<tr>
<td>extended</td>
<td>Contents of a standard ejb-jar file, plus optional WLE XML extensions to the deployment descriptor (in a separate XML file).</td>
</tr>
<tr>
<td>deployable</td>
<td>Contents of an extended ejb-jar file, plus additional classes and interfaces generated by the WebLogic EJB Deployer for the EJB Container.</td>
</tr>
</tbody>
</table>

The WebLogic EJB Deployer imports all three formats and exports extended and deployable file formats.
This section describes how to use the WebLogic EJB Deployer to configure and deploy Enterprise JavaBeans (EJBs) to use with your WebLogic Enterprise server. It contains the following topics:

- Navigating the WebLogic EJB Deployer
- EJB Deployment Tasks

For more information about deploying EJBs in the WLE environment, see Building and Deploying Enterprise JavaBeans (EJBs).
Navigating the WebLogic EJB Deployer

This section describes how to navigate the WebLogic EJB Deployer. It contains the following topics:

- Starting the WebLogic EJB Deployer
- Navigating Menus
- Using the Toolbar
- Configuring the WebLogic EJB Deployer
- Exiting the WebLogic EJB Deployer

Starting the WebLogic EJB Deployer

This topic describes how to start the WebLogic EJB Deployer in the Unix or Windows environment.

Setting the CLASSPATH

Before you start the WebLogic EJB Deployer, make sure that the CLASSPATH contains the WebLogic classes (M3.jar and weblogicaux.jar), the jar file you want to open, and any others classes used by your EJBs. Listing 2-1 provides an example (on Windows NT) in which the first two command lines show user-specific settings and the final command line shows setting the CLASSPATH.

Listing 2-1   Setting the CLASSPATH Before Starting the WebLogic EJB Deployer

```plaintext
set TUXDIR=D:\bea_prod\wledir
set WLECP=%TUXDIR%\udataobj\java\jdk
set CLASSPATH=%WLECP%\M3.jar;%WLECP%\weblogicaux.jar;jarfilepath;%CLASSPATH%

where jarfilepath is the path to the JAR file you want to open.
```
Running the Startup Command

To start the WebLogic EJB Deployer:

```bash
$ java weblogic.EJBDeployWizard options
```

where **options** are:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-help</td>
<td>Prints help text.</td>
</tr>
<tr>
<td>-laf [basic</td>
<td>metal</td>
</tr>
</tbody>
</table>

The main screen appears. Figure 2-1 shows an example of the Main Screen (with an opened ejb-jar file).

Figure 2-1  Main Screen

Navigating Menus

The WebLogic EJB Deployer contains the following menus:

- File menu
- Edit menu
- Help menu

File menu

The File menu contains the following commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Opening ejb-jar Files</td>
</tr>
<tr>
<td>Save</td>
<td>Saving Changes to ejb-jar Files</td>
</tr>
</tbody>
</table>
Using the WebLogic EJB Deployer

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save As</td>
<td>Saving the Current ejb-jar File to a New Name</td>
</tr>
<tr>
<td>Save All</td>
<td>Saving Changes to ejb-jar Files</td>
</tr>
<tr>
<td>Close</td>
<td>Closing ejb-jar Files</td>
</tr>
<tr>
<td>Close All</td>
<td>Closing ejb-jar Files</td>
</tr>
<tr>
<td>Exit</td>
<td>Exiting the WebLogic EJB Deployer</td>
</tr>
</tbody>
</table>

### Edit menu

The Edit menu contains the following commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deploy</td>
<td>Deploying ejb-jar Files</td>
</tr>
<tr>
<td>Options</td>
<td>Configuring the WebLogic EJB Deployer</td>
</tr>
</tbody>
</table>

### Help menu

The Help menu contains the following commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using</td>
<td>Displays Help</td>
</tr>
<tr>
<td>About</td>
<td>Displays the splash screen</td>
</tr>
</tbody>
</table>

### Using the Toolbar

Figure 2-3 shows the toolbar.

Figure 2-2  Toolbar
Navigating the WebLogic EJB Deployer

The Toolbar contains the following command buttons:

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Folder" /></td>
<td>Open</td>
<td>Opening ejb-jar Files</td>
</tr>
<tr>
<td><img src="image" alt="Save" /></td>
<td>Save</td>
<td>Saving Changes to ejb-jar Files</td>
</tr>
<tr>
<td><img src="image" alt="Save As" /></td>
<td>Save As</td>
<td>Saving the Current ejb-jar File to a New Name</td>
</tr>
<tr>
<td><img src="image" alt="Save All" /></td>
<td>Save All</td>
<td>Saving Changes to ejb-jar Files</td>
</tr>
<tr>
<td><img src="image" alt="Close" /></td>
<td>Close</td>
<td>Closing ejb-jar Files</td>
</tr>
<tr>
<td><img src="image" alt="Close All" /></td>
<td>Close All</td>
<td>Closing ejb-jar Files</td>
</tr>
<tr>
<td><img src="image" alt="Settings" /></td>
<td>Deploy</td>
<td>Deploying ejb-jar Files</td>
</tr>
</tbody>
</table>

To run a command, click the associated command button.
Configuring the WebLogic EJB Deployer

You can configure environment and ejbc settings for the WebLogic EJB Deployer.

Configuring Environment Settings

To configure the environment settings for the WebLogic EJB Deployer:

1. Click the Options button on the toolbar, or choose Options from the Edit menu.

   The WebLogic EJB Deployer Properties dialog box appears. Figure 2-3 shows an example of the WebLogic EJB Deployer Properties dialog box.

   **Figure 2-3  WebLogic EJB Deployer Properties Dialog Box**

2. Double-click the plus (+) sign to the left of the deployer.

3. Click Environment.

   The Environment Properties pane appears. Figure 2-4 shows an example of the Environment Properties pane.

   **Figure 2-4  Environment Properties**

   The Environment Properties pane contains the following options:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>saveWindowLocations</td>
<td>Determines whether the WebLogic EJB Deployer saves window attributes (true, the default), or not (false)</td>
</tr>
<tr>
<td>tmpdir</td>
<td>Specifies the temporary directory for saving window attributes (default is \tmp).</td>
</tr>
</tbody>
</table>

4. Make the changes you want, if any, and then click OK.

Configuring EJBC Settings

You can configure the settings that the BEA ejbc compiler uses when it generates container classes. To configure these settings:

1. Click the Options button on the toolbar, or choose Options from the Edit menu.
Navigating the WebLogic EJB Deployer

The WebLogic EJB Deployer Properties dialog box appears. Figure 2-5 shows an example of the WebLogic EJB Deployer Properties dialog box.

**Figure 2-5  WebLogic EJB Deployer Properties Dialog Box**

2. Double-click the plus (+) sign to the left of the deployer.

3. Click ejbc.

   The EJBC Properties pane appears. Figure 2-6 shows an example of the EJBC Properties pane.

**Figure 2-6  EJBC Properties**

The EJBC Properties dialog box contains the following options:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>keepgenerated</td>
<td>If True is selected, ejbc saves the generated Java files for deployment in the ejb-jar file. If False is selected, ejbc performs a compliance check but does not update the ejb-jar file.</td>
</tr>
<tr>
<td>compiler</td>
<td>Specifies the Java compiler to use for generating container classes.</td>
</tr>
</tbody>
</table>

4. Make the changes you want, if any.

5. Click OK.

For more information about the ejbc compiler, see the *Command Reference*.

Exiting the WebLogic EJ B Deployer

To exit the WebLogic EJB Deployer:

1. Choose Exit from the File menu.

   If the WebLogic EJB Deployer detects that changes to any open ejb-jar files have not been saved, it prompts you whether to save it (Yes) or not (No). Figure 2-7 shows an example of the Save prompt.
2. Click Yes or No as appropriate.
EJB Deployment Tasks

This section describes the following EJB deployment tasks.

- Opening ejb-jar Files
- Navigating the ejb-jar Contents List
- Navigating a Deployment Descriptor
- Changing Deployment Descriptor Properties
- Saving Changes to ejb-jar Files
- Deploying ejb-jar Files
- Closing ejb-jar Files
- Clearing Messages

**Note:** Not all tasks are required to deploy EJBs.

**Summary of Deployment Steps**

Deploying an EJB involves the following basic steps:

1. Create the standard deployment descriptor and optionally, the extended deployment descriptor. The Bean Provider is responsible for creating these files separately. The WebLogic EJB Deployer then uses these files to create deployable ejb-jar files.

2. Start the WebLogic EJB Deployer.

3. Load an ejb-jar file.

4. Optionally, change settings in the deployment descriptor.

5. Deploy the ejb-jar file, which generates the container classes, adds the container classes to the ejb-jar file, and saves any changes.

7. Copy the ejb-jar file to a target Java server.
8. Update the UBBCONFIG with the name and location of this ejb-jar file.

You can also use WebLogic EJB Deployer to simply view the settings in the ejb-jar file’s deployment descriptor.

Opening ejb-jar Files

To open an ejb-jar file:

1. Click the Open button on the toolbar, or choose Open from the File menu.

   The Open EJB-JAR File dialog box appears. Figure 2-8 shows an example of the Open EJB-JAR File dialog box.

   **Figure 2-8  Open EJB-JAR File Dialog Box**

   The Open EJB-JAR File dialog box contains the following options:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Look In</td>
<td>Selects a directory.</td>
</tr>
<tr>
<td>Directory List</td>
<td>Displays a list of directories and files in the selected directory.</td>
</tr>
<tr>
<td>File Name</td>
<td>Type or double-click the name of the ejb-jar file you want to open.</td>
</tr>
<tr>
<td>File of Type</td>
<td>Select a file type.</td>
</tr>
</tbody>
</table>

2. Select the source directory that contains the ejb-jar file that you want to open.
3. Select or type the name of the ejb-jar file you want to open.
4. Click Open.

The WebLogic EJB Deployer opens the ejb-jar file, validates its contents, and displays it.
Navigating the ejb-jar Contents List

The ejb-jar Contents List displays all the elements of the selected ejb-jar file in a tree hierarchy.

You can expand or collapse a list in order to show or hide subordinate items in the hierarchy. A plus (+) or minus (−) sign to the left of an entry indicates whether an item contains subordinate (child) items. To show or hide a subordinate item, click the parent name.

Navigating a Deployment Descriptor

To navigate properties in a deployment descriptor (both standard and optional extended XML elements), click its corresponding entry in the ejb-jar Contents List. The panel associated with the entry appears in the frame to the right.

Table 2-1 lists the panels in the WebLogic EJB Deployer.

Table 2-1 Panels in the WebLogic EJB Deployer

<table>
<thead>
<tr>
<th>Panel Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ejb-jar panel</td>
<td>Displays information about the ejb-jar file, including the following tabs:</td>
</tr>
<tr>
<td></td>
<td>- Description tab</td>
</tr>
<tr>
<td></td>
<td>- Security tab</td>
</tr>
<tr>
<td></td>
<td>- Module Initialization tab</td>
</tr>
<tr>
<td>Bean Properties Panel</td>
<td>Displays a list of EJBs associated with the selected ejb-jar file,</td>
</tr>
<tr>
<td></td>
<td>including the name and type of each EJB.</td>
</tr>
<tr>
<td>EJB panel</td>
<td>For the selected EJB, displays the bean class, home interface, remote</td>
</tr>
<tr>
<td></td>
<td>interface, a summary of bean characteristics, and a description.</td>
</tr>
<tr>
<td>Method panel</td>
<td>For the selected method, displays the transaction attribute, a list of</td>
</tr>
<tr>
<td></td>
<td>allowed callers, and a description.</td>
</tr>
</tbody>
</table>
Changing Deployment Descriptor Properties

Sun Microsystems’s *Enterprise JavaBeans Specification 1.1* defines which EJB Role is responsible for maintaining each attribute in the deployment descriptor. The WebLogic EJB Deployer allows you to change only the standard information that Deployers are authorized to change, in addition to certain WLE XML extension elements.
Table 2-2 lists the information that you can change using the WebLogic EJB Deployer.

### Table 2-2 Editable Information in the WebLogic EJB Deployer

<table>
<thead>
<tr>
<th>Name</th>
<th>Panel</th>
<th>XML Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction Attribute</td>
<td>Method panel</td>
<td>trans-attribute</td>
</tr>
<tr>
<td>Application Role</td>
<td>Security Role References panel</td>
<td>role-link</td>
</tr>
<tr>
<td>Value</td>
<td>Environment Entries panel</td>
<td>env-entry-value</td>
</tr>
<tr>
<td>Table name</td>
<td>Configuration tab on the JDBC panel</td>
<td>table-name</td>
</tr>
<tr>
<td>Connection Pool Name</td>
<td>Configuration tab on the JDBC panel</td>
<td>pool-name</td>
</tr>
<tr>
<td>Table column</td>
<td>Field-Column Map tab on the JDBC panel</td>
<td>table-column-name</td>
</tr>
<tr>
<td>Expression</td>
<td>Finders tab on the JDBC panel</td>
<td>query-grammar</td>
</tr>
<tr>
<td>Module Initialization Class</td>
<td>Module Initialization tab</td>
<td>module-initializer-class</td>
</tr>
<tr>
<td>Persistent directory root</td>
<td>File panel</td>
<td>persistence-store-directory-root</td>
</tr>
</tbody>
</table>

**Note:** If you change a property value, the WebLogic EJB Deployer validates the value when you change focus. If the change fails the validation test, the WebLogic EJB Deployer displays a message in the Errors tab and displays the bean in a red font. The WebLogic EJB Deployer will not allow you to save any changes or deploy the EJB until you have fixed the problem.

### Saving Changes to ejb-jar Files

This section includes the following topics:

- Saving the Current ejb-jar File
Saving All Open ejb-jar Files

Saving the Current ejb-jar File

Saving the Current ejb-jar File to a New Name

2-14 Using the WebLogic EJB Deployer
2. Select a target directory.

3. Type or select a different name for the target ejb-jar file.

4. Click Save.

   If the WebLogic EJB Deployer detects any errors in the ejb-jar file, it displays a message in the Errors tab. You must fix the problem before saving the file.

### Deploying ejb-jar Files

To deploy an ejb-jar file:

1. Click the Deploy button on the toolbar, or choose Deploy from the Edit menu.

   The Generate container classes dialog box appears. Figure 2-10 shows an example of the Generate container classes dialog box.

   **Figure 2-10   Generate Container Classes Dialog Box**

2. For each EJB, enter the name for the EJB home that is registered in the Global WLE JNDI namespace.

   This name corresponds to the \texttt{jndi-name} element, which is a WLE XML extension to the deployment descriptor.

   To allow Java clients to perform a lookup on the JNDI name for the EJB home and gain access to the object, the name of the EJB home must be registered in the Global WLE JNDI namespace. The JNDI name must be unique among all global factory names in a WLE domain, including EJB homes, CORBA factories, and RMI named objects.

3. Click Deploy.

   The WebLogic EJB Deployer saves changes to the ejb-jar file automatically.

   Next, the WebLogic EJB Deployer generates the container classes for the ejb-jar file, displaying a message dialog box to indicate progress and displaying status messages in the Messages tab. Figure 2-11 shows an example of the message dialog box.

   **Figure 2-11   Message Dialog Box for Generating Container Classes**

   To interrupt this process, click Cancel.
Closing ejb-jar Files

This section includes the following topics:

- Closing the Current ejb-jar File
- Closing All Open ejb-jar Files

Closing the Current ejb-jar File

To close the ejb-jar file in the current window:

1. Click the Close button on the toolbar, or choose Close from the File menu.
   
   If the WebLogic EJB Deployer detects that changes have not been saved to the current ejb-jar file, it prompts you whether to save it (Yes) or not (No). Figure 2-12 shows an example of the Save prompt.

   Figure 2-12   Save Prompt

2. Click Yes or No as appropriate.

Closing All Open ejb-jar Files

To close all open ejb-jar files:

1. Click the Close All button on the toolbar, or choose Close All from the File menu.

2. If the WebLogic EJB Deployer detects that changes to any open ejb-jar files have not been saved, it prompts you whether to save it (Yes) or not (No). Figure 2-13 shows an example of the Save prompt.

   Figure 2-13   Save Prompt

3. Click Yes or No as appropriate.
Clearing Messages

To clear messages from the Messages or Errors tab:

1. Right-click the mouse anywhere in the Messages or Errors tab.
2. Click the Clear Messages button.

   The WebLogic EJB Deployer clears the text.

**Note:** When you open a new ejb-jar file, the WebLogic EJB Deployer clears the message text automatically.
Using the WebLogic EJB Deployer
Panels display deployment descriptor information for the selected component in the ejb-jar Contents List. Table 3-1 describes the panels in the WebLogic EJB Deployer.

Table 3-1 Panels in the WebLogic EJB Deployer

<table>
<thead>
<tr>
<th>Panel Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ejb-jar panel</td>
<td>Information about the ejb-jar file, including the following panels:</td>
</tr>
<tr>
<td></td>
<td>■ Description tab</td>
</tr>
<tr>
<td></td>
<td>■ Security tab</td>
</tr>
<tr>
<td></td>
<td>■ Module Initialization tab</td>
</tr>
<tr>
<td>Bean Properties Panel</td>
<td>Displays a list of EJBs associated with the selected ejb-jar file,</td>
</tr>
<tr>
<td></td>
<td>including the name and type of each EJB.</td>
</tr>
<tr>
<td>EJB panel</td>
<td>For the selected EJB, displays the bean class, home interface, remote</td>
</tr>
<tr>
<td></td>
<td>interface, a summary of bean characteristics, and a description.</td>
</tr>
<tr>
<td>Method panel</td>
<td>For the selected method, displays the transaction attribute, a list of</td>
</tr>
<tr>
<td></td>
<td>allowed callers, and a description.</td>
</tr>
<tr>
<td>Persistence panel</td>
<td>For the selected EJB, displays persistence information: the primary key</td>
</tr>
<tr>
<td></td>
<td>class, a list of container-managed fields, and a description.</td>
</tr>
<tr>
<td>Security Role References</td>
<td>For the selected EJB, displays the security role reference and</td>
</tr>
<tr>
<td>panel</td>
<td>link to a security role.</td>
</tr>
<tr>
<td>Environment Entries panel</td>
<td>For the selected EJB, displays the name, type, value, and a description of</td>
</tr>
<tr>
<td></td>
<td>one or more environment entries.</td>
</tr>
</tbody>
</table>
Panels

Using the WebLogic EJB Deployer

ejb-jar panel

The ejb-jar panel displays the following tabs that contain information about the selected ejb-jar file:

- Description tab
- Security tab
- Module Initialization tab

Description tab

Figure 3-1 shows an example of the Description tab.

Figure 3-1 Description Tab
The Description tab displays text that describes the selected ejb-jar file:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>XML Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Text description of the ejb-jar file.</td>
<td>description</td>
</tr>
</tbody>
</table>

**Security tab**

Figure 3-2 shows an example of the Security tab.

**Figure 3-2 Security Tab**

The Security tab displays the following security information for the selected ejb-jar file:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>XML Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Roles</td>
<td>Displays a list of any security roles that are available in this environment.</td>
<td>security-role</td>
</tr>
<tr>
<td>Description</td>
<td>Text description.</td>
<td>description</td>
</tr>
</tbody>
</table>

System administrators define security roles using the TUXEDO tpaddgrp feature. For more information on managing groups, see the BEA TUXEDO documentation.

**Module Initialization tab**

Figure 3-3 shows an example of the Module Initialization tab.

**Figure 3-3 Module Initialization Tab**
The Module Initialization tab displays the following module initialization information for the selected ejb-jar file:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>XML Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Name</td>
<td>Name of the server implementation class for handling the initialize and release methods.</td>
<td><code>module-initializer-class-name</code></td>
</tr>
</tbody>
</table>

You can edit the Class Name in the WebLogic EJB Deployer. To add a new class, click Add and specify the Class Name in the new row. To delete an existing class, select the row and click Delete.

**Note:** These XML elements are WLE XML extensions.

### Bean Properties Panel

Figure 3-4 shows an example of the Bean Properties panel.

**Figure 3-4  Bean Properties Panel**

The Bean Properties panel displays a list of EJBs associated with the selected ejb-jar file, with the following information for each EJB:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>XML Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Icon</td>
<td>Icon associated with the EJB.</td>
<td><code>small-icon</code> or default</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the EJB.</td>
<td><code>ejb-name</code></td>
</tr>
<tr>
<td>Type</td>
<td>Type of EJB (Entity or Session).</td>
<td><code>entity</code> <code>session</code></td>
</tr>
</tbody>
</table>
**EJB panel**

Figure 3-5 shows an example of the EJB panel.

**Figure 3-5  EJB Panel**

The EJB panel displays the following information about the selected EJB:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>XML Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bean class</td>
<td>Fully qualified name of the EJB’s class.</td>
<td>ejb-class</td>
</tr>
<tr>
<td>Home interface class</td>
<td>Fully qualified name of the EJB’s home interface.</td>
<td>home</td>
</tr>
<tr>
<td>Remote interface class</td>
<td>Fully qualified name of the EJB’s remote interface.</td>
<td>remote</td>
</tr>
<tr>
<td>Summary</td>
<td>Summary text about an EJB: the bean type, session type (for session beans), persistence type (for entity beans), transaction type, and whether an entity bean is re-entrant.</td>
<td>entity, session, session-type, persistence-type, transaction-type, reentrant</td>
</tr>
<tr>
<td>Description</td>
<td>Text description of the EJB.</td>
<td>description</td>
</tr>
</tbody>
</table>

**Method panel**

Figure 3-6 shows an example of the Method panel.

**Figure 3-6  Method Panel**
The Method panel displays the following information about the selected method:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>XML Element</th>
</tr>
</thead>
</table>
| Transaction Attribute | Specifies the transaction attribute for the EJB. Select of the following values:
  - NotSupported
  - Supports
  - Required
  - RequiresNew
  - Mandatory
  - Never | trans-attribute |
| Allowed callers  | Specifies the role name(s) allowed to invoke the method. | role-name |
| Description      | Text description of the method. | description |

You can select a different Transaction Attribute from the list using the WebLogic EJB Deployer.

**Persistence panel**

Figure 3-7 shows an example of the Persistence panel.

**Figure 3-7 Persistence Panel**

The Persistence panel displays the following persistence information for the selected EJB:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>XML Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary key type</td>
<td>Fully qualified name of an entity bean’s primary key class.</td>
<td>prim-key-class</td>
</tr>
<tr>
<td>Primary key field</td>
<td>Primary key field, if specified.</td>
<td>primkey-field</td>
</tr>
<tr>
<td>Fields</td>
<td>Container-managed fields.</td>
<td>cmp-field</td>
</tr>
</tbody>
</table>
Using the WebLogic EJB Deployer

**Security Role References panel**

Figure 3-8 shows an example of the Security Role References panel.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>XML Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Name of a container-managed field.</td>
<td>field-name</td>
</tr>
<tr>
<td>Type</td>
<td>Java data type associated with the field.</td>
<td>N/A</td>
</tr>
<tr>
<td>Description</td>
<td>Text description.</td>
<td>description</td>
</tr>
</tbody>
</table>

You can edit the Application Role field using the WebLogic EJB Deployer.

**Environment Entries panel**

Figure 3-9 shows an example of the Environment Entries panel.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>XML Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Referenced Role</td>
<td>Name of a security role.</td>
<td>role-name</td>
</tr>
<tr>
<td>Application Role</td>
<td>Links a security role reference to a defined role.</td>
<td>role-link</td>
</tr>
<tr>
<td>Description</td>
<td>Text description of the security role.</td>
<td>description</td>
</tr>
</tbody>
</table>

Using the WebLogic EJB Deployer 3-7
The Environment Entries panel displays the following environment entry (\texttt{env-entry}) information for the selected EJB:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>XML Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of an EJB’s environment entry.</td>
<td>\texttt{env-entry-name}</td>
</tr>
<tr>
<td>Type</td>
<td>Java type of the environment entry value that is expected by the EJB’s code.</td>
<td>\texttt{env-entry-type}</td>
</tr>
<tr>
<td>Value</td>
<td>Value of the EJB’s environment entry.</td>
<td>\texttt{env-entry-value}</td>
</tr>
<tr>
<td>Description</td>
<td>Text description.</td>
<td>\texttt{description}</td>
</tr>
</tbody>
</table>

You can edit the Value field using the WebLogic EJB Deployer.

**EJB Reference panel**

Figure 3-7 shows an example of the EJB Reference panel.

**Figure 3-10  EJB Reference Panel**

The EJB Reference panel displays a list of the following EJB reference (\texttt{ejb-ref-name}) information for the selected EJB:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>XML Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home interface class</td>
<td>Home interface of the EJB reference.</td>
<td>\texttt{ejb-ref}</td>
</tr>
<tr>
<td>Remote interface class</td>
<td>Remote interface of the EJB reference.</td>
<td>\texttt{ejb-ref}</td>
</tr>
<tr>
<td>Linked bean JNDI name</td>
<td>Links this EJB reference to another EJB in the ejb-jar file.</td>
<td>\texttt{ejb-link}</td>
</tr>
<tr>
<td>Description</td>
<td>Text description.</td>
<td>\texttt{description}</td>
</tr>
</tbody>
</table>
Resource Factory Reference panel

Figure 3-11 shows an example of the Resource Factory Reference panel.

**Figure 3-11 Resource Factory Reference Panel**

The Resource Factory Reference panel displays the following resource factory reference (res-ref-name) information for the selected EJB:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>XML Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource factory type</td>
<td>Type of the data source, which is the Java interface (or class) expected to be implemented by the data source.</td>
<td>res-type</td>
</tr>
<tr>
<td>Sign-on responsibility</td>
<td>Specifies whether the EJB code signs on programmatically to the resource manager, or whether the Container will sign on to the resource manager on behalf of the bean. In the latter case, the Container uses information that is supplied by the Deployer.</td>
<td>res-auth</td>
</tr>
<tr>
<td>Description</td>
<td>Text description.</td>
<td>description</td>
</tr>
</tbody>
</table>

**JDBC panel**

The JDBC panel contains the following tabs that describe container-managed persistence for the selected EJB:

- Configuration tab
- Field-Column Map tab
- Finders tab

This panel appears only when persistence-store-jdbc is specified for the EJB, which instructs the container to store the object state in a database that is managed by a JDBC connection pool.
Panels

Note: These XML elements are WLE XML extensions.

Configuration tab

Figure 3-12 shows an example of the Configuration tab.

Figure 3-12 Configuration Tab

The Configuration tab displays the following container-managed persistence configuration information for the selected EJB:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>XML Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table name</td>
<td>Name of the table used to store EJB state information.</td>
<td>table-name</td>
</tr>
<tr>
<td>Connection Pool Name</td>
<td>Name of the JDBC connection pool associated with the database used to store EJB state information.</td>
<td>pool-name</td>
</tr>
<tr>
<td>Description</td>
<td>Text description.</td>
<td>description</td>
</tr>
</tbody>
</table>

You can edit the Table Name and Connection Pool Name fields using the WebLogic EJB Deployer.

Field-Column Map tab

Figure 3-13 shows an example of the Field-Column Map tab.

Figure 3-13 Field-Column Map Tab

The Field-Column Map tab displays a list of the following container-managed persistence mapping information (attribute-map and attribute-map-entry) for the selected EJB:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>XML Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Name of an EJB field mapped to the table column in this attribute map entry.</td>
<td>bean-field-name</td>
</tr>
</tbody>
</table>
You can edit the Table column field using the WebLogic EJB Deployer.

Finders tab

Figure 3-14 shows an example of the Finders tab.

**Figure 3-14  Finders Tab**

The Finders tab displays a list of the following container-managed persistence finders information for the selected EJB:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>XML Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finder signature</td>
<td>Finder descriptor for finding one or more EJBs.</td>
<td>finder-descriptor</td>
</tr>
</tbody>
</table>

You can edit an expression for a Finder signature by selecting it in the list, and then clicking Edit Expression. The Edit Expression dialog box appears. Figure 3-15 shows an example of the Edit Expression dialog box.

**Figure 3-15  Edit Expression Dialog Box**

The Edit Expression dialog box displays an editing window for changing the expression for the selected finder signature. Edit the expression, and then click OK to close the dialog box.

The expression must comply with the syntax rules specified for the query-grammar element. For more information about the expression syntax, see the detailed description in query-grammar.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>XML Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Java data type associated with the field.</td>
<td>N/A</td>
</tr>
<tr>
<td>Table column</td>
<td>Name of a table column mapped to the field in this attribute map entry.</td>
<td>table-column-name</td>
</tr>
</tbody>
</table>
File panel

Figure 3-16 shows an example of the File panel.

Figure 3-16  File Panel

For stateful session beans with container-managed persistence, the File panel displays the following information for the selected File entry (when `persistence-store-file` is specified for the EJB):

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>XML Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persistent directory</td>
<td>If persistent storage is to a flat file, specifies the directory root for storing all instances of an EJB. Each <code>ejb-name</code> element is converted to a directory name.</td>
<td>persistence-store-direc</td>
</tr>
<tr>
<td>root</td>
<td></td>
<td>tory-root</td>
</tr>
<tr>
<td>Description</td>
<td>Text description.</td>
<td>description</td>
</tr>
</tbody>
</table>

You can change the directory root using the WebLogic EJB Deployer, either by typing a different path or by clicking the Select button to browse the file system and select a different path.

**Note:** You must specify a path that already exists.

If you click the Select button, the Specify Directory for Storing Entity Beans dialog box appears. Figure 3-17 shows an example of this dialog box.

Figure 3-17  Specify Directory for Storing Entity Beans Dialog Box

Browse the file system, select the path you want, and then click Select.

To save changes, see “Saving Changes to ejb-jar Files” on page 2-13.
This section describes the XML DTD for the EJB deployment descriptor as specified in Sun Microsystem’s Enterprise JavaBeans Specification 1.1 (Public Release 2, dated October 18, 1999). This topic includes the following sections:

- About Deployment Descriptors
- DOCTYPE Declaration
- Summary of Deployment Descriptor Elements
- Description of Deployment Descriptor Elements

In addition, WebLogic Enterprise (WLE) provides the WebLogic EJB Extensions, which are XML elements used to further customize EJBs. For more information, see Chapter 5, “WebLogic EJB Extensions Reference.” For more information about deploying EJBs in the WLE environment, see Building and Deploying Enterprise JavaBeans (EJBs).
About Deployment Descriptors

A deployment descriptor is a file in the ejb-jar file that defines the following kinds of information:

- **EJB structural information**, such as the EJB name, class, home and remote interfaces, bean type (session or entity), environment entries, resource factory references, EJB references, security role references, as well as additional information based on the bean type.

- **Application assembly information**, such as EJB references, security roles, security role references, method permissions, and container transaction attributes. Specifying assembly descriptor information is an optional task that an Application Assembler performs.

For a complete discussion of the EJB deployment descriptor, see Chapter 16, “Deployment Descriptor,” in Sun Microsystems’s *Enterprise JavaBeans Specification 1.1*.

**EJB Roles and Deployment Descriptors**

Specifying deployment descriptor information is a required task that a Bean Provider performs. The Bean Provider creates a standard EJB deployment descriptor file using the XML markup conventions in accordance with the syntax described in Sun Microsystems’s *Enterprise JavaBeans Specification 1.1*. Multiple EJBs can be defined in a single deployment descriptor. For more information about creating deployment descriptors, see *Building and Deploying Enterprise JavaBeans (EJBs)*.

An Application Assembler or Deployer may subsequently modify only certain information (such as the EJB name, values of environment entries, and description entries) in the deployment descriptor. The Deployer may use the WebLogic EJB Deployer to update certain settings in the deployment descriptor.

Thereafter, the Application Assembler creates a deployable ejb-jar file using the `ejbc` command-line tool, specifying the `-i` parameter and passing this filename (`ejb-jar.xml`). For more information about the `ejbc` command-line tool, see the *Command Reference*. 

4-2 Using the WebLogic EJB Deployer
File Formats

Deployment descriptors are stored in the ejb-jar file. The WebLogic EJB Deployer accepts standard, extended, and deployable ejb-jar files.

DOCTYPE Declaration

All valid ejb-jar deployment descriptors must contain the following DOCTYPE declaration:

```xml
<!DOCTYPE ejb-jar PUBLIC "-//Sun Microsystems, Inc./DTD Enterprise JavaBeans 1.1//EN"
"http://java.sun.com/j2ee/dtds/ejb-jar_1_1.dtd">
```

Summary of Deployment Descriptor Elements

Table 4-1  List of Standard Deployment Descriptor Elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>EJB Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>assembly-descriptor</td>
<td>Contains application-assembly information.</td>
<td>Application Assembler</td>
</tr>
<tr>
<td>cmp-field</td>
<td>Describes a container-managed field.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>container-transaction</td>
<td>Specifies how the container must manage transaction scopes for the EJB’s method invocations.</td>
<td>Application Assembler</td>
</tr>
<tr>
<td>description</td>
<td>Provides text describing the parent element.</td>
<td>Bean Provider or</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Application Assembler</td>
</tr>
<tr>
<td>Element</td>
<td>Description</td>
<td>EJB Role</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>display-name</td>
<td>Specifies a short name that is intended to be displayed by tools.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>ejb-class</td>
<td>Specifies the fully qualified name of the EJB’s class.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>ejb-jar</td>
<td>Root element of the EJB deployment descriptor.</td>
<td>All</td>
</tr>
<tr>
<td>ejb-link</td>
<td>Used in the ejb-ref element to specify that an EJB reference is linked to another EJB in the ejb-jar file.</td>
<td>Application Assembler</td>
</tr>
<tr>
<td>ejb-name</td>
<td>Specifies an EJB’s name.</td>
<td>Bean Provider or Application Assembler</td>
</tr>
<tr>
<td>ejb-ref</td>
<td>Declares a reference to another EJB’s home.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>ejb-ref-name</td>
<td>Specifies the name of an EJB reference, which is an entry in the EJB’s environment.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>ejb-ref-type</td>
<td>Specifies the expected type (entity or session) of the referenced EJB.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>enterprise-beans</td>
<td>Declares one or more EJBs.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>entity</td>
<td>Declares an entity bean.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>env-entry</td>
<td>Declares an EJB’s environment entries.</td>
<td>Bean Provider or Application Assembler</td>
</tr>
<tr>
<td>env-entry-name</td>
<td>Specifies the name of an EJB’s environment entry.</td>
<td>Bean Provider or Application Assembler</td>
</tr>
<tr>
<td>env-entry-type</td>
<td>Specifies the fully qualified Java type of the environment entry value that is expected by the EJB’s code.</td>
<td>Bean Provider or Application Assembler</td>
</tr>
<tr>
<td>env-entry-value</td>
<td>Specifies the value of an EJB’s environment entry.</td>
<td>Bean Provider or Application Assembler</td>
</tr>
<tr>
<td>field-name</td>
<td>Specifies the name of a container-managed field.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>Element</td>
<td>Description</td>
<td>EJB Role</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>home</td>
<td>Specifies the fully qualified name of the EJB’s home interface.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>large-icon</td>
<td>Specifies the name of a file containing a large (32 x 32) icon image. The</td>
<td>Bean Provider</td>
</tr>
<tr>
<td></td>
<td>file name is a relative path within the ejb-jar file.</td>
<td></td>
</tr>
<tr>
<td>method</td>
<td>Denotes a method of an EJB’s home or remote interface, or a set of methods.</td>
<td>Application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Assembler</td>
</tr>
<tr>
<td>method-intf</td>
<td>Allows a method element to differentiate between the methods, with the same</td>
<td>Application</td>
</tr>
<tr>
<td></td>
<td>name and signature, that are defined in both the remote and home interfaces.</td>
<td>Assembler</td>
</tr>
<tr>
<td>method-name</td>
<td>Specifies the name of an EJB method, or the asterisk (*) character, which</td>
<td>Application</td>
</tr>
<tr>
<td></td>
<td>is used when the element denotes all the methods of an EJB’s remote and</td>
<td>Assembler</td>
</tr>
<tr>
<td></td>
<td>home interfaces.</td>
<td></td>
</tr>
<tr>
<td>method-param</td>
<td>Specifies the fully qualified Java type name of a method parameter.</td>
<td>Application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Assembler</td>
</tr>
<tr>
<td>method-params</td>
<td>Contains a list of the fully qualified Java type names of the method</td>
<td>Application</td>
</tr>
<tr>
<td></td>
<td>parameters.</td>
<td>Assembler</td>
</tr>
<tr>
<td>method-permission</td>
<td>Specifies one or more security roles that are allowed to invoke one or</td>
<td>Application</td>
</tr>
<tr>
<td></td>
<td>more EJB methods.</td>
<td>Assembler</td>
</tr>
<tr>
<td>persistence-type</td>
<td>Specifies an entity bean’s persistence management type (bean- or</td>
<td>Bean Provider</td>
</tr>
<tr>
<td></td>
<td>container-managed).</td>
<td></td>
</tr>
<tr>
<td>prim-key-class</td>
<td>Specifies the fully qualified name of an entity bean’s primary key class.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>primkey-field</td>
<td>Specifies the name of the primary key field for an entity with container-</td>
<td>Bean Provider</td>
</tr>
<tr>
<td></td>
<td>managed persistence.</td>
<td></td>
</tr>
<tr>
<td>reentrant</td>
<td>Specifies whether an entity bean is reentrant (True) or not (False).</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>remote</td>
<td>Specifies the fully qualified name of the EJB’s remote interface.</td>
<td>Bean Provider</td>
</tr>
</tbody>
</table>
### Table 4-1  List of Standard Deployment Descriptor Elements  (Continued)

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>EJB Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>res-auth</td>
<td>Specifies whether the EJB code signs on programmatically to the resource manager, or whether the Container will sign on to the resource manager on behalf of the bean. In the latter case, the Container uses information that is supplied by the Deployer.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>res-ref-name</td>
<td>Specifies the name of a resource factory reference.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>res-type</td>
<td>Specifies the type of the data source, which is the Java interface (or class) expected to be implemented by the data source.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>resource-ref</td>
<td>Declares the EJB’s reference to an external resource.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>role-link</td>
<td>Links a security role reference to a defined security role.</td>
<td>Application Assembler</td>
</tr>
<tr>
<td>role-name</td>
<td>Specifies the name of a security role.</td>
<td>Application Assembler</td>
</tr>
<tr>
<td>security-role</td>
<td>Defines a security role.</td>
<td>Application Assembler</td>
</tr>
<tr>
<td>security-role-ref</td>
<td>Declares a security role reference in the EJB’s code.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>session-type</td>
<td>Specifies the type of a session bean (stateful or stateless).</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>session</td>
<td>Declares a session bean.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>small-icon</td>
<td>Specifies the name of a file containing a small (16 x 16) icon image. The file name is relative path within the ejb-jar file.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>transaction-type</td>
<td>Specifies the transaction management type of an EJB (bean- or container-managed).</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>trans-attribute</td>
<td>Specifies how the container must manage the transaction boundaries when delegating a method invocation to an EJB’s business method.</td>
<td>Application Assembler</td>
</tr>
</tbody>
</table>
Description of Deployment Descriptor Elements

**assembly-descriptor**

Synopsis: Contains application-assembly information.

Syntax:  
```
<!ELEMENT assembly-descriptor (security-role*, method-permission*,
container-transaction*)>
```

EJB Role: Application Assembler

Used In: ejb-jar

**cmp-field**

Synopsis: Describes a container-managed field.

Syntax:  
```
<!ELEMENT cmp-field (description?, field-name)>
```

EJB Role: Bean Provider

Used In: entity

**container-transaction**

Synopsis: Specifies how the container must manage transaction scopes for the EJB’s method invocations.

Syntax:  
```
<!ELEMENT container-transaction (description?, method+,
trans-attribute)>
```

EJB Role: Application Assembler

Used In: assembly-descriptor

**description**

Synopsis: Provides text describing the parent element.
Syntax: `<!ELEMENT description (#PCDATA)>`

**EJB Role:** Bean Provider or Application Assembler, and

**Used In:**
- `cmp-field`
- `container-transaction`
- `ejb-jar`
- `entity`
- `env-entry`
- `ejb-ref`
- `method`
- `method-permission`
- `resource-ref`
- `security-role`
- `security-role-ref`
- `session`

**display-name**

**Synopsis:** Specifies a short name that is intended to be displayed by tools.

**Syntax:** `<!ELEMENT display-name (#PCDATA)>`

**EJB Role:** Bean Provider

**Used In:**
- `ejb-jar`
- `session`
- `entity`

**ejb-class**

**Synopsis:** Specifies the fully qualified name of the EJB’s class.

**Syntax:** `<!ELEMENT ejb-class (#PCDATA)>`

**EJB Role:** Bean Provider

**Used In:**
- `entity`
Description of Deployment Descriptor Elements

- **session**

**ejb-jar**

**Synopsis**
Root element of the EJB deployment descriptor.

**Syntax**
```xml
<!ELEMENT ejb-jar (description?, display-name?, small-icon?,
large-icon?, enterprise-beans, assembly-descriptor?)>
```

**EJB Role**
All

**ejb-link**

**Synopsis**
Used in the `ejb-ref` element to specify that an EJB reference is linked to another EJB in the ejb-jar file.

**Syntax**
```xml
<!ELEMENT ejb-link (#PCDATA)>
```

**Valid Values**
The value of the `ejb-link` element must be the `ejb-name` of an EJB in the same ejb-jar file, or in another ejb-jar file in the same J2EE application unit.

**EJB Role**
Application Assembler

**Used In**
ejb-ref

**ejb-name**

**Synopsis**
Specifies an EJB’s name.

**Syntax**
```xml
<!ELEMENT ejb-name (#PCDATA)>
```

**Valid Values**
- The name must be unique among the names of the EJBs in the same ejb-jar file.
- The name must conform to the lexical rules for an NMTOKEN.

**EJB Role**
Bean Provider or Application Assembler

**Used In**
- entity
- method
- session
ejb-ref

Synopsis  Declaras a reference to another EJB’s home.

Syntax  <!ELEMENT ejb-ref (description?, ejb-ref-name, ejb-ref-type, home, remote, ejb-link?)>

EJB Role  Bean Provider

Used In  entity
        session

ejb-ref-name

Synopsis  Specifies the name of an EJB reference, which is an entry in the EJB’s environment.

Syntax  <!ELEMENT ejb-ref-name (#PCDATA)>

Valid Values  It is recommended that name is prefixed with ejb/.

EJB Role  Bean Provider

Used In  ejb-ref

ejb-ref-type

Synopsis  Specifies the expected type (entity or session) of the referenced EJB.

Syntax  <!ELEMENT ejb-ref-type (#PCDATA)>

Valid Values  One of the following values:
        Entity
        Session

EJB Role  Bean Provider

Used In  ejb-ref

enterprise-beans

Synopsis  Declares one or more EJBs.
Description of Deployment Descriptor Elements

Syntax  
<!ELEMENT enterprise-beans (session | entity)+>

EJB Role  
Bean Provider

definition

Synopsis  
Declares an entity bean.

Syntax  
<!ELEMENT entity (description?, display-name?, small-icon?, large-icon?, ejb-name, home, remote, ejb-class, persistence-type, prim-key-class, reentrant, cmp-field*, primkey-field?, env-entry*, ejb-ref*, security-role-ref*, resource-ref*)>

Valid Values
- The optional primkey-field may be present in the descriptor if the entity’s persistence-type is Container.
- At least one cmp-field element must be present in the descriptor if the entity’s persistency-type is Container, and none must not be present if the entity’s persistence-type is Bean.

EJB Role  
Bean Provider

Used In  
entity

env-entry

Synopsis  
Declares an EJB’s environment entries.

Syntax  
<!ELEMENT env-entry (description?, env-entry-name, env-entry-type, env-entry-value?)>

EJB Role  
Bean Provider or Application Assembler

Used In
- entity
- session

env-entry-name

Specifies the name of an EJB’s environment entry.

Syntax  
<!ELEMENT env-entry-name (#PCDATA)>

EJB Role  
Bean Provider or Application Assembler
Used In env-entry

env-entry-type

Synopsis Specifies the fully qualified Java type of the environment entry value that is expected by the EJB’s code.

Syntax `<!ELEMENT env-entry-type (#PCDATA)>`

Valid Values One of the following values:
- `java.lang.Boolean`
- `java.lang.String`
- `java.lang.Integer`
- `java.lang.Double`
- `java.lang.Byte`
- `java.lang.Short`
- `java.lang.Long`
- `java.lang.Float`

EJB Role Bean Provider or Application Assembler

Used In env-entry

Env-entry-value

Synopsis Specifies the value of an EJB’s environment entry.

Syntax `<!ELEMENT env-entry-value (#PCDATA)>`

EJB Role Bean Provider or Application Assembler

Used In env-entry

Field-name

Synopsis Specifies the name of a container-managed field.

Syntax `<!ELEMENT field-name (#PCDATA)>`
Description of Deployment Descriptor Elements

Valid Values  The name must be a public field of the EJB class or one of its superclasses.

EJB Role  Bean Provider

Used In  cmp-field

home

Synopsis  Specifies the fully qualified name of the EJB’s home interface.

Syntax  <!ELEMENT home (#PCDATA)>

EJB Role  Bean Provider

Used In  
  - ejb-ref
  - entity
  - session

large-icon

Synopsis  Specifies the name of a file containing a large (32 x 32) icon image. The file name is a relative path within the ejb-jar file.

Syntax  <!ELEMENT large-icon (#PCDATA)>

Valid Values  
  - The image must be either in the JPEG or GIF format.
  - The file name must end with the suffix .jpg or .gif respectively.

EJB Role  Bean Provider

method

Synopsis  Denotes a method of an EJB’s home or remote interface, or a set of methods.

Syntax  <!ELEMENT method (description?, ejb-name, method-intf?, method-name, method-params?)>

Valid Values  The ejb-name element must be the name of one of the EJBs declared in the deployment descriptor.

EJB Role  Application Assembler
method-intf

Synopsis
Allows a method element to differentiate between the methods, with the same name and signature, that are defined in both the remote and home interfaces.

Syntax
<!ELEMENT method-intf (#PCDATA)>

Valid Values
One of the following values:
- Home
- Remote

EJB Role
Application Assembler

Used In
method

method-name

Synopsis
Specifies the name of an EJB method, or the asterisk (*) character, which is used when the element denotes all the methods of an EJB’s remote and home interfaces.

Syntax
<!ELEMENT method-name (#PCDATA)>

EJB Role
Application Assembler

Used In
method

method-param

Synopsis
Specifies the fully qualified Java type name of a method parameter.

Syntax
<!ELEMENT method-param (#PCDATA)>

EJB Role
Application Assembler

Used In
method-params
**method-params**

**Synopsis**
Contains a list of the fully qualified Java type names of the method parameters.

**Syntax**
```xml
<!ELEMENT method-params (method-param*)>
```

**EJB Role**
Application Assembler

**Used In**
method

**method-permission**

**Synopsis**
Specifies one or more security roles that are allowed to invoke one or more EJB methods.

**Syntax**
```xml
<!ELEMENT method-permission (description?, role-name+, method+)>```

**Valid Values**
- The security roles used in the method-permission element must be defined in the security-role element of the deployment descriptor.
- The methods must be methods defined in the EJB’s remote and/or home interfaces.

**EJB Role**
Application Assembler

**Used In**
assembly-descriptor

**persistence-type**

**Synopsis**
Specifies an entity bean’s persistence management type (bean- or container-managed).

**Syntax**
```xml
<!ELEMENT persistence-type (#PCDATA)>```

**Valid Values**
One of the following values:
- Bean
- Container

**EJB Role**
Bean Provider

**Used In**
entity
4 EJB Deployment Descriptor Reference

**prim-key-class**

**Synopsis**
Specifies the fully qualified name of an entity bean’s primary key class.

**Syntax**
<!ELEMENT prim-key-class (#PCDATA)>

**Valid Values**
If the definition of the primary key class is deferred to deployment time, the prim-key-class element should specify `java.lang.Object`.

**EJB Role**
Bean Provider

**Used In**
entity

**primkey-field**

**Synopsis**
Specifies the name of the primary key field for an entity with container-managed persistence.

**Syntax**
<!ELEMENT primkey-field (#PCDATA)>

**Valid Values**
- The `primkey-field` must be one of the fields declared in the `cmp-field` element.
- The type of the field must be the same as the primary key type.

**EJB Role**
Bean Provider

**Used In**
entity

**reentrant**

**Synopsis**
Specifies whether an entity bean is reentrant (True) or not (False).

**Syntax**
<!ELEMENT reentrant (#PCDATA)>

**Valid Values**
One of the following values:
- True
- False

**EJB Role**
Bean Provider

**Used In**
entity
Description of Deployment Descriptor Elements

**remote**
- **Synopsis**: Specifies the fully qualified name of the EJB’s remote interface.
- **Syntax**: `<!ELEMENT remote (#PCDATA)>`
- **EJB Role**: Bean Provider
- **Used In**:
  - `ejb-ref`
  - `entity`
  - `session`

**res-auth**
- **Synopsis**: Specifies whether the EJB code signs on programmatically to the resource manager, or whether the Container will sign on to the resource manager on behalf of the bean. In the latter case, the Container uses information that is supplied by the Deployer.
- **Syntax**: `<!ELEMENT res-auth (#PCDATA)>`
- **Valid Values**: One of the following values:
  - Application
  - Container
- **EJB Role**: Bean Provider

**res-ref-name**
- **Synopsis**: Specifies the name of a resource manager connection factory reference.
- **Syntax**: `<!ELEMENT res-ref-name (#PCDATA)>`
- **EJB Role**: Bean Provider
- **Used In**: `resource-ref`

**res-type**
- **Synopsis**: Specifies the type of the data source, which is the Java interface (or class) expected to be implemented by the data source.
4 EJ B Deployment Descriptor Reference

Syntax  <!ELEMENT res-type (#PCDATA)>

EJB Role  Bean Provider

Used In  resource-ref

resource-ref

Synopsis  Declares the EJB’s reference to an external resource.

Syntax  <!ELEMENT resource-ref (description?, res-ref-name, res-type, res-auth)>

EJB Role  Bean Provider

Used In  entity
  ■ session

role-link

Synopsis  Links a security role reference to a defined security role.

Syntax  <!ELEMENT role-link (#PCDATA)>

Valid Values  The role-link element must contain the name of one of the security roles defined in the security-role elements.

EJB Role  Application Assembler

Used In  security-role-ref

role-name

Synopsis  Specifies the name of a security role.

Syntax  <!ELEMENT role-name (#PCDATA)>

Valid Values  The name must conform to the lexical rules for an NMTOKEN.

EJB Role  Application Assembler

Used In  ■ method-permission
  ■ security-role
Description of Deployment Descriptor Elements

- `security-role-ref`

**security-role**

Synopsis  Defines a security role.
Syntax     `<!ELEMENT security-role (description?, role-name)>`
EJB Role   Application Assembler
Used In   assembly-descriptor

**security-role-ref**

Synopsis  Declares a security role reference in the EJB’s code.
Syntax    `<!ELEMENT security-role-ref (description?, role-name, role-link?)>
Valid Values
- The value of the `role-name` element must be the String used as the parameter to the `EJBContext.isCallerInRole(String roleName)` method.
- The value of the `role-link` element must be the name of one of the security roles defined in the `security-role` elements.

EJB Role   Bean Provider
Used In   - entity
           - session

**session-type**

Synopsis  Specifies the type of a session bean (stateful or stateless).
Syntax    `<!ELEMENT session-type (#PCDATA)>`
Valid Values One of the following values:
- Stateful
- Stateless

EJB Role   Bean Provider
4 EJB Deployment Descriptor Reference

session

Synopsis
Declares a session bean.

Syntax
<!ELEMENT session (description?, display-name?, small-icon?,
large-icon?, ejb-name, home, remote, ejb-class, session-type,
transaction-type, env-entry*, ejb-ref*, security-role-ref*,
resource-ref*)>

EJB Role
Bean Provider

Used In
enterprise-beans

small-icon

Synopsis
Specifies the name of a file containing a small (16 x 16) icon image. The file name is relative path within the ejb-jar file.

Syntax
<!ELEMENT small-icon (#PCDATA)>

Valid Values
- The image must be either in the JPEG or GIF format.
- The file name must end with the suffix .jpg or .gif respectively.

EJB Role
Bean Provider

transaction-type

Synopsis
Specifies the transaction management type of an EJB (bean- or container-managed).

Syntax
<!ELEMENT transaction-type (#PCDATA)>

Valid Values
One of the following values:
- Bean
- Container
Description of Deployment Descriptor Elements

trans-attribute

Synopsis
Specifies how the container must manage the transaction boundaries when delegating
a method invocation to an EJB’s business method.

Syntax
<!ELEMENT trans-attribute (#PCDATA)>

Valid Values
One of the following values:

- NotSupported
- Supports
- Required
- RequiresNew
- Mandatory
- Never

EJB Role
Application Assembler

Used In
container-transaction

ID

Synopsis
The ID mechanism allows tools that produce additional deployment information (such
as information beyond the standard EJB deployment descriptor information) to store
the non-standard information in a separate file, and easily refer from these
tools-specific files to the information in the standard deployment descriptor.

The EJB architecture does not allow the tools to add the nonstandard information into
the EJB deployment descriptor.

Syntax
<!ATTLIST assembly-descriptor id ID #IMPLIED>
<!ATTLIST cmp-field id ID #IMPLIED>
<!ATTLIST container-transaction id ID #IMPLIED>
<!ATTLIST description id ID #IMPLIED>
<!ATTLIST display-name id ID #IMPLIED>
<!ATTLIST ejb-class id ID #IMPLIED>
<!ATTLIST ejb-jar id ID #IMPLIED>
<!ATTLIST ejb-link id ID #IMPLIED>
<!ATTLIST ejb-name id ID #IMPLIED>
<!ATTLIST ejb-ref id ID #IMPLIED>
<!ATTLIST ejb-ref-name id ID #IMPLIED>
<!ATTLIST ejb-ref-type id ID #IMPLIED>
<!ATTLIST enterprise-beans id ID #IMPLIED>
<!ATTLIST entity id ID #IMPLIED>
<!ATTLIST env-entry id ID #IMPLIED>
<!ATTLIST env-entry-name id ID #IMPLIED>
<!ATTLIST env-entry-type id ID #IMPLIED>
<!ATTLIST env-entry-value id ID #IMPLIED>
<!ATTLIST field-name id ID #IMPLIED>
<!ATTLIST home id ID #IMPLIED>
<!ATTLIST large-icon id ID #IMPLIED>
<!ATTLIST method id ID #IMPLIED>
<!ATTLIST method-intf id ID #IMPLIED>
<!ATTLIST method-name id ID #IMPLIED>
<!ATTLIST method-param id ID #IMPLIED>
<!ATTLIST method-params id ID #IMPLIED>
<!ATTLIST method-permission id ID #IMPLIED>
<!ATTLIST persistence-type id ID #IMPLIED>
<!ATTLIST prim-key-class id ID #IMPLIED>
<!ATTLIST primkey-field id ID #IMPLIED>
<!ATTLIST reentrant id ID #IMPLIED>
<!ATTLIST remote id ID #IMPLIED>
<!ATTLIST res-auth id ID #IMPLIED>
<!ATTLIST res-ref-name id ID #IMPLIED>
<!ATTLIST res-type id ID #IMPLIED>
<!ATTLIST resource-ref id ID #IMPLIED>
<!ATTLIST role-link id ID #IMPLIED>
<!ATTLIST role-name id ID #IMPLIED>
<!ATTLIST security-role id ID #IMPLIED>
<!ATTLIST security-role-ref id ID #IMPLIED>
<!ATTLIST session-type id ID #IMPLIED>
<!ATTLIST session id ID #IMPLIED>
<!ATTLIST small-icon id ID #IMPLIED>
<!ATTLIST transaction-type id ID #IMPLIED>
<!ATTLIST trans-attribute id ID #IMPLIED>
This topic describes the WebLogic EJB Extensions, which are XML elements used to configure EJBs used in applications that run under WebLogic Enterprise (WLE). This topic includes the following sections:

- About the WebLogic EJB Extensions
- Summary of WebLogic EJB Extensions
- Description of WebLogic EJB Extension Elements

For more information about deploying EJBs in the WLE environment, see *Building andDeploying Enterprise JavaBeans (EJBs)*.
About the WebLogic EJB Extensions

Sun’s Enterprise JavaBeans Specification 1.1 (Public Release 2 dated October 18, 1999) specification defines a standard DTD for describing EJB deployment information. For a list of the standard XML elements, see Chapter 4, “EJB Deployment Descriptor Reference.”

WLE extends the ability to configure EJBs by providing additional, optional XML elements that can be associated with EJBs. The Enterprise JavaBeans Specification 1.1 mandates that these extensions must be specified in a separate file. The WebLogic EJB Extensions reside in a file name \texttt{weblogic-ejb-extensions.xml}, which is included in an ejb-jar file.

The Bean Developer is responsible for creating this file, if needed, and the Application Assembler is responsible for including it in a deployable ejb-jar file. To add this extensions file to an ejb-jar file, an Application Assembler uses the \texttt{ejbc} command line tool, specifying the \texttt{-x} parameter and passing this file name \texttt{(weblogic-ejb-extensions.xml)}. For more information about the \texttt{ejbc} command line tool, see the Command Reference.
### Summary of WebLogic EJB Extensions

#### Table 5-1 WebLogic Enterprise XML Extensions to the EJB Deployment Descriptor

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>EJB Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>attribute-map</td>
<td>Specifies an attribute map containing one or more attribute map entries, which define a mapping between an EJB field (a container-managed attribute) and a table column name.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>attribute-map-entry</td>
<td>Specifies the mapping between an EJB field and a table column.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>bean-field-name</td>
<td>Specifies the name of an EJB field in an attribute map entry.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>finder-descriptor</td>
<td>Specifies the finder descriptor for finding one or more EJBs.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>jndi-name</td>
<td>Specifies the name for the EJB home that is registered in the Global WLE JNDI namespace.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>module-initializer-class</td>
<td>Specifies the server implementation class for handling the initialize and release methods.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>module-initializer-class-name</td>
<td>Specifies the name of the server implementation class for handling the initialize and release methods.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>persistence-store-descriptor</td>
<td>Specifies the type of persistence storage (in a flat file or a JDBC-managed database) for container-managed persistence.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>persistence-store-directory-root</td>
<td>If persistent storage is kept in a flat file, specifies the directory root for storing all instances of an EJB. Each ejb-name element is converted to a directory name.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>persistence-store-file</td>
<td>Specifies persistent storage in a flat file.</td>
<td>Bean Provider</td>
</tr>
</tbody>
</table>
### Table 5-1  WebLogic Enterprise XML Extensions to the EJB Deployment Descriptor  (Continued)

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>EJB Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>persistence-store-jdbc</td>
<td>Specifies persistent storage in a database that is managed by a WLE JDBC connection pool.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>pool-name</td>
<td>Specifies the name of the JDBC connection pool associated with the database used to store EJB state information.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>query-grammar</td>
<td>Specifies the query expression for a finder method.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>table-column-name</td>
<td>Specifies the name of a table column in an attribute map entry.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>table-name</td>
<td>Specifies the name of the table used to store EJB state information.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>trans-timeout-seconds</td>
<td>Specifies the default timeout, in seconds, for container-initiated transactions for this bean. If a transaction duration exceeds this value, the transaction will be rolled back.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>weblogic-deployment-params</td>
<td>Specifies deployment options for the WebLogic Server.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>weblogic-ejb-extensions</td>
<td>Root element of the WLE XML extensions to the deployment descriptor.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>weblogic-enterprise-bean</td>
<td>Specifies deployment information for a bean that is specific to WLE.</td>
<td>Bean Provider</td>
</tr>
<tr>
<td>weblogic-version</td>
<td>Specifies the WLE version.</td>
<td>Bean Provider</td>
</tr>
</tbody>
</table>
Description of WebLogic EJB Extension Elements

attribute-map

Synopsis Specifies an attribute map containing one or more attribute map entries, which define a mapping between an EJB field (a container-managed attribute) and a table column name.

Syntax <!ELEMENT attribute-map (description?, attribute-map-entry+)> 

EJB Role Bean Provider

Used In persistence-store-jdbc

attribute-map-entry

Synopsis Specifies the mapping between an EJB field and a table column.

Syntax <!ELEMENT attribute-map-entry (bean-field-name, table-column-name)> 

Description No automatic type conversion between types is performed on primary key columns, and therefore the specified mapping must be between equivalent types.

EJB Role Bean Provider

Used In attribute-map

bean-field-name

Synopsis Specifies the name of an EJB field in an attribute map entry.

Syntax <!ELEMENT bean-field-name (#PCDATA)> 

EJB Role Bean Provider

Used In attribute-map-entry
finder-descriptor

Synopsis  Specifies the finder descriptor for finding one or more EJBs.

Syntax  
<!ELEMENT finder-descriptor (description?, method?,
query-grammar?)>

Valid Values
- Method names must always start with the prefix find.
- If you use a nonprimitive data type in a finder descriptor, you must specify a
  fully qualified name, such as java.sql.Timestamp.
- Finder descriptors always use EJB field names, not column names, except for
  the ordering of the returned beans (the OrderBy expression).

Description  Finder descriptors are WLE's implementation of the EJB find methods. If an EJB uses
container-managed JDBC persistence, you can specify finder methods of the form
findMethod() to find either an individual or collection of EJBs.

The finder-descriptor elements are pairs of method signatures and expressions.
You specify a method signature in the EJBHome interface and specify the method's
expression in the deployment descriptor using the query-grammar element.
Expressions contain method parameters and EJB attributes. The EJB Container
automatically maps the attributes to the appropriate columns in the persistent store.

For complex SQL lookups, such as a dynamically set WHERE clause, you need to use
bean-managed persistence and write custom finders.

EJB Role  Bean Provider

Used In  persistence-store-jdbc

Example  The findBigAccounts method has the following interface:

```java
public interface AccountHome extends EJBHome {
// ...
    public Enumeration findBigAccounts(double balanceGreaterThan)
        throws FinderException, RemoteException;
// ...
}
```

The following sample expression associated with the findBigAccounts method is
defined in the deployment descriptor:

```xml
(> balance $balanceGreaterThan)
```
where balance is an attribute (field) of the EJB and $balanceGreaterThan is the symbol for the method parameter balanceGreaterThan.

A call in the client application, such as myEJBean.findBigAccounts(amount), will return a list of all EJBs whose balance attribute is greater than the value of amount.

**jndi-name**

<table>
<thead>
<tr>
<th>Description</th>
<th>Specifies the name for the EJB home that is registered in the Global WLE JNDI namespace.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Synopsis</strong></td>
<td><img src="ELEMENT%20jndi-name%20(#PCDATA)%3E" alt="ELEMENT jndi-name (#PCDATA)&gt;" /></td>
</tr>
<tr>
<td><strong>Syntax</strong></td>
<td><img src="ELEMENT%20jndi-name%20(#PCDATA)%3E" alt="ELEMENT jndi-name (#PCDATA)&gt;" /></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>To allow Java clients to perform a lookup on the JNDI name for the EJB home and gain access to the object, the name of the EJB home must be registered in the Global WLE JNDI namespace. This name can differ from the ejb-name element specified in the standard EJB XML. Whereas the ejb-name element must be unique only among the names of the EJBs in the same ejb-jar file, however, the JNDI name must be unique among all global factory names in a WLE domain, including EJB homes, CORBA factories, and RMI-named objects.</td>
</tr>
</tbody>
</table>

**EJB Role** | Bean Provider |
**Used In**  | weblogic-deployment-params |
**Example**  | `<weblogic-ejb-extensions>`<weblogic-enterprise-bean>
  `<ejb-name>Trader</ejb-name>`
  `<weblogic-deployment-params>`
    `<jndi-name>EJB_TraderHome</jndi-name>`
  `</weblogic-deployment-params>`
`</weblogic-enterprise-bean>`
`</weblogic-ejb-extensions>`
module-initializer-class

Synopsis Specifies the server implementation class for handling the initialize and release methods.

Syntax
```xml
<!ELEMENT module-initializer-class
(module-initializer-class-name)>
```

Description The syntax for specifying a Server implementation class for handling the initialize and release methods is specified in the EJB XML deployment descriptor extensions. These XML elements are processed by the ejbc command. The WLE EJB framework parses the XML at run time and performs the startup/shutdown processing.

For a complete description of startup and shutdown handling in WLE, see Starting and Shutting Down Applications.

EJB Role Bean Provider

Used In weblogic-ejb-extensions

module-initializer-class-name

Synopsis Specifies the fully qualified name of the server implementation class for handling the initialize and release methods.

Syntax
```xml
<!ELEMENT module-initializer-class-name (#PCDATA)>
```

EJB Role Bean Provider

Used In module-initializer-class

Example
```xml
<weblogic-ejb-extensions>
  ...
  
  <module-initializer-class>
    <module-initializer-class-name>ServerImpl</module-initializer-class-name>
  </module-initializer-class>
</weblogic-ejb-extensions>
```
### persistence-store-descriptor

**Synopsis**
Specifies the type of persistence storage (in a flat file or a JDBC-managed database) for container-managed persistence.

**Syntax**
```xml
<!ELEMENT persistence-store-descriptor (description?,
(persistence-store-file | persistence-store-jdbc)?)>
```

**Description**
For container-managed persistence, WLE EJB Container supports storing an EJB in either of the following formats:

- a flat file (the `persistence-store-file` element)
- a database that is managed with a WLE JDBC connection pool (the `persistence-store-jdbc` element)

**Note:** Database persistence is *required* for an EJB to fully cooperate in a WLE global transaction. File persistence is intended primarily for development and prototyping.

The standard EJB XML descriptor provides XML elements for specifying which fields in the entity bean are to be persisted (the `cmp-field` element), and information about the primary key. The WebLogic XML extensions provide additional information for mapping to the persistent store.

The code for implementing the persistence is generated by the EJB compiler, based on the bean’s deployment descriptors.

**EJB Role**
Bean Provider

**Used In**
`weblogic-deployment-params`

**Example**
Using JDBC persistence, the following sample XML code shows how to define the automatic saving of two fields (`accountId`, `balance`) in an entity bean to a database table (`ejbAccounts`) using a connection pool (`ejbPool`):

```xml
<weblogic-ejb-extensions>
  <weblogic-enterprise-bean>
    <ejb-name>Account</ejb-name>
    <weblogic-deployment-params>
      <jndi-name>EJB_AccountHome</jndi-name>
      <persistence-store-descriptor>
      ...
      </persistence-store-descriptor>
    </weblogic-deployment-params>
  </weblogic-enterprise-bean>
</weblogic-ejb-extensions>
```
<persistence-store-jdbc>
  <pool-name>ejbPool</pool-name>
  <table-name>ejbAccounts</table-name>
  <attribute-map>
    <attribute-map-entry>
      <bean-field-name>accountId</bean-field-name>
      <table-column-name>id</table-column-name>
    </attribute-map-entry>
    <attribute-map-entry>
      <bean-field-name>balance</bean-field-name>
      <table-column-name>bal</table-column-name>
    </attribute-map-entry>
  </attribute-map>
</persistence-store-jdbc>
</persistence-store-descriptor>
</weblogic-deployment-params>
</weblogic-enterprise-bean>
</weblogic-ejb-extensions>

**persistence-store-directory-root**

**Synopsis**  
If persistent storage is kept in a flat file, specifies the directory root for storing all instances of an EJB. Each ejb-name element is converted to a directory name.

**Syntax**  
<!ELEMENT persistence-store-directory-root (#PCDATA)>

**Description**  
For EJBs with container-managed persistence, stateful session beans are stored using file persistence when they are passivated. The persistence-store-directory-root element defines the storage location for these instances. By default, the root directory is pstore, which WLE creates in the directory where the JavaServer is run.
The default file name for a passivated instance is the fully qualified name of the bean, with underscores replacing the periods, plus additional information to ensure uniqueness. For example, if `persistence-store-directory-root` is `c:\weblogic\mystore`, all instances of the EJB `examples.ejb.AccountBean` might be stored in:

```
c:\weblogic\mystore\examples_ejb_AccountBean\SRVGRP_GROUPX_SRVID_Y_Z_examples.ejb.AccountBean_someBigNumber.db
```

**persistence-store-file**

**Synopsis**
Specifies persistent storage in a flat file.

**Syntax**
```
<!ELEMENT persistence-store-file (description?,
persistence-store-directory-root)>
```

**Description**
File persistence for EJBs is intended primarily for development and prototyping. For an EJB to fully cooperate in a WLE global transaction, use `persistence-store-jdbc` instead.

**EJB Role**
Bean Provider

**Used In**
persistence-store-file

**persistence-store-jdbc**

**Synopsis**
Specifies persistent storage in a database that is managed by a WLE JDBC connection pool.

**Syntax**
```
<!ELEMENT persistence-store-jdbc (description?, pool-name,
table-name, attribute-map, finder-descriptor*)>
```

**Description**
EJB instances are stored in a database that has been previously set up with the JDBC connection pool specified in the `pool-name` element. The `table-name` and `attribute-map` elements map the EJB fields to the appropriate table columns in the database.

**EJB Role**
Bean Provider

**Used In**
persistence-store-descriptor
pool-name

**Synopsis**
Specifies the name of the JDBC connection pool associated with the database used to store EJB state information.

**Syntax**
```xml
<!ELEMENT pool-name (#PCDATA)>
```

**EJB Role**
Bean Provider

**Used In**
persistence-store-jdbc

query-grammar

**Synopsis**
Specifies the query expression for a finder method.

**Syntax**
```xml
<!ELEMENT query-grammar (#PCDATA)>
```

**Description**
The query grammar syntax has an expression prefix notation of:
```plaintext
[operator operand1 operand2]
```
Valid operators include:
- `( ) = < > <= >= / ! & |`
- `like`
- `isNull`
- `isNotNull`
- `orderBy`
- `!` is the Boolean *not*
- `&` is the Boolean *and*
- `|` is the Boolean *or*

Valid operands include:
- another expression
- an EJB field
- a finder method parameter (when preceded by `$`)

**EJB Role**
Bean Provider
Examples

The following examples show sample query expressions in finder methods.

The following expression determines whether balance exceeds amount.

"(> balance $amount)"

The following example shows the use of compound expressions. Note that single quotes are used to denote strings.

"(& (> bal $amount) (! (= accountType 'checking')))

The following example is a method signature and an expression that is converted to a findBigAccounts method in the generated persistent storage class. The method returns an enumeration of beans of this class that satisfy this expression.

"findBigAccounts(double amount)" "(> balance $amount)"

The following example finds all the EJBs in a table.

"findAll()" "(= 1 1)"

The following example finds all EJBs whose lastName fields starts with M.

"findNamesStartingWithM()" "(like lastName M%)"

The following example finds all EJBs whose firstName field is null.

"findNullFirstNames()" "(isNull firstName)"

The following example finds all EJBs whose firstName field is not null.

"findNullFirstNames()" "(isNotNull firstName)"

The following expression finds all EJBs in which the value of the balance field exceeds balanceGreaterThan and returns the beans ordered by the id database column. Notice that this expression should all contained in a single line and that no column mapping is performed. The database column must be specified directly.

"findOrderedAccounts(double balanceGreaterThan)"
    "(orderBy 'id' (orderBy 'id desc (> balance $balanceGreaterThan)))"

The following expression is identical to the previous one except that it returns the beans in descending order:

"findOrderedAccounts(double balanceGreaterThan)"
    "(orderBy 'id desc (> balance $balanceGreaterThan))"

Used In finder-descriptor
table-column-name

Synopsis  Specifies the name of a table column in an attribute map entry.
Syntax  <!ELEMENT table-column-name (#PCDATA)>
EJB Role  Bean Provider
Used In  attribute-map-entry

table-name

Synopsis  Specifies the name of the table used to store EJB state information.
Syntax  <!ELEMENT table-name (#PCDATA)>
EJB Role  Bean Provider
Used In  persistence-store-jdbc

trans-timeout-seconds

Synopsis  Sets the default timeout, in seconds, for container-initiated transactions for this bean. If a transaction duration exceeds this value, the transaction will be rolled back.
Syntax  <!ELEMENT trans-timeout-seconds (#PCDATA)>
EJB Role  Bean Provider
Used In  weblogic-deployment-params

weblogic-deployment-params

Synopsis  Specifies deployment options for the WebLogic Server.
Syntax  <!ELEMENT weblogic-deployment-params (trans-timeout-seconds?, jndi-name?, persistence-store-descriptor?)>
EJB Role  Bean Provider
Used In  weblogic-enterprise-bean
Description of WebLogic EJB Extension Elements

weblogic-ejb-extensions

Synopsis Root element of the WLE XML extensions to the deployment descriptor.
Syntax `<!ELEMENT weblogic-ejb-extensions (description?,
weblogic-version?, weblogic-enterprise-bean*,
module-initializer-class*)>`
EJB Role Bean Provider

weblogic-enterprise-bean

Synopsis Specifies deployment information for a bean that is specific to WLE.
Syntax `<!ELEMENT weblogic-enterprise-bean (ejb-name,
weblogic-deployment-params?)>`
EJB Role Bean Provider
Used In weblogic-ejb-extensions

weblogic-version

Synopsis Specifies the WLE version.
Syntax `<!ELEMENT weblogic-version (#PCDATA)>`
EJB Role Bean Provider
Used In weblogic-ejb-extensions
Index

A
assembly-descriptor element 4-7
attribute-map element 5-5
attribute-map-entry element 5-5

B
bean-field-name element 5-5
Beans list 3-4

closing ejb-jar files 2-16
cmp-field element 4-7
Configuration tab 3-10
configuring the WebLogic EJB Deployer 2-6
customer support contact information iii

closing ejb-jar files 2-16
cmp-field element 4-7
Configuration tab 3-10
configuring the WebLogic EJB Deployer 2-6
customer support contact information iii

C
changing deployment descriptor properties 2-12
closing ejb-jar files 2-16
cmp-field element 4-7
Configuration tab 3-10
configuring the WebLogic EJB Deployer 2-6
customer support contact information iii

D
deploying ejb-jar files 2-15
deployment descriptor
  about deployment descriptors 4-2
  assembly-descriptor element 4-7
  attribute-map element 5-5
  attribute-map-entry element 5-5
  bean-field-name element 5-5
  changing properties 2-12
  cmp-field element 4-7
customer support contact information iii
exit the WebLogic EJB Deployer 2-7
extensions 5-2
about WLE XML extensions 5-2
attribute-map element 5-5
attribute-map-entry element 5-5
bean-field-name element 5-5
finder-descriptor element 5-6
jndi-name element 5-7
module-initializer-class element 5-8
module-initializer-class-name element 5-8
persistence-store element 5-9
persistence-store-directory-root element 5-10
persistence-store-file element 5-11
persistence-store-jdbc element 5-11
pool-name element 5-12
query-grammar element 5-12
table-column-name element 5-14
table-name element 5-14
trans-timeout-seconds element 5-14
weblogic-deployment-params element 5-14
weblogic-ejb-extensions element 5-15
weblogic-enterprise-bean element 5-15
weblogic-version element 5-15
finder-descriptor element 5-6
Finders tab 3-11
G
generating container classes 2-15
H
Help menu 2-4
About command 2-4
Using command 2-4
home element 4-13
I
ID mechanism 4-21
J
JDBC panel 3-9
Configuration tab 3-10
Edit Expression dialog 3-11
Field-Column Map tab 3-10
Finders tab 3-11
jndi-name element 5-7
L
large-icon element 4-13
M
menus 2-3
Edit menu 2-4
File menu 2-3
Help menu 2-4
method element 4-13
Method panel 3-5
method-intf element 4-14
method-name element 4-14
method-param element 4-14
Using the WebLogic EJB Deployer 3
method-params element 4-15
method-permission element 4-15
Module Initialization tab 3-3
module-initializer-class element 5-8
module-initializer-class-name element 5-8

N

navigating
deployment descriptor 2-11
ejb-jar Contents List 2-11
menus 2-3

O

opening ejb-jar files 2-10

P

panels
Beans list 3-4
EJB panel 3-5
EJB Reference panel 3-8
ejb-jar panel 3-2
Environment Entries panel 3-7
File panel 3-12
JDBC panel 3-9
Method panel 3-5
Persistence panel 3-6
Resource Factory Reference panel 3-9
Security Role References panel 3-7
summary of 3-1
Persistence panel 3-6
persistence-store element 5-9
persistence-store-directory-root element 5-10
persistence-store-file element 5-11
persistence-store-jdbc element 5-11
persistence-type element 4-15
pool-name element 5-12
prim-key-class element 4-16
primkey-field element 4-16

printing product documentation ii

Q

query-grammar element 5-12

R

reentrant element 4-16
related information ii
remote element 4-17
res-auth element 4-17
Resource Factory Reference panel 3-9
resource-ref element 4-18
res-ref-name element 4-17
res-type element 4-17
role-link element 4-18
role-name element 4-18

S

saving
changes to ejb-jar files 2-13
to different ejb-jar file names 2-14
Security Role References panel 3-7
Security tab 3-3
security-role element 4-19
security-role-ref element 4-19
session element 4-20
session-type element 4-19
small-icon element 4-20
starting the WebLogic EJB Deployer 2-2
support
technical iii

T

table-column-name element 5-14
table-name element 5-14
tabs
Configuration tab 3-10
Description tab 3-2
Field-Column Map tab 3-10
Finders tab 3-11
Module Initialization tab 3-3
Security tab 3-3
tasks, summary of 2-9
toolbar 2-4
transaction-type element 4-20
trans-attribute element 4-21
trans-timeout-seconds element 5-14

W
WebLogic EJB Deployer
      about the WebLogic EJB Deployer 1-1
      configuring 2-6
      exiting 2-7
      starting 2-2
weblogic-deployment-params element 5-14
weblogic-ejb-extensions element 5-15
weblogic-enterprise-bean element 5-15
weblogic-version element 5-15

X
XML extensions 5-2