



BEA WebLogic Enterprise

Using the WebLogic EJB Deployer

WebLogic Enterprise 5.0
Document Edition 5.0
December 1999

Copyright

Copyright © 1999 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems, Inc. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems, Inc. DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks

BEA, ObjectBroker, TOP END, and Tuxedo are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Connect, BEA Manager, BEA MessageQ, BEA Jolt, M3, eSolutions, eLink, WebLogic, and WebLogic Enterprise are trademarks of BEA Systems, Inc.

All other company names may be trademarks of the respective companies with which they are associated.

Using the WebLogic EJB Deployer

Document Edition	Date	Software Version
5.0	December 1999	BEA WebLogic Enterprise 5.0

Contents

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

About Deployment Descriptors	4-2
EJB Roles and Deployment Descriptors	4-2
File Formats	4-3
DOCTYPE Declaration	4-3
Summary of Deployment Descriptor Elements	4-3
Description of Deployment Descriptor Elements.....	4-7
assembly-descriptor	4-7
cmp-field	4-7
container-transaction	4-7
description	4-7
display-name	4-8
ejb-class	4-8
ejb-jar	4-9
ejb-link	4-9
ejb-name	4-9
ejb-ref	4-10
ejb-ref-name	4-10
ejb-ref-type	4-10
enterprise-beans	4-10
entity	4-11
env-entry	4-11
env-entry-name	4-11
env-entry-type	4-12
env-entry-value	4-12
field-name	4-12
home	4-13
large-icon	4-13
method	4-13
method-intf	4-14
method-name	4-14
method-param	4-14
method-params	4-15
method-permission	4-15
persistence-type	4-15

prim-key-class	4-16
primkey-field	4-16
reentrant	4-16
remote	4-17
res-auth	4-17
res-ref-name	4-17
res-type	4-17
resource-ref	4-18
role-link	4-18
role-name	4-18
security-role	4-19
security-role-ref	4-19
session-type	4-19
session	4-20
small-icon	4-20
transaction-type	4-20
trans-attribute	4-21
ID	4-21

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	5-9
persistence-store-directory-root	5-10
persistence-store-file	5-11
persistence-store-jdbc	5-11
pool-name	5-12

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

Index



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.
- Chapter 5, “WebLogic EJB Extensions Reference,” provides a reference to BEA Systems’ WLE XML extensions to 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.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.

Convention	Item
<i>italics</i>	Indicates emphasis or book titles.
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. <i>Examples:</i> #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. <i>Example:</i> void commit ()
<i>monospace italic text</i>	Identifies variables in code. <i>Example:</i> String <i>expr</i>
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. <i>Examples:</i> LPT1 SIGNON OR
{ }	Indicates a set of choices in a syntax line. The braces themselves should never be typed.
[]	Indicates optional items in a syntax line. The brackets themselves should never be typed. <i>Example:</i> buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...

Convention	Item
	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.
...	Indicates one of the following in a command line: <ul style="list-style-type: none">■ That an argument can be repeated several times in a command line■ That the statement omits additional optional arguments■ That you can enter additional parameters, values, or other information The ellipsis itself should never be typed. <i>Example:</i> <code>buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...</code>
.	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.



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.

- 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

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

The WebLogic EJB Deployer imports all three formats and exports extended and deployable file formats.

2 Using the WebLogic EJB Deployer

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

```
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:

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

where *options* are:

Option	Description
-help	Prints help text.
-laf [basic metal motif win]	Sets the look and feel of the editor. Defaults to system-native.

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:

Command	Description
Open	Opening ejb-jar Files
Save	Saving Changes to ejb-jar Files

Command	Description
Save As	Saving the Current ejb-jar File to a New Name
Save All	Saving Changes to ejb-jar Files
Close	Closing ejb-jar Files
Close All	Closing ejb-jar Files
Exit	Exiting the WebLogic EJB Deployer

Edit menu

The Edit menu contains the following commands:

Command	Description
Deploy	Deploying ejb-jar Files
Options	Configuring the WebLogic EJB Deployer

Help menu

The Help menu contains the following commands:

Command	Description
Using	Displays Help
About	Displays the splash screen

Using the Toolbar

Figure 2-3 shows the toolbar.

Figure 2-2 Toolbar

The Toolbar contains the following command buttons:

Button	Name	Description
	Open	Opening ejb-jar Files
	Save	Saving Changes to ejb-jar Files
	Save As	Saving the Current ejb-jar File to a New Name
	Save All	Saving Changes to ejb-jar Files
	Close	Closing ejb-jar Files
	Close All	Closing ejb-jar Files
	Deploy	Deploying ejb-jar Files

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:

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

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.

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:

Name	Description
keepgenerated	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.
compiler	Specifies the Java compiler to use for generating container classes.

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 EJB 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.

Figure 2-7 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.
6. Close the ejb-jar file.

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:

Name	Description
Look In	Selects a directory.
Directory List	Displays a list of directories and files in the selected directory.
File Name	Type or double-click the name of the ejb-jar file you want to open.
File of Type	Select a file type.

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

Panel Name	Description
ejb-jar panel	Displays information about the ejb-jar file, including the following tabs: <ul style="list-style-type: none">■ Description tab■ Security tab■ Module Initialization tab
Bean Properties Panel	Displays a list of EJBs associated with the selected ejb-jar file, including the name and type of each EJB.
EJB panel	For the selected EJB, displays the bean class, home interface, remote interface, a summary of bean characteristics, and a description.
Method panel	For the selected method, displays the transaction attribute, a list of allowed callers, and a description.

Table 2-1 Panels in the WebLogic EJB Deployer (Continued)

Panel Name	Description
Persistence panel	For the selected EJB, displays persistence information: the primary key class, a list of container-managed fields, and a description.
Security Role References panel	For the selected EJB, displays the security role reference and link to a security role.
Environment Entries panel	For the selected EJB, displays the name, type, value, and a description of one or more environment entries.
EJB Reference panel	For the selected EJB, displays the home interface, remote interface, and a description of the EJB reference.
Resource Factory Reference panel	For the selected EJB, displays the type, sign-on responsibility, JNDI reference link, and a description of the resource factory reference.
JDBC panel	For the selected EJB with container-managed persistence, displays the following tabs: <ul style="list-style-type: none">■ Configuration tab■ Field-Column Map tab■ Finders tab
File panel	For the selected stateful session bean with container-managed persistence, displays the root directory for persistent storage.

Changing Deployment Descriptor Properties

Sun Microsystem'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

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

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 to a New Name

Saving the Current ejb-jar File



To save changes to the ejb-jar file in the current window:

- Click the Save button on the toolbar, or choose Save from the File menu.

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

Saving All Open ejb-jar Files



To save changes to all open ejb-jar files:

- Click the Save All button on the toolbar, or choose Save All from the File menu.

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

Saving the Current ejb-jar File to a New Name



To save an ejb-jar file to a different name:

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

The Save As dialog box appears. Figure 2-9 shows an example of the Save As dialog box.

Figure 2-9 Save As Dialog Box

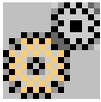
The Save ejb-jar File As dialog box contains the following options:

Name	Description
Look In	Selects a directory.
Directory List	Displays a list of directories and files in the selected directory.
File Name	Specifies the file name.
File of Type	Selects a file type.

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 `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.

3 Panels

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

Panel Name	Description
ejb-jar panel	Information about the <code>ejb-jar</code> file, including the following panels: <ul style="list-style-type: none">■ Description tab■ Security tab■ Module Initialization tab
Bean Properties Panel	Displays a list of EJBs associated with the selected <code>ejb-jar</code> file, including the name and type of each EJB.
EJB panel	For the selected EJB, displays the bean class, home interface, remote interface, a summary of bean characteristics, and a description.
Method panel	For the selected method, displays the transaction attribute, a list of allowed callers, and a description.
Persistence panel	For the selected EJB, displays persistence information: the primary key class, a list of container-managed fields, and a description.
Security Role References panel	For the selected EJB, displays the security role reference and link to a security role.
Environment Entries panel	For the selected EJB, displays the name, type, value, and a description of one or more environment entries.

Table 3-1 Panels in the WebLogic EJB Deployer (Continued)

Panel Name	Description
EJB Reference panel	For the selected EJB, displays the home interface, remote interface, linked JNDI name, and a description of the EJB reference.
Resource Factory Reference panel	For the selected EJB, displays the type, sign-on responsibility, and a description of the resource factory reference.
JDBC panel	For the selected EJB with container-managed persistence, displays the following panels and a description: <ul style="list-style-type: none">■ Configuration tab■ Field-Column Map tab■ Finders tab
File panel	For a selected stateful session bean with container-managed persistence, displays the root directory for persistent storage and a description.

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:

Name	Description	XML Element
Description	Text description of the ejb-jar file.	description

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:

Name	Description	XML Element
Application Roles	Displays a list of any security roles that are available in this environment.	security-role
Description	Text description.	description

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:

Name	Description	XML Element
Class Name	Name of the server implementation class for handling the initialize and release methods.	module-initializer-classes-name

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:

Name	Description	XML Element
	Icon associated with the EJB.	small-icon or default
Name	Name of the EJB.	ejb-name
Type	Type of EJB (Entity or Session).	<input type="checkbox"/> entity <input type="checkbox"/> session

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:

Name	Description	XML Element
Bean class	Fully qualified name of the EJB's class.	<code>ejb-class</code>
Home interface class	Fully qualified name of the EJB's home interface.	<code>home</code>
Remote interface class	Fully qualified name of the EJB's remote interface.	<code>remote</code>
Summary	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.	<ul style="list-style-type: none">■ <code>entity</code>■ <code>session</code>■ <code>session-type</code>■ <code>persistence-type</code>■ <code>transaction-type</code>■ <code>reentrant</code>
Description	Text description of the EJB.	<code>description</code>

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:

Name	Description	XML Element
Transaction Attribute	Specifies the transaction attribute for the EJB. Select of the following values: <ul style="list-style-type: none"> ■ NotSupported ■ Supports ■ Required ■ RequiresNew ■ Mandatory ■ Never 	<code>trans-attribute</code>
Allowed callers	Specifies the role name(s) allowed to invoke the method.	<code>role-name</code>
Description	Text description of the method.	<code>description</code>

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:

Name	Description	XML Element
Primary key type	Fully qualified name of an entity bean's primary key class.	<code>prim-key-class</code>
Primary key field	Primary key field, if specified.	<code>primkey-field</code>
Fields	Container-managed fields.	<code>cmp-field</code>

Name	Description	XML Element
Field	Name of a container-managed field.	field-name
Type	Java data type associated with the field.	N/A
Description	Text description.	description

Security Role References panel

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

Figure 3-8 Security Role References Panel

The Security Role References panel displays the following security role reference (`security-role-ref`) information for the selected EJB:

Name	Description	XML Element
Referenced Role	Name of a security role.	role-name
Application Role	Links a security role reference to a defined security role.	role-link
Description	Text description of the security role.	description

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.

Figure 3-9 Environment Entries Panel

The Environment Entries panel displays the following environment entry (*env-entry*) information for the selected EJB:

Name	Description	XML Element
Name	Name of an EJB's environment entry.	<i>env-entry-name</i>
Type	Java type of the environment entry value that is expected by the EJB's code.	<i>env-entry-type</i>
Value	Value of the EJB's environment entry.	<i>env-entry-value</i>
Description	Text description.	<i>description</i>

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 (*ejb-ref-name*) information for the selected EJB:

Name	Description	XML Element
Home interface class	Home interface of the EJB reference.	Home interface of <i>ejb-ref</i>
Remote interface class	Remote interface of the EJB reference.	Remote interface of <i>ejb-ref</i>
Linked bean JNDI name	Links this EJB reference to another EJB in the <i>ejb-jar</i> file.	<i>ejb-link</i>
Description	Text description.	<i>description</i>

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:

Name	Description	XML Element
Resource factory type	Type of the data source, which is the Java interface (or class) expected to be implemented by the data source.	<i>res-type</i>
Sign-on responsibility	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.	<i>res-auth</i>
Description	Text description.	<i>description</i>

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.

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:

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

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:

Name	Description	XML Element
Field	Name of an EJB field mapped to the table column in this attribute map entry.	<code>bean-field-name</code>

Name	Description	XML Element
Type	Java data type associated with the field.	N/A
Table column	Name of a table column mapped to the field in this attribute map entry.	table-column-name

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:

Name	Description	XML Element
Finder signature	Finder descriptor for finding one or more EJBs.	finder-descriptor

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`.

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):

Name	Description	XML Element
Persistent directory root	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.	<code>persistence-store-directory-root</code>
Description	Text description.	<code>description</code>

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.

4 EJB Deployment Descriptor Reference

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 Microsystem’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 Microsystem’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*.

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:

```
<!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

Element	Description	EJB Role
<code>assembly-descriptor</code>	Contains application-assembly information.	Application Assembler
<code>cmp-field</code>	Describes a container-managed field.	Bean Provider
<code>container-transaction</code>	Specifies how the container must manage transaction scopes for the EJB's method invocations.	Application Assembler
<code>description</code>	Provides text describing the parent element.	Bean Provider or Application Assembler

4 EJB Deployment Descriptor Reference

Table 4-1 List of Standard Deployment Descriptor Elements (Continued)

Element	Description	EJB Role
<code>display-name</code>	Specifies a short name that is intended to be displayed by tools.	Bean Provider
<code>ejb-class</code>	Specifies the fully qualified name of the EJB's class.	Bean Provider
<code>ejb-jar</code>	Root element of the EJB deployment descriptor.	All
<code>ejb-link</code>	Used in the <code>ejb-ref</code> element to specify that an EJB reference is linked to another EJB in the <code>ejb-jar</code> file.	Application Assembler
<code>ejb-name</code>	Specifies an EJB's name.	Bean Provider or Application Assembler
<code>ejb-ref</code>	Declares a reference to another EJB's home.	Bean Provider
<code>ejb-ref-name</code>	Specifies the name of an EJB reference, which is an entry in the EJB's environment.	Bean Provider
<code>ejb-ref-type</code>	Specifies the expected type (entity or session) of the referenced EJB.	Bean Provider
<code>enterprise-beans</code>	Declares one or more EJBs.	Bean Provider
<code>entity</code>	Declares an entity bean.	Bean Provider
<code>env-entry</code>	Declares an EJB's environment entries.	Bean Provider or Application Assembler
<code>env-entry-name</code>	Specifies the name of an EJB's environment entry.	Bean Provider or Application Assembler
<code>env-entry-type</code>	Specifies the fully qualified Java type of the environment entry value that is expected by the EJB's code.	Bean Provider or Application Assembler
<code>env-entry-value</code>	Specifies the value of an EJB's environment entry.	Bean Provider or Application Assembler
<code>field-name</code>	Specifies the name of a container-managed field.	Bean Provider

Table 4-1 List of Standard Deployment Descriptor Elements (Continued)

Element	Description	EJB Role
home	Specifies the fully qualified name of the EJB's home interface.	Bean Provider
large-icon	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.	Bean Provider
method	Denotes a method of an EJB's home or remote interface, or a set of methods.	Application Assembler
method-intf	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.	Application Assembler
method-name	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.	Application Assembler
method-param	Specifies the fully qualified Java type name of a method parameter.	Application Assembler
method-params	Contains a list of the fully qualified Java type names of the method parameters.	Application Assembler
method-permission	Specifies one or more security roles that are allowed to invoke one or more EJB methods.	Application Assembler
persistence-type	Specifies an entity bean's persistence management type (bean- or container-managed).	Bean Provider
prim-key-class	Specifies the fully qualified name of an entity bean's primary key class.	Bean Provider
primkey-field	Specifies the name of the primary key field for an entity with container-managed persistence.	Bean Provider
reentrant	Specifies whether an entity bean is reentrant (True) or not (False).	Bean Provider
remote	Specifies the fully qualified name of the EJB's remote interface.	Bean Provider

Table 4-1 List of Standard Deployment Descriptor Elements (Continued)

Element	Description	EJB Role
<code>res-auth</code>	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.	Bean Provider
<code>res-ref-name</code>	Specifies the name of a resource factory reference.	Bean Provider
<code>res-type</code>	Specifies the type of the data source, which is the Java interface (or class) expected to be implemented by the data source.	Bean Provider
<code>resource-ref</code>	Declares the EJB's reference to an external resource.	Bean Provider
<code>role-link</code>	Links a security role reference to a defined security role.	Application Assembler
<code>role-name</code>	Specifies the name of a security role.	Application Assembler
<code>security-role</code>	Defines a security role.	Application Assembler
<code>security-role-ref</code>	Declares a security role reference in the EJB's code.	Bean Provider
<code>session-type</code>	Specifies the type of a session bean (stateful or stateless).	Bean Provider
<code>session</code>	Declares a session bean.	Bean Provider
<code>small-icon</code>	Specifies the name of a file containing a small (16 x 16) icon image. The file name is relative path within the <code>ejb-jar</code> file.	Bean Provider
<code>transaction-type</code>	Specifies the transaction management type of an EJB (bean- or container-managed).	Bean Provider
<code>trans-attribute</code>	Specifies how the container must manage the transaction boundaries when delegating a method invocation to an EJB's business method.	Application Assembler

Description of Deployment Descriptor Elements

assembly-descriptor

Synopsis	Contains application-assembly information.
Syntax	<code><!ELEMENT assembly-descriptor (security-role*, method-permission*, container-transaction*)></code>
EJB Role	Application Assembler
Used In	<code>ejb-jar</code>

cmp-field

Synopsis	Describes a container-managed field.
Syntax	<code><!ELEMENT cmp-field (description?, field-name)></code>
EJB Role	Bean Provider
Used In	<code>entity</code>

container-transaction

Synopsis	Specifies how the container must manage transaction scopes for the EJB's method invocations.
Syntax	<code><!ELEMENT container-transaction (description?, method+, trans-attribute)></code>
EJB Role	Application Assembler
Used In	<code>assembly-descriptor</code>

description

Synopsis	Provides text describing the parent element.
----------	--

4 *EJB Deployment Descriptor Reference*

Syntax	<code><!ELEMENT description (#PCDATA)></code>
EJB Role	Bean Provider or Application Assembler, and
Used In	<ul style="list-style-type: none">■ <code>cmp-field</code>■ <code>container-transaction</code>■ <code>ejb-jar</code>■ <code>entity</code>■ <code>env-entry</code>■ <code>ejb-ref</code>■ <code>method</code>■ <code>method-permission</code>■ <code>resource-ref</code>■ <code>security-role</code>■ <code>security-role-ref</code>■ <code>session</code>

display-name

Synopsis	Specifies a short name that is intended to be displayed by tools.
Syntax	<code><!ELEMENT display-name (#PCDATA)></code>
EJB Role	Bean Provider
Used In	<ul style="list-style-type: none">■ <code>ejb-jar</code>■ <code>session</code>■ <code>entity</code>

ejb-class

Synopsis	Specifies the fully qualified name of the EJB's class.
Syntax	<code><!ELEMENT ejb-class (#PCDATA)></code>
EJB Role	Bean Provider
Used In	<ul style="list-style-type: none">■ <code>entity</code>

- `session`

ejb-jar

Synopsis	Root element of the EJB deployment descriptor.
Syntax	<code><!ELEMENT ejb-jar (description?, display-name?, small-icon?, large-icon?, enterprise-beans, assembly-descriptor?)></code>
EJB Role	All

ejb-link

Synopsis	Used in the <code>ejb-ref</code> element to specify that an EJB reference is linked to another EJB in the <code>ejb-jar</code> file.
Syntax	<code><!ELEMENT ejb-link (#PCDATA)></code>
Valid Values	The value of the <code>ejb-link</code> element must be the <code>ejb-name</code> of an EJB in the same <code>ejb-jar</code> file, or in another <code>ejb-jar</code> file in the same J2EE application unit.
EJB Role	Application Assembler
Used In	<code>ejb-ref</code>

ejb-name

Synopsis	Specifies an EJB's name.
Syntax	<code><!ELEMENT ejb-name (#PCDATA)></code>
Valid Values	<ul style="list-style-type: none">■ The name must be unique among the names of the EJBs in the same <code>ejb-jar</code> file.■ The name must conform to the lexical rules for an NMTOKEN.
EJB Role	Bean Provider or Application Assembler
Used In	<ul style="list-style-type: none">■ <code>entity</code>■ <code>method</code>■ <code>session</code>

ejb-ref

- Synopsis Declares 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.

Syntax `<!ELEMENT enterprise-beans (session | entity)+>`
EJB Role Bean Provider

entity

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 `persistence-type` is `Container`, and none must not be present if the entity's `persistence-type` is `Bean`.

EJB Role Bean Provider

Used In `enterprise-beans`

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)>`

Valid Values	The name must be a public field of the EJB class or one of its superclasses.
EJB Role	Bean Provider
Used In	<code>cmp-field</code>

home

Synopsis	Specifies the fully qualified name of the EJB's home interface.
Syntax	<code><!ELEMENT home (#PCDATA)></code>
EJB Role	Bean Provider
Used In	<ul style="list-style-type: none">■ <code>ejb-ref</code>■ <code>entity</code>■ <code>session</code>

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 <code>ejb-jar</code> file.
Syntax	<code><!ELEMENT large-icon (#PCDATA)></code>
Valid Values	<ul style="list-style-type: none">■ The image must be either in the JPEG or GIF format.■ The file name must end with the suffix <code>.jpg</code> or <code>.gif</code> respectively.
EJB Role	Bean Provider

method

Synopsis	Denotes a method of an EJB's home or remote interface, or a set of methods.
Syntax	<code><!ELEMENT method (description?, ejb-name, method-intf?, method-name, method-params?)></code>
Valid Values	The <code>ejb-name</code> element must be the name of one of the EJBs declared in the deployment descriptor.
EJB Role	Application Assembler

- Used In
- `method-permission`
 - `container-transaction`

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

Valid Values One of the following values:

- `Bean`
- `Container`

EJB Role Bean Provider

Used In `entity`

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`

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 EJB 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`

- `security-role-ref`

security-role

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

security-role-ref

Synopsis	Declares a security role reference in the EJB's code.
Syntax	<code><!ELEMENT security-role-ref (description?, role-name, role-link?)></code>
Valid Values	<ul style="list-style-type: none">■ The value of the <code>role-name</code> element must be the String used as the parameter to the <code>EJBContext.isCallerInRole(String roleName)</code> method.■ The value of the <code>role-link</code> element must be the name of one of the security roles defined in the <code>security-role</code> elements.
EJB Role	Bean Provider
Used In	<ul style="list-style-type: none">■ <code>entity</code>■ <code>session</code>

session-type

Synopsis	Specifies the type of a session bean (stateful or stateless).
Syntax	<code><!ELEMENT session-type (#PCDATA)></code>
Valid Values	One of the following values: <ul style="list-style-type: none">■ Stateful■ Stateless
EJB Role	Bean Provider

session

Synopsis	Declares a session bean.
Syntax	<code><!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*)></code>
EJB Role	Bean Provider
Used In	<code>enterprise-beans</code>

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 <code>ejb-jar</code> file.
Syntax	<code><!ELEMENT small-icon (#PCDATA)></code>
Valid Values	<ul style="list-style-type: none">■ The image must be either in the JPEG or GIF format.■ The file name must end with the suffix <code>.jpg</code> or <code>.gif</code> respectively.
EJB Role	Bean Provider

transaction-type

Synopsis	Specifies the transaction management type of an EJB (bean- or container-managed).
Syntax	<code><!ELEMENT transaction-type (#PCDATA)></code>
Valid Values	One of the following values: <ul style="list-style-type: none">■ Bean■ Container
EJB Role	Bean Provider
Used In	<code>session</code>

trans-attribute

Synopsis	Specifies how the container must manage the transaction boundaries when delegating a method invocation to an EJB's business method.
Syntax	<code><!ELEMENT trans-attribute (#PCDATA)></code>
Valid Values	One of the following values: <ul style="list-style-type: none">■ NotSupported■ Supports■ Required■ RequiresNew■ Mandatory■ Never
EJB Role	Application Assembler
Used In	<code>container-transaction</code>

ID

Synopsis	<p>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.</p> <p>The EJB architecture does not allow the tools to add the nonstandard information into the EJB deployment descriptor.</p>
Syntax	<pre><!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></pre>

```
<!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>
```


5 WebLogic EJB Extensions Reference

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 and Deploying 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 `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 `ejbc` command line tool, specifying the `-x` parameter and passing this file name (`weblogic-ejb-extensions.xml`). For more information about the `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

Element	Description	EJB Role
<code>attribute-map</code>	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.	Bean Provider
<code>attribute-map-entry</code>	Specifies the mapping between an EJB field and a table column.	Bean Provider
<code>bean-field-name</code>	Specifies the name of an EJB field in an attribute map entry.	Bean Provider
<code>finder-descriptor</code>	Specifies the finder descriptor for finding one or more EJBs.	Bean Provider
<code>jndi-name</code>	Specifies the name for the EJB home that is registered in the Global WLE JNDI namespace.	Bean Provider
<code>module-initializer-class</code>	Specifies the server implementation class for handling the <code>initialize</code> and <code>release</code> methods.	Bean Provider
<code>module-initializer-class-name</code>	Specifies the name of the server implementation class for handling the <code>initialize</code> and <code>release</code> methods.	Bean Provider
<code>persistence-store-descriptor</code>	Specifies the type of persistence storage (in a flat file or a JDBC-managed database) for container-managed persistence.	Bean Provider
<code>persistence-store-directory-root</code>	If persistent storage is kept in 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.	Bean Provider
<code>persistence-store-file</code>	Specifies persistent storage in a flat file.	Bean Provider

Table 5-1 WebLogic Enterprise XML Extensions to the EJB Deployment Descriptor (Continued)

Element	Description	EJB Role
<code>persistence-store-jdbc</code>	Specifies persistent storage in a database that is managed by a WLE JDBC connection pool.	Bean Provider
<code>pool-name</code>	Specifies the name of the JDBC connection pool associated with the database used to store EJB state information.	Bean Provider
<code>query-grammar</code>	Specifies the query expression for a finder method.	Bean Provider
<code>table-column-name</code>	Specifies the name of a table column in an attribute map entry.	Bean Provider
<code>table-name</code>	Specifies the name of the table used to store EJB state information.	Bean Provider
<code>trans-timeout-seconds</code>	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.	Bean Provider
<code>weblogic-deployment-params</code>	Specifies deployment options for the WebLogic Server.	Bean Provider
<code>weblogic-ejb-extensions</code>	Root element of the WLE XML extensions to the deployment descriptor.	Bean Provider
<code>weblogic-enterprise-bean</code>	Specifies deployment information for a bean that is specific to WLE.	Bean Provider
<code>weblogic-version</code>	Specifies the WLE version.	Bean Provider

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:

```
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:

```
(> 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 `myEJBBean.findBigAccounts(amount)`, will return a list of all EJBs whose `balance` attribute is greater than the value of `amount`.

jndi-name

Synopsis	Specifies the name for the EJB home that is registered in the Global WLE JNDI namespace.
Syntax	<code><!ELEMENT jndi-name (#PCDATA)></code>
Description	<p>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.</p> <p>This name can differ from the <code>ejb-name</code> element specified in the standard EJB XML. Whereas the <code>ejb-name</code> element must be unique only among the names of the EJBs in the same <code>ejb-jar</code> file, however, the JNDI name must be unique among <i>all</i> global factory names in a WLE domain, including EJB homes, CORBA factories, and RMI-named objects.</p>
EJB Role	Bean Provider
Used In	<code>weblogic-deployment-params</code>
Example	<pre><weblogic-<i>ejb-extensions</i>> <weblogic-<i>enterprise-bean</i>> <ejb-name>Trader</ejb-name> <weblogic-<i>deployment-params</i>> <jndi-name>EJB_TraderHome</jndi-name> </weblogic-<i>deployment-params</i>> </weblogic-<i>enterprise-bean</i>> </weblogic-<i>ejb-extensions</i>></pre>

module-initializer-class

Synopsis	Specifies the server implementation class for handling the <code>initialize</code> and <code>release</code> methods.
Syntax	<code><!ELEMENT module-initializer-class (module-initializer-class-name)></code>
Description	<p>The syntax for specifying a Server implementation class for handling the <code>initialize</code> and <code>release</code> methods is specified in the EJB XML deployment descriptor extensions. These XML elements are processed by the <code>ejbc</code> command. The WLE EJB framework parses the XML at run time and performs the startup/shutdown processing.</p> <p>For a complete description of startup and shutdown handling in WLE, see <i>Starting and Shutting Down Applications</i>.</p>
EJB Role	Bean Provider
Used In	<code>weblogic-ejb-extensions</code>

module-initializer-class-name

Synopsis	Specifies the fully qualified name of the server implementation class for handling the <code>initialize</code> and <code>release</code> methods.
Syntax	<code><!ELEMENT module-initializer-class-name (#PCDATA)></code>
EJB Role	Bean Provider
Used In	<code>module-initializer-class</code>
Example	<pre><weblogic-ejb-extensions> ... <module-initializer-class> <moduleinitializer-class-name>ServerImpl </moduleinitializer-class-name> </module-initializer-class> </weblogic-ejb-extensions></pre>

persistence-store-descriptor

Synopsis	Specifies the type of persistence storage (in a flat file or a JDBC-managed database) for container-managed persistence.
Syntax	<code><!ELEMENT persistence-store-descriptor (description?, (persistence-store-file persistence-store-jdbc)?)></code>
Description	<p>For container-managed persistence, WLE EJB Container supports storing an EJB in either of the following formats:</p> <ul style="list-style-type: none"> ■ a flat file (the <code>persistence-store-file</code> element) ■ a database that is managed with a WLE JDBC connection pool (the <code>persistence-store-jdbc</code> element) <p>Note: Database persistence is <i>required</i> for an EJB to fully cooperate in a WLE global transaction. File persistence is intended primarily for development and prototyping.</p> <p>The standard EJB XML descriptor provides XML elements for specifying which fields in the entity bean are to be persisted (the <code>cmp-field</code> element), and information about the primary key. The WebLogic XML extensions provide additional information for mapping to the persistent store.</p> <p>The code for implementing the persistence is generated by the EJB compiler, based on the bean's deployment descriptors.</p>
EJB Role	Bean Provider
Used In	<code>weblogic-deployment-params</code>
Example	<p>Using JDBC persistence, the following sample XML code shows how to define the automatic saving of two fields (<code>accountId</code>, <code>balance</code>) in an entity bean to a database table (<code>ejbAccounts</code>) using a connection pool (<code>ejbPool</code>):</p>

```
<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-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
```

EJB Role Bean Provider
Used In

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-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

pool-name

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

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

Description The query grammar syntax has an expression prefix notation of:

`[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' (> 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`

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

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
- container-transaction element 4-7
- 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

- container-transaction element 4-7
- description element 4-7
- display-name element 4-8
- DOCTYPE declaration 4-3
- editing properties 2-12
- EJB roles 4-2
- ejb-class element 4-8
- ejb-jar element 4-9
- ejb-link element 4-9
- ejb-name element 4-9
- ejb-ref element 4-10
- ejb-ref-name element 4-10
- ejb-ref-type element 4-10
- enterprise-beans element 4-10
- entity element 4-11
- env-entry element 4-11
- env-entry-name element 4-11
- env-entry-type element 4-12
- env-entry-value element 4-12
- extensions (see extensions) 5-2
- field-name element 4-12
- file formats 4-3
- finder-descriptor element 5-6
- home element 4-13
- ID mechanism 4-21
- jndi-name element 5-7
- large-icon element 4-13
- method element 4-13
- method-intf element 4-14
- method-name element 4-14
- method-param element 4-14

- method-params element 4-15
- method-permission element 4-15
- module-initializer-class element 5-8
- module-initializer-class-name element 5-8
- navigating 2-11
- 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
- query-grammar element 5-12
- reentrant element 4-16
- remote element 4-17
- res-auth element 4-17
- 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
- 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
- table-column-name element 5-14
- table-name element 5-14
- transaction-type element 4-20
- trans-attribute element 4-21
- 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

deployment steps 2-9

description element 4-7

- Description tab 3-2
- display-name element 4-8
- DOCTYPE declaration 4-3
- documentation, where to find it ii

E

- Edit Expression dialog 3-11
- Edit menu 2-4
 - Deploy command 2-4
 - Options command 2-4
- editing deployment descriptor properties 2-12
- EJB panel 3-5
- EJB Reference panel 3-8
- EJB roles 4-2
- ejb-class element 4-8
- ejb-jar Contents List 2-11
- ejb-jar element 4-9
- ejb-jar files
 - closing 2-16
 - deploying 2-15
 - opening 2-10
 - saving changes to 2-13
 - saving to different names 2-14
- ejb-jar panel 3-2
 - Description tab 3-2
 - Module Initialization tab 3-3
 - Security tab 3-3
- ejb-link element 4-9
- ejb-name element 4-9
- ejb-ref element 4-10
- ejb-ref-name element 4-10
- ejb-ref-type element 4-10
- enterprise-beans element 4-10
- entity element 4-11
- env-entry element 4-11
- env-entry-name element 4-11
- env-entry-type element 4-12
- env-entry-value element 4-12
- Environment Entries panel 3-7

exiting 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

F

Field-Column Map tab 3-10
field-name element 4-12
file formats 4-3
File menu 2-3
 Close All command 2-4
 Close command 2-4
 Exit command 2-4
 Open command 2-3
 Save All command 2-4
 Save As command 2-4
 Save command 2-3
File panel 3-12

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

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