



# BEA WebLogic Enterprise

## EJB XML Reference

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.

## EJB XML Reference

<b>Document Edition</b>	<b>Date</b>	<b>Software Version</b>
5.0	December 1999	BEA WebLogic Enterprise 5.0

---

# Contents

## About This Document

What You Need to Know .....	vii
e-docs Web Site .....	viii
How to Print the Document .....	viii
Related Information .....	viii
Contact Us! .....	ix
Documentation Conventions .....	ix

## 1. EJB Deployment Descriptor Reference

About Deployment Descriptors .....	1-2
EJB Roles and Deployment Descriptors .....	1-2
File Formats .....	1-3
DOCTYPE Declaration .....	1-3
Summary of Deployment Descriptor Elements .....	1-3
Description of Deployment Descriptor Elements .....	1-7
assembly-descriptor .....	1-7
cmp-field .....	1-7
container-transaction .....	1-7
description .....	1-7
display-name .....	1-8
ejb-class .....	1-8
ejb-jar .....	1-9
ejb-link .....	1-9
ejb-name .....	1-9
ejb-ref .....	1-10
ejb-ref-name .....	1-10
ejb-ref-type .....	1-10

---

enterprise-beans .....	1-10
entity .....	1-11
env-entry .....	1-11
env-entry-name .....	1-11
env-entry-type .....	1-12
env-entry-value .....	1-12
field-name .....	1-12
home .....	1-13
large-icon .....	1-13
method .....	1-13
method-intf .....	1-14
method-name .....	1-14
method-param .....	1-14
method-params .....	1-15
method-permission .....	1-15
persistence-type .....	1-15
prim-key-class .....	1-16
primkey-field .....	1-16
reentrant .....	1-16
remote .....	1-17
res-auth .....	1-17
res-ref-name .....	1-17
res-type .....	1-17
resource-ref .....	1-18
role-link .....	1-18
role-name .....	1-18
security-role .....	1-19
security-role-ref .....	1-19
session-type .....	1-19
session .....	1-20
small-icon .....	1-20
transaction-type .....	1-20
trans-attribute .....	1-21
ID .....	1-21

---

## 2. WebLogic EJB Extensions Reference

About the WebLogic EJB Extensions .....	2-2
Summary of WebLogic EJB Extensions .....	2-3
Description of WebLogic EJB Extension Elements.....	2-5
attribute-map .....	2-5
attribute-map-entry .....	2-5
bean-field-name .....	2-5
finder-descriptor .....	2-6
jndi-name .....	2-7
module-initializer-class .....	2-8
module-initializer-class-name .....	2-8
persistence-store-descriptor .....	2-9
persistence-store-directory-root .....	2-10
persistence-store-file .....	2-11
persistence-store-jdbc .....	2-11
pool-name .....	2-12
query-grammar .....	2-12
table-column-name .....	2-14
table-name .....	2-14
trans-timeout-seconds .....	2-14
weblogic-deployment-params .....	2-14
weblogic-ejb-extensions .....	2-15
weblogic-enterprise-bean .....	2-15
weblogic-version .....	2-15

## Index



---

# About This Document

This document provides XML reference information for configuring EJB applications that run under WebLogic Enterprise (WLE).

This document covers the following topics:

- Chapter 1, “EJB Deployment Descriptor Reference,” 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).
- Chapter 2, “WebLogic EJB Extensions Reference,” describes the WLE XML extensions for EJB applications. 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 (`weblogic-ejb-extensions.xml`).

## What You Need to Know

This document is intended for Bean Providers, Application Assemblers, and Deployers who need to review or change the configuration for EJBs in EJB applications. It assumes that you are knowledgeable about EJB programming and familiar with Sun Microsystem’s *Enterprise JavaBeans Specification 1.1*.

---

# 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
<b>boldface text</b>	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.

---

<b>Convention</b>	<b>Item</b>
<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
<b>monospace boldface text</b>	Identifies significant words in code. <i>Example:</i> void <b>commit</b> ( )
<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"><li>■ That an argument can be repeated several times in a command line</li><li>■ That the statement omits additional optional arguments</li><li>■ That you can enter additional parameters, values, or other information</li></ul> 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 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 2, "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 1-1 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

# 1 EJB Deployment Descriptor Reference

---

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

<b>Element</b>	<b>Description</b>	<b>EJB Role</b>
<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 1-1 List of Standard Deployment Descriptor Elements (Continued)**

<b>Element</b>	<b>Description</b>	<b>EJB Role</b>
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 1-1 List of Standard Deployment Descriptor Elements (Continued)**

<b>Element</b>	<b>Description</b>	<b>EJB Role</b>
<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>&lt;!ELEMENT assembly-descriptor (security-role*, method-permission*, container-transaction*)&gt;</code>
EJB Role	Application Assembler
Used In	<code>ejb-jar</code>

## cmp-field

Synopsis	Describes a container-managed field.
Syntax	<code>&lt;!ELEMENT cmp-field (description?, field-name)&gt;</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>&lt;!ELEMENT container-transaction (description?, method+, trans-attribute)&gt;</code>
EJB Role	Application Assembler
Used In	<code>assembly-descriptor</code>

## description

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

# 1 *EJB Deployment Descriptor Reference*

---

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

## **display-name**

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

## **ejb-class**

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

- session

## ejb-jar

Synopsis	Root element of the EJB deployment descriptor.
Syntax	<code>&lt;!ELEMENT.ejb-jar (description?, display-name?, small-icon?, large-icon?, enterprise-beans, assembly-descriptor?)&gt;</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>&lt;!ELEMENT.ejb-link (#PCDATA)&gt;</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>&lt;!ELEMENT.ejb-name (#PCDATA)&gt;</code>
Valid Values	<ul style="list-style-type: none"><li>■ The name must be unique among the names of the EJBs in the same <code>ejb-jar</code> file.</li><li>■ The name must conform to the lexical rules for an NMTOKEN.</li></ul>
EJB Role	Bean Provider or Application Assembler
Used In	<ul style="list-style-type: none"><li>■ <code>entity</code></li><li>■ <code>method</code></li><li>■ <code>session</code></li></ul>

## 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	<b>Bean Provider</b>
Used In	<code>cmp-field</code>

## home

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

**Syntax** `<!ELEMENT home (#PCDATA)>`

**EJB Role** **Bean Provider**

**Used In**

- `ejb-ref`
- `entity`
- `session`

## large-icon

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

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

**Valid Values**

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

**EJB Role** **Bean Provider**

## method

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

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

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

**EJB Role** **Application Assembler**

- 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	<code>&lt;!ELEMENT method-params (method-param*)&gt;</code>
EJB Role	Application Assembler
Used In	<code>method</code>

## method-permission

Synopsis	Specifies one or more security roles that are allowed to invoke one or more EJB methods.
Syntax	<code>&lt;!ELEMENT method-permission (description?, role-name+, method+)&gt;</code>
Valid Values	<ul style="list-style-type: none"><li>■ The security roles used in the method-permission element must be defined in the <code>security-role</code> element of the deployment descriptor.</li><li>■ The methods must be methods defined in the EJB's remote and/or home interfaces.</li></ul>
EJB Role	Application Assembler
Used In	<code>assembly-descriptor</code>

## persistence-type

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

## **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.

# 1 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 `<!ELEMENT security-role (description?, role-name)>`

EJB Role Application Assembler

Used In `assembly-descriptor`

## security-role-ref

Synopsis Declares a security role reference in the EJB's code.

Syntax `<!ELEMENT security-role-ref (description?, role-name, role-link?)>`

- Valid Values
- The value of the `role-name` element must be the String used as the parameter to the `EJBContext.isCallerInRole(String roleName)` method.
  - The value of the `role-link` element must be the name of one of the security roles defined in the `security-role` elements.

EJB Role Bean Provider

Used In

- `entity`
- `session`

## session-type

Synopsis Specifies the type of a session bean (stateful or stateless).

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

Valid Values One of the following values:

- Stateful
- Stateless

EJB Role Bean Provider

## session

Synopsis Declares a session bean.

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

EJB Role Bean Provider

Used In `enterprise-beans`

## small-icon

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

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

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

EJB Role Bean Provider

## transaction-type

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

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

Valid Values One of the following values:

- Bean
- Container

EJB Role Bean Provider

Used In `session`

## 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>&lt;!ELEMENT trans-attribute (#PCDATA)&gt;</code>
Valid Values	One of the following values: <ul style="list-style-type: none"><li>■ <b>NotSupported</b></li><li>■ <b>Supports</b></li><li>■ <b>Required</b></li><li>■ <b>RequiresNew</b></li><li>■ <b>Mandatory</b></li><li>■ <b>Never</b></li></ul>
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>&lt;!ATTLIST assembly-descriptor id ID #IMPLIED&gt; &lt;!ATTLIST cmp-field id ID #IMPLIED&gt; &lt;!ATTLIST container-transaction id ID #IMPLIED&gt; &lt;!ATTLIST description id ID #IMPLIED&gt; &lt;!ATTLIST display-name id ID #IMPLIED&gt; &lt;!ATTLIST ejb-class id ID #IMPLIED&gt;</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>
```



# 2 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 1, "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 2-1 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 2-1 WebLogic Enterprise XML Extensions to the EJB Deployment Descriptor (Continued)**

<b>Element</b>	<b>Description</b>	<b>EJB Role</b>
<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>&lt;!ELEMENT jndi-name (#PCDATA)&gt;</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> &lt;weblogic-<i>ejb-extensions</i>&gt;   &lt;weblogic-<i>enterprise-bean</i>&gt;     &lt;ejb-name&gt;Trader&lt;/ejb-name&gt;     &lt;weblogic-<i>deployment-params</i>&gt;       &lt;jndi-name&gt;EJB_TraderHome&lt;/jndi-name&gt;     &lt;/weblogic-<i>deployment-params</i>&gt;   &lt;/weblogic-<i>enterprise-bean</i>&gt; &lt;/weblogic-<i>ejb-extensions</i>&gt; </pre>

### module-initializer-class

Synopsis	Specifies the server implementation class for handling the <code>initialize</code> and <code>release</code> methods.
Syntax	<code>&lt;!ELEMENT module-initializer-class (module-initializer-class-name)&gt;</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>&lt;!ELEMENT module-initializer-class-name (#PCDATA)&gt;</code>
EJB Role	Bean Provider
Used In	<code>module-initializer-class</code>
Example	<pre>&lt;weblogic-ejb-extensions&gt; ...   &lt;module-initializer-class&gt;     &lt;moduleinitializer-class-name&gt;ServerImpl   &lt;/moduleinitializer-class-name&gt; &lt;/module-initializer-class&gt; &lt;/weblogic-ejb-extensions&gt;</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>&lt;!ELEMENT persistence-store-descriptor (description?, (persistence-store-file   persistence-store-jdbc)?)&gt;</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"> <li>■ a flat file (the <code>persistence-store-file</code> element)</li> <li>■ a database that is managed with a WLE JDBC connection pool (the <code>persistence-store-jdbc</code> element)</li> </ul> <p><b>Note:</b> 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 1-7
  - attribute-map element 2-5
  - attribute-map-entry element 2-5
- B**
- bean-field-name element 2-5
- C**
- cmp-field element 1-7
  - container-transaction element 1-7
  - customer support contact information ix
- D**
- deployment descriptor
    - about deployment descriptors 1-2
    - assembly-descriptor element 1-7
    - attribute-map element 2-5
    - attribute-map-entry element 2-5
    - bean-field-name element 2-5
    - cmp-field element 1-7
    - container-transaction element 1-7
    - description element 1-7
    - display-name element 1-8
    - DOCTYPE declaration 1-3
    - EJB roles 1-2
    - ejb-class element 1-8
    - ejb-jar element 1-9
    - ejb-link element 1-9
    - ejb-name element 1-9
    - ejb-ref element 1-10
    - ejb-ref-name element 1-10
    - ejb-ref-type element 1-10
    - enterprise-beans element 1-10
    - entity element 1-11
    - env-entry element 1-11
    - env-entry-name element 1-11
    - env-entry-type element 1-12
    - env-entry-value element 1-12
    - extensions (see extensions) 2-2
    - field-name element 1-12
    - file formats 1-3
    - finder-descriptor element 2-6
    - home element 1-13
    - ID mechanism 1-21
    - jndi-name element 2-7
    - large-icon element 1-13
    - method element 1-13
    - method-intf element 1-14
    - method-name element 1-14
    - method-param element 1-14
    - method-params element 1-15
    - method-permission element 1-15
    - module-initializer-class element 2-8
    - module-initializer-class-name element 2-8
    - persistence-store element 2-9
    - persistence-store-directory-root element 2-10
    - persistence-store-file element 2-11

---

- persistence-store-jdbc element 2-11
- persistence-type element 1-15
- pool-name element 2-12
- prim-key-class element 1-16
- primkey-field element 1-16
- query-grammar element 2-12
- reentrant element 1-16
- remote element 1-17
- res-auth element 1-17
- resource-ref element 1-18
- res-ref-name element 1-17
- res-type element 1-17
- role-link element 1-18
- role-name element 1-18
- security-role element 1-19
- security-role-ref element 1-19
- session element 1-20
- session-type element 1-19
- small-icon element 1-20
- table-column-name element 2-14
- table-name element 2-14
- transaction-type element 1-20
- trans-attribute element 1-21
- trans-timeout-seconds element 2-14
- weblogic-deployment-params element 2-14
  - weblogic-ejb-extensions element 2-15
  - weblogic-enterprise-bean element 2-15
  - weblogic-version element 2-15

- description element 1-7
- display-name element 1-8
- DOCTYPE declaration 1-3
- documentation, where to find it viii

## E

- EJB roles 1-2
- ejb-class element 1-8
- ejb-jar element 1-9
- ejb-link element 1-9
- ejb-name element 1-9

- ejb-ref element 1-10
- ejb-ref-name element 1-10
- ejb-ref-type element 1-10
- enterprise-beans element 1-10
- entity element 1-11
- env-entry element 1-11
- env-entry-name element 1-11
- env-entry-type element 1-12
- env-entry-value element 1-12
- extensions 2-2
  - about WLE XML extensions 2-2
  - attribute-map element 2-5
  - attribute-map-entry element 2-5
  - bean-field-name element 2-5
  - finder-descriptor element 2-6
  - jndi-name element 2-7
  - module-initializer-class element 2-8
  - module-initializer-class-name element 2-8
  - persistence-store element 2-9
  - persistence-store-directory-root element 2-10
  - persistence-store-file element 2-11
  - persistence-store-jdbc element 2-11
  - pool-name element 2-12
  - query-grammar element 2-12
  - table-column-name element 2-14
  - table-name element 2-14
  - trans-timeout-seconds element 2-14
  - weblogic-deployment-params element 2-14
    - weblogic-ejb-extensions element 2-15
    - weblogic-enterprise-bean element 2-15
    - weblogic-version element 2-15

## F

- field-name element 1-12
- file formats 1-3
- finder-descriptor element 2-6

---

## H

home element 1-13

## I

ID mechanism 1-21

## J

jndi-name element 2-7

## L

large-icon element 1-13

## M

method element 1-13  
method-intf element 1-14  
method-name element 1-14  
method-param element 1-14  
method-params element 1-15  
method-permission element 1-15  
module-initializer-class element 2-8  
module-initializer-class-name element 2-8

## P

persistence-store element 2-9  
persistence-store-directory-root element 2-10  
persistence-store-file element 2-11  
persistence-store-jdbc element 2-11  
persistence-type element 1-15  
pool-name element 2-12  
prim-key-class element 1-16  
primkey-field element 1-16  
printing product documentation viii

## Q

query-grammar element 2-12

## R

reentrant element 1-16  
related information viii  
remote element 1-17  
res-auth element 1-17  
resource-ref element 1-18  
res-ref-name element 1-17  
res-type element 1-17  
role-link element 1-18  
role-name element 1-18

## S

security-role element 1-19  
security-role-ref element 1-19  
session element 1-20  
session-type element 1-19  
small-icon element 1-20  
support  
    technical ix

## T

table-column-name element 2-14  
table-name element 2-14  
transaction-type element 1-20  
trans-attribute element 1-21  
trans-timeout-seconds element 2-14

## W

weblogic-deployment-params element 2-14  
weblogic-ejb-extensions element 2-15  
weblogic-enterprise-bean element 2-15  
weblogic-version element 2-15

## X

XML extensions 2-2