Contents

12. Connection

13. Connector Connection Pool Runtime

14. ConsoleLink

15. Conversion

16. Customize This View

17. Custom Realm

18. Deployment Descriptor Editors

19. Domain

20. Domain Log Filter

21. EJB Component

31. Group

32. JDBC Connection Pool

33. JDBC Connection Pool Runtime

34. JDBC Data Sources

35. JDBC MultiPools

36. JDBC Transaction Data Sources

37. JMS Connection Consumer

38. JMS Connection Factories

39. JMS Connection Runtime

40. JMS Consumer Runtime

41. JMS Destinations

42. JMS Destination Key

43. JMS Destination Runtime

44. JMS Durable Subscriber Runtime

45. JMS File Store

46. JMSJDBCStore

47. JMS Producer Runtime

48. JMS Queue

49. JMS Runtime

50. JMS Server

51. JMS Server Runtime

52. JMS Session Pools

53. JMS Sessionpool Runtime

54. JMS Stores

55. JMS Templates

56. JMS Topic

57. Messaging Bridge

58. JMS Bridge Destination

59. General Bridge Destination

60. JNDIBinding

61. JNDI Context

62. Jolt

63. Jolt Connection Pool

64. Jolt Connection Pool Runtime

65. Web Application Deployment Descriptor Editor Help

66. JTA

67. JTA Transaction

68. LDAP Realm (Deprecated)

69. Machine

70. MailSession

71. MBean

72. Message Driven EJB Runtime

73. NTRealm

74. Preferences

75. RDBMS Realm

76. RDBMS Deployment Descriptors

83. Shutdown Class

84. SNMP Trap Runtime

85. SNMP Runtime

86. SNMP Attribute Change

87. SNMP Attribute Change Runtime

88. SNMP Counter Monitor

106.Transaction Name Runtime

107.Transaction Resource Runtime

108.UnixMachine

109.UnixRealm

110.User

111.VirtualHost

112.Web Applications

113.WebApp Component Runtime

114. Web Application Deployment Descriptor Editor Help

115. Web Application Deployment Descriptor Editor Help

116. Web Application Deployment Descriptor Editor Help

117.WLEC Connection Pool Runtime

118.WLEC Connection Pool

119.WLEC Connection

120.XML Entity Spec Registry Entry

121.XML Registry

122.XML Registry Entry

BEA Systems WebLogic Server Administration Console Help

Copyright

Restricted Rights Legend

Trademarks or Service Marks

BEA WebLogic Server Administration Console Online Help

Part Number	Document Date	Software Version
N/A	<Month> <Date>, 2001	BEA WebLogic Server Version 6.X

Copyright

Restricted Rights Legend

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. 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 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 WebLogic Server Administration Guide

Document Date	Software Version
December 19, 2001	BEA WebLogic Server Version 6.1

About This Document

This document introduces the BEA WebLogic Server™ Administration Console features and the procedures used in the administration of a BEA WebLogic Server domain.

The document is organized as follows:

- There is one chapter for each node in the Administration Console
- Chapters may include information on the attributes that can be set or about which information is available in the node, procedures for creating and deploying entities, and links to additional programming and administrative information.

Audience

This document is written for Systems Administrators and operators who will be managing one or more WebLogic Server domains. It is assumed that readers know Web technologies.

e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation or go directly to the WebLogic Server Product Documentation page at http://e-docs.bea.com/wls/docs61.

How to Print the Document

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

A PDF version of this document is available on the WebLogic Server 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 Server documentation Home page, click Download Documentation, and select the document you want to print.

Adobe Acrobat Reader is available at no charge from the Adobe Web site at http://www.adobe.com.

Related Information

The BEA corporate Web site provides all documentation for WebLogic Server.

Contact Us!

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

In your e-mail message, please indicate the software name and version you are using, as well as the title and document date of your documentation. If you have any questions about this version of BEA WebLogic Server, or if you have problems installing and running BEA WebLogic Server, contact BEA Customer Support through BEA WebSupport at http://www.bea.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	Usage
Ctrl+Tab	Keys you press simultaneously.
italics	Emphasis and book titles.

`monospace text`	Code samples, commands and their options, Java classes, data types, directories, and file names and their extensions. Monospace text also indicates text that you enter from the keyboard.
	Examples:
	`import java.util.Enumeration;`
	`chmod u+w *`
	`config/examples/applications`
	`.java`
	`config.xml`
	`float`

`monospace italic text`	Variables in code. *Example*: `String` *`CustomerName`*`;`

Convention	Usage
UPPERCASE TEXT	Device names, environment variables, and logical operators. *Example*s: LPT1 BEA_HOME OR
{ }	A set of choices in a syntax line.
[ ]	Optional items in a syntax line. *Example*: java utils.MulticastTest -n *name* -a *address* [-p *portnumber*] [-t *timeout*] [-s *send*]
|	Separates mutually exclusive choices in a syntax line. *Example*: java weblogic.deploy [list|deploy|undeploy|update] password {application} {source}
...	Indicates one of the following in a command line: ■ An argument can be repeated several times in the command line. ■ The statement omits additional optional arguments. ■ You can enter additional parameters, values, or other information
. . .	Indicates the omission of items from a code example or from a syntax line.

1 ACL

The following procedures describe how to use the Administration Console to set the attributes for creating and managing Acess Control Lists (ACLs). To read more about ACLs, please see Managing Security in the *Administration Guide*.

Create a New ACL

1. Click the ACLs node in the left pane. The ACLs table displays in the right-hand pane showing all the ACLs in the domain.
2. Enter a value in the New ACL Name field.
3. Click the Create button. A dialog displays in the right pane showing information associated with creating a new ACL.
4. Enter values in the Permission, Grant to User, and Grant to Group fields to specify what permissions to assign and which users and groups should receive those permissions. Add a space between multiple values in each field.
5. Click the Update Permission button to create the ACL.

For additional information about creating new ACLs, please see Managing Security in the *Administration Guide*.

Modify an ACL

1. Click the ACLs node in the left pane. The ACLs table displays in the right pane showing all the ACLs in the domain.
2. Click the name of the ACL you will be modifying.
3. Change values in the Permission, Grant to User, and Grant to Group fields to specify what permissions to assign and which users and groups should receive those permissions. Add a space between multiple values in each field.
4. Click the Update Permission button to modify the ACL.

For additional information about modifying ACLs, please see Managing Security in the *Administration Guide*, .

Permissions

Attribute	Description	Range of Values	Default Value
Permissions	This attribute allows a certain type of access to an individual or a group	Permissions include Write, Read, Modify	Null
Grant to Users	This attribute allows a certain type of access to an individual	Permissions include Write, Read, Modify	Null
Grant to Groups	This attribute allows a certain type of access to a group	Permissions include Write, Read, Modify	Null

For additional information about ACLs, please see Managing Security in the *Administration Guide.*

2 ACL Permission

The following table describes the attributes you use to set Permissions in the Acess Control Lists (ACLs). To read more about ACLs, please see Managing Security in the *Administration Guide*.

Permissions

Attribut e	Description	Range of Values	Default Value
Permissio ns	This attribute allows a certain type of access to an individual or a group	Permissions include Write, Read, Modify	Null
Grant to Users	This attribute allows a certain type of access to an individual	Permissions include Write, Read, Modify	Null
Grant to Groups	This attribute allows a certain type of access to a group	Permissions include Write, Read, Modify	Null

To read more about ACLs, please see Managing Security in the *Administration Guide*.

3 Applications

The following procedure describes how to use the Administration Console to set the attributes for installing and deploying a new application. For additional information about Applications, please see Deploying Applications in the *Administration Guide*.

Install a New Application

1. Click the Applications node in the left-hand pane. The Applications table displays in the right pane showing all the applications installed in the domain.
2. Click the Configure a New Application link to open the Create a New Application page.
3. Fill in the configuration information as follows:
   - Enter the name of the application in the Name field.
   - Enter the path of the .ear, . jar, or .war application in the Path field.
   - Select the Deployment box if you want to deploy the new application.
4. Click the Create button to install the application. The new application is added under the Applications node in the left pane.

Note that if you have stored applications in the `config/`*`mydomain`*`/applications` directory, this procedure will automatically deploy them as well. In production mode, WebLogic Server will not detect applications newly placed in the `config/`*`mydomain`*`/applications` directory, but applications that were detected in the directory before switching to production mode will continue to deploy automatically.

For more information about the `config/`*`mydomain`*`/applications` directory, and application deployment in general, see Deploying Applications in the *Administration Guide*

Configuration

Attribut e	**Description**	**Range of Values**	**Default Value**
Name	This attribute reports the name of the application.	String	N/A
Path	This attribute reports the complete path of this application.	String	N/A
Deployed	This attribute reports the deployment status of this application.		N/A

Notes

Attribut e	**Description**	**Range of Values**	**Default Value**
Notes	This attribute provides a space for optional user supplied information.	The value must be an alphanumeric string.	Null

For additional information about Applications, please see Deploying Applications in the *Administration Guide*

4 Basic Realm

By default, this pane allows the user to sort the objects by the following criteria:

- Name

5 Caching Realm

The following procedures describe how to use the Administration Console to set the attributes for configuring and managing Caching Realms. Read about Caching Realms in Managing Security in the *Administration Guide.*

Configure a New Caching Realm

1. Click the Caching Realms node in the left pane. The Caching Realms table displays in the right pane showing all the caching realms defined in the domain.
2. Click the Configure a New Caching Realm text link. A dialog displays in the right pane showing the tabs associated with configuring a new caching realm.
3. Enter values in the Name and Basic Realm attribute fields. Click the Case Sensitive Cache checkbox to enable case sensitivity.
4. Click Create to create a caching realm instance with the name you specified in the Name field. The new instance is added under the Caching Realms node in the left pane.
5. Click the ACL, Authentication, Groups, Users, and Permission tabs and change the attribute fields or accept the default values as assigned.
6. Click Apply to save any changes you made.

Clone a Caching Realm

1. Click the Caching Realms node in the left pane. The Caching Realms table displays in the right pane showing all the caching realms defined in the domain.
2. Click the Clone icon in the row of the caching realm you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a caching realm.
3. Enter values in the Name and Basic Realm attribute fields. Click the Case Sensitive Cache checkbox to enable case sensitivity.
4. Click the Create button in the lower right corner to create a caching realm instance with the name you specified in the Name field. The new instance is added under the Caching Realms node in the left pane.
5. Click the ACL, Authentication, Groups, Users, and Permission tabs and change the attribute fields or accept the default values as assigned.
6. Click Apply to save any changes you made.

Delete a Caching Realm

1. Click the Caching Realms node in the left pane. The Caching Realms table displays in the right pane showing all the caching realms defined in the domain.
2. Click the Delete icon in the row of the caching realm you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click Yes to delete the caching realm. The caching realm icon under the Caching Realms node is deleted.

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	Displays the active security realm. This field can not be changed.	String	MyCachingRealm
Basic Realm	The name of the alternate security realm or custom security realm being used with the Caching realm.	List	None

Case Sensitive Cache	Defines whether the specified security realm is case-sensitive. By default, this field is enabled: the realm is case-sensitive. To use a realm that is not case-sensitive (such as the Windows NT and LDAP security realms), you must disable this field.	Boolean,
		Selected = True
		Unselected = False
			True

ACL

Attribute	Description	Range of Values	Default Value

Enable ACL Cache	Option for enabling the ACL cache.	Boolean,
		Selected = True
		Unselected = False
			True

ACLCache Size	The maximum number of ACL lookups to cache.	This attribute should be a prime number for best lookup performance.	211
ACLCache TTLPositive	The number of seconds to retain the results of a successful lookup.	Integer	60
ACLCache TTLNegative	The number of seconds to retain the results of an unsuccessful lookup.	Integer	10

Authentication

Attribute	Description	Range of Values	Default Value

Enable Authentication Cache	This attribute is set true if the authentication cache is enabled.	Boolean,
		Selected = True
		Unselected = False
			True

Authentication Cache Size	This attribute sets the maximum number of Authenticate requests to cache.	This attribute should be a prime number for best lookup performance.	211
Authentication Cache TTLPositive	This attribute sets the number of seconds to retain the result of a successful lookup.	Integer	60
Authentication Cache TTLNegative	This attribute sets the number of seconds to retain the result of an unsuccessful lookup.	Integer	10

Groups

Attribute	Description	Range of Values	Default Value

Enable Group Cache	This attribute is set true if group cache is enabled.	Boolean,
		Selected = True
		Unselected = False
			True

Attribute	Description	Range of Values	Default Value
Group Cache Size	This attribute sets the maximum number of Group lookups to cache.	This attribute should be a prime number for best lookup performance	211
Group Cache TTLPositive	This attribute sets the number of seconds to retain the result of a successful lookup.	Integer	60
Group Cache TTLNegative	This attribute sets the number of seconds to retain the result of an unsuccessful lookup.	Integer	10
Group Membership Cache TTL	This attribute sets the number of seconds to store the members of a group before updating it.	Integer	300

Users

Attribute	Description	Range of Values	Default Value

Enable User Cache	This attribute is set true if user cache is enabled.	Boolean,
		Selected = True
		Unselected = False
			True

User Cache Size	This attribute sets the maximum number of User lookups to cache.	This attribute should be a prime number for best lookup performance.	211
User Cache TTLPositive	This attribute sets the number of seconds to retain the result of a successful lookup.	Integer	60
User Cache TTLNegative	This attribute sets the number of seconds to retain the result of an unsuccessful lookup.	Integer	10

Permissions

Attribute	Description	Range of Values	Default Value

Enable Permission Cache	This attribute is set true if permission cache is enabled.	Boolean,
		Selected = True
		Unselected = False
			True

Permission Cache Size	This attribute sets the maximum number of Permission lookups to cache.	This attribute should be a prime number for best lookup performance.	211
Permission Cache TTLPositive	This attribute sets the number of seconds to retain the result of a successful lookup.	Integer	60
Permission Cache TTLNegative	This attribute sets the number of seconds to retain the result of an unsuccessful lookup.	Integer	10

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for user supplied information.	The value must be an alphanumeric string.	Null

Read about Caching Realms in Managing Security in the *Administration Guide.*

6 Startup and Shutdown Class Deployment

By default, this pane allows the user to sort the objects by the following criteria:

- Name
- Type

7 Cluster

The following procedures describe how to use the Administration Console to set the attributes for configuring and managing clusters. For additional information about clusters, please see Configuring WebLogic Servers and Clusters in the *Administration Guide* and *Using WebLogic Server Clusters*.

Configure a Cluster

1. Click the Clusters node in the left pane. The Clusters table displays in the right pane showing all the clusters defined in the domain.
2. Click the Configure a New Cluster link. A dialog displays in the right pane showing the tabs associated with configuring a new cluster.
3. Enter values in the Name, Default Load Algorithm, and Service Age Threshold attribute fields.

Note: Assign a unique name to the cluster. Each configurable resource in your WebLogic environment should have a unique name.

4. Enter a value for the Cluster Address. Supply a cluster address that identifies the Managed Servers in the cluster. The cluster address is used in entity and stateless beans to construct the host name portion of URLs. If the cluster address is not set, EJB handles may not work properly.

   For production systems, this address should be a DNS host name that maps to the individual IP addresses of the Managed Servers participating in the cluster. For development (non-production) systems, this address can be either a DNS host name or a comma-separated list of single address host names or IP addresses. For more information on the cluster address format, see "Cluster Address" in *Using WebLogic Server Clusters*.

5. Click the Create button in the lower right corner to create a cluster with the name you specified in the Name field. The new cluster is added under the Clusters node in the left pane.
6. Click the Multicast tab and change the Send Delay and TTL attribute fields or accept the default values as assigned. The valid range of multicast addresses is from 224.0.0.1 to 239.255.255.255.
7. Click the Apply button in the lower right corner of each tab to save any changes you made.

Clone a Cluster

1. Click the Clusters node in the left pane. The Clusters table displays in the right pane showing all the clusters defined in the domain.
2. Click the Clone icon in the row of the cluster you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a cluster.
3. Enter values in the Name, Cluster Address, Default Load Algorithm, and Service Age Threshold attribute fields
4. Click the Clone button in the lower right corner to create a cluster instance with the name you specified in the Name field. The new instance is added under the Clusters node in the left pane.
5. Click the Multicast tab and change the attribute fields or accept the default values as assigned.
6. Click the Apply button in the lower right corner of each tab to save any changes you made.

Delete a Cluster

1. Click the Clusters node in the left pane. The Clusters table displays in the right pane showing all the clusters defined in the domain.
2. Click the Delete icon in the row of the cluster you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click the Yes button to delete the cluster. The cluster icon under the Clusters node is deleted.

Assign Servers to a Cluster

1. Click the instance node in the left pane under Clusters to select a cluster for server assignment. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Servers tab.
3. Select one or more servers in the Available column that you want to assign to the cluster.
4. Click the mover control to move the servers you selected to the Chosen column.
5. Click the Apply button in the lower right corner to save your assignments.

Monitor Server Participation in a Cluster

1. Click the instance node in the left pane under Clusters to select a cluster for server monitoring. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Monitoring tab.

3. Click the Monitor Server Participation in This Cluster text link. The server table displays in the right pane showing all the servers assigned to this cluster.

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	This attribute reports the name of the cluster.	The cluster name can be up to 256 alphanumeric characters, but may not contain commas or spaces.	Null
Cluster Address	This attribute returns the address to be used by clients to connect to this cluster. This address may be either a DNS host name that maps to multiple IP addresses or a comma separated list of single address host names or IP addresses.	This address may be either a DNS host name that maps to multiple IP addresses or a comma separated list of single address host names or IP addresses.	Null
Default Load Algorithm	The default cluster load algorithm is the algorithm by which the load is distributed. When the last server has been assigned a transaction, the load comes back to the starting server.	Round Robin, Weight-based Round Robin, Random, and Parameter-based Routing	Round Robin

Attribute	Description	Range of Values	Default Value
WebLogic Plug-In Enabled	A value of `true` for a cluster that receives requests from `HttpClusterServlet`. causes `getRemoteAddr` to return the address of the browser client from the proprietary `WL-Proxy-Client-IP` header, instead of the web server. For non-clustered servers that will receive proxied requests, this attribute may be set at the server level, on the Server -->Configuration-->General tab. *MBean:* `weblogic.management.configuration.ClusterMBean` *Attribute:* `WeblogicPluginEnabled` **Note:** This attribute is duplicated in `ServerMBean`. The value in `ClusterMBean` overrides `ServerMBean`.	true false	false

Attribute	**Description**	**Range of Values**	**Default Value**
Service Age Threshold	This attribute describes the number of seconds by which the age of two conflicting services must differ before one is considered older than the other.	Integer (seconds)	180

Multicast

Attribute	Description	Range of Values	Default Value
Multicast Address	The cluster multicast address sets the multicast address used by cluster members to communicate with each other. It must be a well-formed IP address.	The valid range of multicast addresses is from 224.0.0.1 to 239.255.255.255	Null
Multicast Send Delay	This attribute returns the number of milliseconds to delay sending message fragments over multicast in order to avoid OS-level buffer overflow.	Integer	12
Multicast TTL	This attribute returns the number of the multicast ttl.	Integer	1

Servers

Attribute	Description	Range of Values	Default Value
Choose Servers for this Cluster	This attribute allows the user to choose servers to participate in this cluster.	List	Null

Monitoring

Attribute	Description	Range of Values	Default Value
Number of Servers Configured for this Cluster	This attribute returns the number of servers that have been configured to run in this cluster.	Integer	0
Number of Servers Currently Participating in this Cluster	This attribute returns the number of servers running in this cluster.	Integer	0

Notes

Attribut e	Description	Range of Values	Default Value
Notes	This attribute provides a space for optional user supplied information.	The value must be an alphanumeric string.	Null

8 Cluster Runtime

You can monitor the following cluster runtime statisticscs using the Monitoring tab

:

Statistic	Description
Cluster	The name of the cluster(s) in the management domain.
Name	The name of individual server instances in a cluster.
Machine	The optional machine name on which an individual server instance is running.
Servers	The number of active servers in the cluster. *MBean:*`weblogic.management.ClusterRuntimeMBean` *Attribute:* `AliveServersCount`
Resend Requests	The number of state change messages that the current Managed Server resent because a receiving Managed Server missed a message. *MBean:*`weblogic.management.ClusterRuntimeMBean` *Attribute:* `ResendRequestsCount`
Fragments Sent	The total number of multicast fragments that an individual server has broadcast to its cluster.

Statistic	Description
Fragments Received	Total number of multicast message fragments received by the current Managed Server from other Managed Servers in the cluster. *MBean:*`weblogic.management.ClusterRuntimeMBean` *Attribute:* `FragmentsReceivedCount`
Lost Multicast Messages	The total number of multicast messages that this server failed to receive. *MBean:*`weblogic.management.ClusterRuntimeMBean` *Attribute:* `MulticastMessagesLostCount`
Foreign Fragments Dropped Count	The number of dropped multicast message fragments that originated in foreign domains/cluster that use the same multicast address. *MBean:*`weblogic.management.ClusterRuntimeMBean` *Attribute:* `ForeignFragmentsDroppedCount`
Known Servers	The names of the active Managed Servers in the cluster. *MBean:*`weblogic.management.ClusterRuntimeMBean` *Attribute:* ServerNames
Secondary Distributions	The names of the remote Managed Servers for which the current Manager Server is hosting session replicas. The name is appended with a number to indicate the number of secondaries hosted for that Managed Server. *MBean:*`weblogic.management.ClusterRuntimeMBean` *Attribute:* `SecondaryDistributionNames`
Primaries	The number of object that the local server hosts. *MBean:*`weblogic.management.ClusterRuntimeMBean` *Attribute:* `PrimaryCount`

9 Component

By default, this pane allows the user to sort the objects by the following criteria:

- Server
- Machine
- Source info
- Servlets
- Session
- Sessions high
- Total sessions

10 BEA WebLogic J2EE Connector Architecture Attribute Descriptions

ra.xml Attributes

ra.xml Deployment Descriptor Editor Attributes

Attribute	Description	Range of Values	Default Value
Connector Description	This optional element specifies any resource adapter specific requirement for the support of security contract and authentication mechanism.	String	NULL
Connector Display Name	This element contains a short name that is intended to be displayed by tools. The display name need not be unique.	String	NULL

Attribute	Description	Range of Values	Default Value
Small Icon	This element contains the name of a file containing a small (16 x 16) icon image. The file name is a relative path within the resource adapter's rar file. The image may be either in the JPEG or GIF format.	JPEG GIF	
Large Icon	This element contains the name of a file containing a large (32 x 32) icon image. The file name is a relative path within the resource adapter's rar file. The image may be either in the JPEG or GIF format.	JPEG GIF	
Connector Vendor Name	This element specifies the name of resource adapter provider vendor.	String	NULL
Connector Spec Version	This element specifies the version of the connector architecture specification that is supported by this resource adapter. This information enables deployer to configure the resource adapter to support deployment and runtime requirements of the corresponding connector architecture specification.	String	NULL
Connector Eis Type	This element contains information about the type of the EIS. For example, the type of an EIS can be product name of EIS independent of any version info.	String	NULL
Connector Version	This element specifies a string-based version of the resource adapter from the resource adapter provider.	String	NULL
License Description	This element is used to provide text describing the license element. This is an optional element.	String	NULL
License Required	This element specifies whether a license is required to deploy and use the resource adapter.	True False	NULL

Attribute	Description	Range of Values	Default Value
Resourceadapter Managedconnectionfactory Class	This element specifies the fully qualified name of the Java class that implements the javax.resource.spi.Managed-ConnectionFactory interface.	String	NULL
Resourceadapter Connectionfactory Interface	This element specifies the fully-qualified name of the ConnectionFactory interface supported by the resource adapter.	String	NULL
Resourceadapter Connectionfactory Impl Class	This element specifies the fully-qualified name of the ConnectionFactory class that implements resource adapter specific ConnectionFactory interface.	String	NULL
Resourceadapter Connection Interface	This element specifies the fully-qualified name of the Connection interface supported by the resource adapter.	String	NULL
Resourceadapter Connection Impl Class	This element specifies the fully-qualified name of the Connection class that implements resource adapter specific Connection interface.	String	NULL
Transaction Support	This element specifies the level of transaction support provided by the resource adapter.	NoTransaction LocalTransaction XATransaction	NULL
Reauthentication Support	This element specifies whether the resource adapter implementation supports re-authentication of existing Managed-Connection instance.	True False	NULL

ra.xml Config Properties

Attribute	Description	Range of Values	Default Value
Description	This element is used to describe any bounds or well-defined values of properties.	String	NULL
Config Property Name	This element contains the name of a configuration property. The connector architecture defines a set of well-defined properties all of type java.lang.String. A resource adapter provider can extend this property set to include properties specific to the resource adapter and its underlying EIS.	ServerName PortNumber UserName Password ConnectionURL	NULL
Config Property Type	This element contains the fully qualified Java type of a configuration property as required by ManagedConnection-Factory instance.	java.lang.Boolean java.lang.String java.lang.Integer java.lang.Double java.lang.Byte java.lang.Short java.lang.Long java.lang.Float java.lang.Character	NULL
Config Property Value	This element contains the value of a configuration entry.	Various	NULL

ra.xml Authentication Mechanisms

ra.xml Authentication Mechanisms

Attribute	Description	Range of Values	Default Value
Description	This element is used to provide text describing the authentication mechanism type.	String	NULL
Authentication Mechanism Type	This element specifies an authentication mechanism supported by the resource adapter. Note that this support is for the resource adapter and not for the underlying EIS instance. The optional description specifies any resource adapter specific requirement for the support of security contract and authentication mechanism. Note that BasicPassword mechanism type should support the javax.resource.spi.security.PasswordCredential interface. The Kerbv5 mechanism type should support the javax.resource.spi.security.GenericCredential interface.	String	NULL
Credential Interface	This element specifies the interface that the resource adapter implementation supports for the representation of the credentials. This element should be used by application server to find out the Credential interface it should use as part of the security contract.	String	NULL

ra.xml Security Permissions

Attribute	Description	Range of Values	Default Value
Description	This element is used to provide text describing the security permission.	String	NULL
Security Permission Spec	This element specifies a security permission based on the Security policy file syntax.	String	NULL

weblogic-ra.xml Attributes

weblogic-ra.xml Deployment Descriptor Editor Attributes

Attribute	Description	Range of Values	Default Value
Description	This element is used to provide text describing the parent element. The description element should include any information that the deployer wants to describe about the deployed Connection Factory. This is an optional element.	String	NULL

Attribute	Description	Range of Values	Default Value
Connection Factory Name	This element defines that logical name that will be associated with this specific deployment of the Resource Adapter and its corresponding Connection Factory. The value of connection-factory-name can be used in other deployed Resource Adapters via the ra-link-ref element. This will allow multiple deployed Connection Factories to utilize a common deployed Resource Adapter, as well as share configuration specifications. This is a required element.	String	NULL
Jndi Name	This element defines the name that will be used to bind the Connection Factory Object into the Weblogic JNDI Namespace. Client EJBs and Servlets will use this same JNDI in their defined Reference Desciptor elements of the weblogic specific deployment descriptors. This is a required element.	String	NULL
Initial Capacity	This element identifies the initial number of managed connections which the Weblogic Server will attempt to obtain during deployment. This is an optional element. Failure to specify this value will result in Weblogic using its defined default value.	Integer	1
Max Capacity	This element identifies the maximum number of managed connections which the Weblogic Server will allow. Requests for newly allocated managed connections beyond this limit will result in a ResourceAllocationException being returned to the caller. This is an optional element. Failure to specify this value will result in Weblogic using its defined default value.	Integer	10

Attribute	Description	Range of Values	Default Value
Capacity Increment	This element identifies the number of additional managed connections which the Weblogic Server will attempt to obtain during resizing of the maintained connection pool. This is an optional element. Failure to specify this value will result in Weblogic using its defined default value.	Integer	1
Shrinking Enabled	This element indicates whether or not the Connection Pool should have unused Managed Connections reclaimed as a means to control system resources. This is an optional element. Failure to specify this value will result in Weblogic using its defined default value.	True False	True
Shrink Period Minutes	This element identifies the amount of time the Connection Pool Management will wait between attempts to reclaim unused Managed Connections. This is an optional element. Failure to specify this value will result in Weblogic using its defined default value.	Integer	1
Ra Link Ref	This element allows for the logical association of multiple deployed Connection Factories with a single deployed Resource Adapter. The specification of this element with a value identifying a separately deployed Connection Factory will result in this newly deployed Connection Factory sharing the Resource Adapter which had been deployed with the referenced Connection Factory. In addition, any values defined in the referred Connection Factories deployment will be inherited by this newly deployed Connection Factory unless specified. This is an optional element.	String	NULL

Attribute	Description	Range of Values	Default Value
Native Lib Dir	This element identifies the directory location to be used for all native libraries present in this resource adapter deployment. As part of deployment processing, all encountered native libraries will be copied to the location specified. It is the responsibility of the Administrator to perform the necessary platform actions such that these libraries will be found during Weblogic Server runtime. This is a required element IF native libraries are present.	String	NULL
Logging Enabled	This element indicates whether or not the log writer is set for either the ManagedConnectionFactory or ManagedConnection. If this element is set to true, output generated from either the ManagedConnectionFactory or ManagedConnection will be sent to the file specified by the log-filename element. This is an optional element. Failure to specify this value will result in Weblogic using its defined default value.	True False	False
Log Filename	This element specifies the name of the log file which output generated from either the ManagedConnectionFactory or a ManagedConnection are sent. The full address of the filename is required. This is an optional element.	String	NULL

weblogic-ra.xml Config Properties

Attribute	Description	Range of Values	Default Value
Description	This element is used to provide text describing the config properties.	String	NULL
Config Property Name	This element identifies a name that corresponds to an `ra.xml` `config-entry` element with the corresponding `config-property-name`.	String	NULL
Config Property Type	This element identifies a type that corresponds to an `ra.xml` `config-entry` element with the corresponding `config-property-name`.	String	NULL
Config Property Value	This element identifies a value that corresponds to an `ra.xml` `config-entry` element with the corresponding `config-property-name`.		NULL

weblogic-ra.xml Security Principal Map Entries

Attribute	Description	Range of Values	Default Value
Initiating Principal	The security-principal-map element provides a mechanism to define appropriate `resource-principal` values for resource adapter and EIS authorization processing, based upon the known WebLogic run time *`initiating-principal`*. This map allows for the specification of a defined set of initiating principals and the corresponding resource principal's username and password that should be used when allocating managed connections and connection handles.	0 or More	
Resource Username	This element specifies the username identified with the `resource-principal`. Used when allocating managed connections and connection handles.		
Resource Userpassword	This element specifies the password identified with the `resource-principal`. Used when allocating managed connections and connection handles.	String	NULL

11 Connection

Monitor Connections

Attribute	Description	Range of Values	Default
Connections	Physical connections to the backend system that the client uses to interact with.		N/A
Free Connections High Count	Total number of available Connections in the pool at any given time over the history of the connection management service for a particular resource adapter		N/A
Average Active Usage	Average number of active Connections in the pool during the history of the connection management service for a particular resource adapter		N/A
Connections Created Total Count	Total number of Connections created over the history of the connection management service for a particular resource adapter		N/A

Attribute	Description	Range of Values	Default
Connections Matched Total Count	Total number of Connections that have matched a connection request over the history of the connection management service for a particular resource adapter		N/A
Connections Destroyed Total Count	Total number of Connections that have been destroyed from the pool over the history of the connection management service for a particular resource adapter		N/A
Connections Rejected Total Count	Total number of rejected connection requests over the history of the connection management service for a particular resource adapter		N/A
Recycled Total	Total number of Connections recycled to satisfy connection requests over the history of the connection management service for a particular resource adapter		N/A

12 Connection

Deploying Resource Adapters Using the Administration Console

To deploy a resource adapter using the WebLogic Server Administration Console:

1. Start WebLogic Server.
2. Open the Administration Console.
3. Open the Domain you will be working in.
4. Under Deployments, select Connectors in the left panel. The Connector Deployments table displays in the right pane showing all the deployed Connectors (Resource Adapters).
5. Select Configure a new Connector Component...
6. Enter the following information:
   - Name—modify the default name of the connector component as needed.
   - URI Path—enter the full path of the resource adapter `.rar` file or a directory containing the resource adapter exploded directory format. For example: `c:\myaps\components\myResourceAdapter.rar`
   - Deployed—indicate whether the Resource Adapter `.rar` file should be deployed when created.
7. Select the Create button.

Note that the new resource adapter now appears in the Deployments table in the right pane.

Viewing Deployed Resource Adapters Using the Administration Console

To view a deployed resource adapter in the Administration Console:

1. In the Administration Console under Deployments, select Connectors (Resource Adapters) in the left panel.
2. View a list of deployed Connectors in the Connector Deployments table in the right pane.

Undeploying Deployed Resource Adapters Using the Administration Console

To undeploy a deployed resource adapter from the WebLogic Server Administration Console:

1. In the Administration Console under Deployments, select Connectors (Resource Adapters) in the left panel.
2. In the Connector Deployments table, select the connector to undeploy.
3. Under the Configuration tab, de-select the Deployed checkbox.
4. Click Apply.

Undeploying a resource adapter does not remove the resource adapter name from WebLogic Server. The resource adapter remains undeployed for the duration of the Server session, as long as you do not change it once it has been undeployed. You cannot re-use the deployment name with the deploy argument until you reboot the server. You can re-use the deployment name to update the deployment, as described in the following section.

Updating Deployed Resource Adapters Using the Administration Console

When you update the contents of the resource adapter `.rar` file or deployment directory that has been deployed to WebLogic Server, those updates are not reflected in WebLogic Server until:

- You reboot the server (if the `.rar` or directory is to be automatically deployed)

  or

- You update the resource adapter deployment using the WebLogic Server Administration Console

From the WebLogic Server Administration Console:

1. In the Administration Console under Deployments, select Connectors (Resource Adapters) in the left panel.
2. In the Connector Deployments table, select the connector to update.
3. Update the Connector Name and Deployed status as needed.
4. Click Apply.

For additional information, see Programming the WebLogic J2EE Connector Architecture.

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	This attribute provides the name of the application	string	Null
URI			
Path	This attribute sers the full path to the application.		
Deployment Order	This attribute sets the order in which this application is to be deployed		
Deployed	This attribute is used to enable or disable deployment for the application		Not Selected

13 Connector Connection Pool Runtime

By default, this pane allows the user to sort the objects by the following criteria:

- Remote address
- Protocol
- Connect time
- Bytes received
- Bytes sent
- Messages received
- Messsages sent

14 ConsoleLink

Preferences

General

Attribute	Description	Range of Values	Default Value
Autorefresh Every	This attribute sets the time for autorefresh of the console.	Integer, in seconds	10

Remember Last Tab	This attribute enables or disables the ability to return to the last tab selected.	Boolean
		Enabled = selected
		Disabled = not selected
			Selected

Use Navigation Tree	TThis attribute enables or disables the ability to use the navigation tree on the console.	Boolean
		Enabled = selected
		Disabled = not selected
			Selected

15 Conversion

If you have used an earlier version of Weblogic Server, you should convert your weblogic.properties files into a Domain. The resources defined in a single XML configuration file comprise a single domain.

This section describes the followingways to convert your .properties files:

- Conversion Using the Administration Console

Conversion Using the Administration Console

The first step in the conversion process is locating the root directory for your original WebLogic installation. All of the weblogic.properties files you convert must reside under this root, and these files must be located on the same machine that this server is running on.

Use the links to navigate the server's file system to find the weblogic root. When you have found it, click the icon next to it and move on to the next step.

The conversion utility provides a list of entities that have been identified as potential servers and clusters.

To use the Conversion Utility from the WebLogic Server Administration Console:

- Click Convert Weblogic Properties on the right pane.

- Select the Root directory (which is the location of the global weblogic.properties files)by navigating the tree using the icon on the left side of the pane.
- Select the Cluster and Server specific weblogic.properties directories.
- Click Convert

Continue to convert your weblogic.properties files until you have created the domains needed for your application.

SSL Security Files

SSL Security Files are set in the old properties file as

```
weblogic.security.certificate.server=democert.pem
weblogic.security.key.server=demokey.pem
weblogic.security.certificate.authority=ca1024.pem
weblogic.security.clientRootCA=ca.pem
```

and will be copied into the config/<NewDomain>/<ServerName>(Server Specific Directory) where the config.xml file will reflect them as

```
 <Server Name=....

 <SSL
ServerCertificateFileName="./config/<NewDomain>/democert.pem"
ServerKeyFileName="./config/<NewDomain>/demokey.pem"
TrustedCAFileName="./config/<NewDomain>/ca.pem"
ServerCertificateChainFileName="./config/<NewDomain>/ca1024.pem"
....

 >

 </Server>
```

If the SSL security files specified in the weblogic.properties are not in the old server specific directory then they will not be set in the config.xml and will have to be copied into the config/<NewDomain>/<ServerName>(Server Specific Directory), and be set in the config.xml.

Servlets

All servlets registered in the weblogic.properties are converted into a single web application

The conversion tool creates the necessary files like web.xml and weblogic.xml in one of the following directorie

1. `config/<NewDomain>/applications/DefaultWebApp_<ServerName>/WEB-INF` directory and will be made as the default webapp, unless there is already a default web app declared in the properties file.
2. `config/<NewDomain>/<Server_Name>/WEB-INF`, if there is already a default web app declared.

All the Servlet Classes registered individually as `weblogic.httpd.register` in the old weblogic.properties except for the weblogic internal servlets have to be copied under the new weblogic(6.0) server tree structure as specified in the web.xml

For example if web.xml has

```
<servlet>
<servlet-class>weblogic.hello.HelloworldServlet</servlet-class>
</servlet>
```

The servlet class Hellowordlservlet.class should be copied into the corresponding WEB-INF/classes/weblogic/hello directory. The current directory signifies the directory from which the weblogic server(6.0) will be started with the new configuration.

EJB jar files and Web App war files

If the weblogic.properties has weblogic.ejb.deploy and weblogic.httpd.webApp.<webAppName> pointing to a relative directory then those jar and war files have to be copied under the new weblogic(6.0) server tree structure.

Example

weblogic.ejb.deploy=weblogic/ejb/HelloEJB.jar

Then the jar file has to be copied under ./weblogic/ejb/ directory.

The "." indicates the directory from which the server will be started with the new configuration.

16 Customize This View

Table Show

1. To add items to the Table Show list, select them from the Available list and use the arrow to move them to the Chosen list.
2. To remove items from the Table Show list, select them from the Chosen list and use the arrow to move them to the Available list.
3. Click Apply to submit your changes.

Table Sort

1. To add items to the Table Sort list, select them from the Available list and use the arrow to move them to the Chosen list.
2. To remove items from the Table Sort list, select them from the Chosen list and use the arrow to move them to the Available list.
3. Click Apply to submit your changes.

17 Custom Realm

Configuration

Attribute	Description	Range of Values	Default Value
Name	This attribute specifies the name of the Custom Security realm, such as, AccountingRealm .	The name can be up to 256 alphanumeric characters, but may not contain commas or spaces.	Null
Realm Classname	This attribute returns the name of the Java class that contains the Security realm. The Java class needs to be in the CLASSPATH of WebLogic Server.	This attribute can not be changed.	Null
Configuration Data	The information needed to connect to the security store.	key=value list	Null
Password	This attribute sets the password for the security store used with the custom security realm.	String	Null

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for user supplied information.	The value must be an alphanumeric string.	Null

18 Deployment Descriptor Editors

For additional information about the Deployment Descriptor Editors, please see the following sections:

BEA WebLogic J2EE Connector Architecture Attribute Descriptions

EJB Deployment Descriptors

RDBMS Deployment Descriptors

Web Application Deployment Descriptor Editor Help

19 Domain

The following procedures describe how to use the Administration Console to set the attributes for viewing Domain information. Read about Domains in Using Log Messages to Manage WebLogic Servers in the *Administration Guide*.

View Domain Health

1. Click the instance node that represents your Domain (i.e. myDomain where mydomain is the name you specified as the WebLogic Admin Domain Name during installation and myserver is the Server Name you specified during installation) in the left pane. The domain dialog displays in the right pane showing the tabs associated with this domain.
2. Click the Monitoring tab.
3. Click the View Domain Health text link. The domain health view displays in the right pane showing the Servers table and the domain log for this domain.

View Domain Log

1. Right-click the instance node that represents your Domain (i.e. myDomain) in the left pane. This opens a pop-up menu.
2. Click View Domain Log. The log displays in the right pane..

Edit Application Management Settings

1. Click the instance node that represents your Domain (i.e. myDomain) in the left pane. The domain dialog displays in the right pane showing the tabs associated with this domain.
2. Click the Applications tab.
3. Edit the value in the Auto Update Interval attribute field.
4. Click Apply to save any changes you made.

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	This attribute returns the name of this configuration.	The name can be up to 256 alphanumeric characters, but may not contain commas or spaces.	mydomain
Console Enabled	This attribute is used to enable or disable the console.		Selected
Console Context Path	This attribute is used to set the context path for the WebLogic Server console.		console

JTA

Attribute	Description	Range of Values	Default Value
TimeoutSeconds	The time, in seconds, a transaction may be active before the system forces a rollback.	Integer	30 seconds
AbandonTimeoutSeconds	The maximum time, in seconds, that a transaction coordinator will persist in attempting to complete a transaction.	Integer	86400 seconds
BeforeCompletionIterationLimit	The number of `beforeCompletion` callbacks that will be processed before the system forces a rollback.	Integer	10
MaxTransactions	The maximum number of transactions that may be active on a particular server at one time.	Integer	10000
MaxUniqueNameStatistics	The maximum number of unique transaction names that may be tracked by a server at one time.	Integer	1000
ForgetHeuristics	A Boolean value specifying whether the transaction manager should instruct a resource to forget any transaction that is has a heuristic outcome.	True, not selected False, selected	Not selected

SNMP

Attribute	Description	Range of Values	Default Value

Enabled	The SNMP Service provides the SNMP agent functionality and is a part of the Administration Server. The agent monitors all WebLogic resources for the domain. The Admnistration Server must be restarted for configuration changes to take effect.	True = enabled
		False = disabled
			False

SNMP Port	This is the port on which the WebLogic SNMP agent listens for incoming requests from SNMP managers.	Valid listen port	161
Mib Data Refresh Interval	The SNMP agent maintains a cache of all the attribute values and responds to manager requests by obtaining the attribute value from this cache. The MIB Data Refresh Interval is the interval, in seconds, at which the SNMP agent does a complete refresh of the cache. When it does a refresh, it does a GET on all the WebLogic attributes represented in the WebLogic SNMP MIB.	Integer, in seconds	120

Attribute	Description	Range of Values	Default Value
Server Status Check Interval Factor	The SNMP agent multiplies this number times the MIB Data Refresh Interval to determine how frequently it should check to determine whether Managed Servers in the domain are up or down. The agent obtains this value from the MIB cache. If the Server Status Check Interval Factor is 1, the WebLogic SNMP agent checks whether Managed Servers are up or down at the interval defined in MIB Data Refresh Interval.	Integer, >0	Obtained from MIB Cache
Community Prefix	This string is used to form the SNMP Community Name which functions as a textual password for communication with SNMP managers. If the community prefix sent by the SNMP manager does not match the value configured in this attribute, the SNMP agent will return an authenticationFailuretrap to the requestor.	String If it is of the form `community_prefix@server_name`, the agent will return data only for the specified Managed Server. If it is of the form `community_prefix@domain_name`, the agent will return data for every server in the domain. If the SNMP manager sends only `community_prefix`, the agent will only retrieve data for the Administration Server.	Null

Attribute	Description	Range of Values	Default Value
Debug Level	This attribute sets the level of debugging messages sent to the administrator. If the value is set to 0, no debug messages are generated. If the value is greater than 0, messages are generated that describe what the agent code is doing. The higher the value, the more detailed the messages.	Integer between 0 and 3	0

Logging

Attribute	Description	Range of Values	Default Value

File Name	This attribute returns the name of the log file.	String
		To include a time or date stamp in the file name when the log file is rotated, add `java.text.SimpleDateFormat` variables. Surround each variable with percentage (%) characters.
		For example, if you enter the following value in the File Name field:
		`wl-domain_%yyyy%_%MM%_%dd%_%hh%_%mm%.log`
		the domain log file will be named `wl-domain_yyyy_MM_dd_hh_mm.log`
		When the Administration Server instance rotates the domain log file, the rotated file name contains the date stamp. For example, if the domain log file is rotated on 2 April, 2003 at 10:05 AM, the log file that contains the old log messages will be named:
		`wl-domain_2003_04_02_10_05.log`
			config/mydomain/logs/wl-domain.log

Attribute	Description	Range of Values	Default Value
Rotation Type	Criteria for moving old log messages to a separate file. After the server renames a file, subsequent messages accumulate in a new file with the name that you specified in the File Name field.	■ `None`. Messages accumulate in a single file. You must erase the contents of the file when the size is unwieldy. ■ `by Size`. When the log file reaches the size that you specify in `File Min Size`, the server renames the file. ■ `by Time`. At each time interval that you specify in `File Time Span`, the server renames the file.	`None`
File Min Size	This attribute sets the threshold at which a new log file is created. Relevant only if `Rotation Type` is `by Size`.	Integer	50
Rotation Time	Determines the start time for a time-based rotation sequence. At the time that this value specifies, the server renames the current log file. Thereafter, the server renames the log file at an interval that you specify in `File Time Span`. Relevant only if `Rotation Type` is `by Time`.	Use the following `java.text.SimpleDateFormat` format to specify a date and time: `k:mm`. For information about this format, refer to the J2EE Javadoc. If the date that you specify has already past, the rotation starts immediately.	00:00

Attribute	Description	Range of Values	Default Value
File Time Span	This attribute sets the threshold at which a new log file is created. Relevant only if `Rotation Type` is `by Time`.	Integer	24

Number of Files Limited	Indicates whether a server will limit the number of log files that it creates when it rotates the log. The limit is based on the value in `File Count`.	Boolean
		True = selected
		False = not selected
			Not Selected

File Count	The maximum number of log files that the server creates when it rotates the log. Only valid if `Number of Files Limited` is `true` and `Rotation Type` is either `by Size` or `by Time`.	Integer	7

Applications

Attribute	Description	Range of Values	Default Value
Auto Update Interval	This attribute sets the interval for the automatic updates of applications.	Integer	3000

Security

General

Attribute	Description	Range of Values	Default Value
Audit Provider Class	This attribute allows you to specify an Audit Provider.	The name of a valid Audit Provider class	Null

Guest Disabled	This attribute disables the default user Guest.	Boolean
		True = selected
		False = not selected
			Selected

Filerealm

Attribute	Description	Range of Values	Default Value
Caching Realm	This attribute allows you to select the caching realm for which you want to set or review security.	List	None Selected
Max Users	The maximum number of Users supported by the File realm.	The maximum value is 10,000. The minimum value is 1.The maximum is not enforced, rather, you are warned when you reach the maximum	1000
Max Groups	The maximum number of Groups supported by the File realm.	The maximum value is 10,000. The minimum value is 1. The maximum is not enforced, rather, you are warned when you reach the maximum.	1000
Max ACLs	The maximum number of ACLs supported by the File realm.	The maximum value is 10,000. The minimum value is 1.The maximum is not enforced, rather, you are warned when you reach the maximum.	1000

Passwords

Attribute	Description	Range of Values	Default Value
Minimum Password Length	This attribute sets the number of characters required in a password. Passwords must contain a minimum of 8 characters.	Greater than or equal to 8 characters	8

Lockout Enabled	This attribute is selected to lock user accounts when invalid login attempts occur.	Boolean
		Enabled = selected
		Disabled = not selected
			Selected

Attribute	Description	Range of Values	Default Value
Lockout Threshold	The number of failed logins for a user that can be tried before that account is locked. Any subsequent attempts to access the account (even if the username/password combination is correct) raise a Security exception; the account remains locked until it is explicitly unlocked by the system administrator or another login attempt is made after the lockout duration period ends. Note that invalid login attempts must be made within a span defined by the `Lockout Reset Duration` attribute to count toward the value of the `LockoutThreshold` attribute.	1 to 99,999	5
Lockout Duration	This attribute specifies the number of minutes that a user's account will be locked after the number of invalid login attempts occurs within the time specified by the `Lockout Reset Duration` attribute.	0 to 999999 minutes	30

Attribute	Description	Range of Values	Default Value
Reset Duration	This attribute specifies the number of minutes within which invalid login attempts must occur in order for the user's account to be locked. An account is locked if the number of invalid login attempts defined in the `Lockout Threshold` field happens within the amount of time defined by this field.	Integer	5 Minutes
Lockout Cache Size	This attribute is used to set the intended cache size of unused and invalid login attempts.	0 to 10	5

Advanced

Attribute	Description	Range of Values	Default Value
Connection Filter	This attribute allows you to select the connection filter for your security realm.	List	None Selected
Results Batch Size			
Log All Checks			

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for user supplied information.	The value must be an alphanumeric string.	Null

Read about Domains in Using Log Messages to Manage WebLogic Servers in the *Administration Guide*.

20 Domain Log Filter

The following procedures describe how to use the Administration Console to set the attributes for configuring and managing Domain Log Filters. Read about Domain Log Filters in Using Log Messages to Manage WebLogic Servers in the *Administration Guide*.

Create a New Domain Log Filter

1. Click the Domain Log Filters node in the left pane. The Domain Log Filters table displays in the right pane showing all the log filters in the domain.
2. Click the Create a New Domain Log Filter text link. A dialog displays in the right pane showing the tabs associated with configuring a new domain log filter.
3. Enter values in the Name and Severity Level attribute fields.
4. Click Create to create a domain log filter instance with the name you specified in the Name field. The new instance is added under the Domain Log Filters node in the left pane.
5. Click the Servers, Subsystems, and Users tabs individually and change the attribute fields or accept the default values as assigned.
6. Click Apply to save any changes you made.

Clone a Domain Log Filter

1. Click the Domain Log Filters node in the left pane. The Domain Log Filters table displays in the right pane showing all the log filters in the domain.
2. Click the Clone icon in the row of the domain log filter you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a domain log filter.
3. Enter values in the Name and Severity Level attribute fields.
4. Click Create to create a domain log filter instance with the name you specified in the Name field. The new instance is added under the Domain Log Filters node in the left pane.
5. Click the Servers, Subsystems, and Users tabs individually and change the attribute fields or accept the default values as assigned.
6. Click Apply to save any changes you made.

Delete a Domain Log Filter

1. Click the Domain Log Filters node in the left pane. The Domain Log Filters table displays in the right pane showing all the log filters in the domain.
2. Click the Delete icon in the row of the domain log filter you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click Yes to delete the domain log filter. The domain log filter icon under the Domain Log Filters node is deleted.

Configuration

Criteria

Attribute	Description	Range of Values	Default Value
Name	This attribute sets the name of the Domain log filter.	This attribute must be an alphanumeric string of no more than 256 characters and cannot contain commas.	MyDomain Log Filter
Severity Level	This attribute sets the severity level to be reported in the Domain log filter	List	Error

Servers

Attribute	Description	Range of Values	Default Value
Server List	This attribute provides a list of available servers from which the Domain log filter will receive messages.	List	Null

Subsystems

Attribute	Description	Range of Values	Default Value
Subsystem Names	List of available subsystems from which the Domain log filter will receive messages	List	Null

Users

Attribute	Description	Range of Values	Default Value
User IDs	This attribute allows the user to select user IDs from the list of available IDs.	List	Null

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for optional user supplied information.	The value must be an alphanumeric string.	Null

Read about Domain Log Filters in Using Log Messages to Manage WebLogic Servers in the *Administration Guide*.

21 EJB Component

The following sections describe how to use the Administration Console to set the attributes for installing or configuring new EJBs. Read about EJBs in *Programming WebLogic Enterprise Java Beans*.

Install a New EJB

1. Expand the Deployments node in the left pane.
2. Double-click EJB. The EJB Deployments table displays in the right pane showing all the deployed EJBs.
3. Click Install a new EJB to install a new EJB on the server. The Upload and Install and Application dialog shows in the right pane.
4. Enter the path of the . jar file in the text-entry field, or click the Browse button to browse your file system and choose the .jar file you want to install.
5. Click Upload to install the .jar file. The new EJB is added under EJB in the left pane.

Configure a New EJB

1. Expand the Deployments node in the left pane.

2. Double-click EJB. The EJB Deployments table displays in the right pane showing all the deployed EJBs.
3. Click Configure a new EJB to create a new component. The EJBComponent pane, where you configure the new EJB displays in the left pane.
4. On the General tab in the EJBComponent pane:
    a. Enter the name of the EJB component in the Name field.
    b. Enter the path you want to use for this component and the Uniform Resource Identifier (URI) in the Path URI field.
    c. Enter the order in which you want to deploy the component to the server in the Deployment Order field.
    d. Click the Deployed checkbox to set the deployment status for the component.
5. To configure the compiler options, on the EJBC options tab in the EJBComponent pane:.
    a. Enter the Java compiler to be used in the Java Compiler field.
    b. Enter the path where the generated files will be stored in the Tmp Path field.
    c. Enter any rmic options in the Extra Rmic Options field.
    d. Click the Keep Generated Source Files checkbox to choose to keep the generated source files.
6. Click Create to create the component.

General

Attribute	Description	Range of Values	Default Value
Name	This attribute returns the name of the EJB component.	The name can be up to 256 alphanumeric characters, but may not contain commas or spaces.	Null
URL	This attribute returns the URL of this EJB component.	Valid URL	Null
Application	This attribute returns the name of the application.	String	Null
Number of Instances	This attribute returns the number of instances of this EJB component.	Integer	Null
Java Compiler	This attribute sets the java compiler to be used for this component.	Valid java compiler	javac
Tmp Path	This attribute sets the path where the generated files are stored by `ejbc`.	String	Null
Extra RMIC Options	This attribute allows the user to set rmic options.	String	Null
Keep Generated Sources	This attribute enables or disables the ability to keep generated source files.	Boolean Selected = enabled Not Selected = not enabled	Not Selected

Attribute	Description	Range of Values	Default Value
Path URI	This attribute allows the user to set the path for this component and the Uniform Resource Identifier (URI) that points to the application component. The path can be a directory name for a component in exploded directory format or an EAR file.	String	Null

Deployed	This attribute allows the user to set the deployment status for a component.	Boolean
		Deployed = selected
		Undeployed = not selected
			Selected

Deployment Order	This attribute allows the user to set the order in which the component is deployed to the server.		Null

Target

Attribute	Description	Range of Values	Default Value
Target Servers	This attribute allows the user to select servers on which this ejb will be deployed.	List	None Selected
Target Clusters	This attribute allows the user to select clusters on which this ejb will be deployed.	List	None Selected

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for user supplied information.	The value must be an alphanumeric string.	Null

Read about EJBs in *Programming WebLogic Enterprise Java Beans*.

22 EJB Component Runtime

By default, this pane allows the user to sort the objects by the following criteria:

- Application
- Name
- URI

23 EJB Home

By default, this pane allows the user to sort the objects by the following criteria:

- Application
- Name
- URI

CHAPTER

24 EJB Deployment Descriptors

weblogic-ejb-jar.xml deployment descriptors

This section describes the 6.1 WebLogic Server-specific XML elements found in `weblogic600-ejb-jar.xml` that are used to define the deployment descriptors in the WebLogic Server EJB container. These elements map to fields, of approximately the same name, in the WebLogic Server Administration Console Use these deployment descriptors for 2.0 EJBs.

allow-concurrent-calls

Range of values:	`true | false`
Default value:	`false`
Requirements:	Requires the server to throw a `RemoteException` when a stateful session bean instance is currently handling a method call and another (concurrent) method call arrives on the server.
Parent elements:	`weblogic-enterprise-bean` `stateful-session-descriptor`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `allow-concurrent-calls` element specifies whether a stateful session bean instance will allow concurrent method calls. By default, `allows-concurrent-calls` is `false`. However, when this value is set to `true`, the EJB container blocks the concurrent method call and allows it to proceed when the previous call has completed.

cache-type

Range of values:	`NRU | LRU`
Default value:	`NRU`
Requirements:	
Parent elements:	`weblogic-enterprise-bean` `stateful-session-cache`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The element sets the cache type for the EJB.

connection-factory-jndi-name

Range of values:	
Default value:	`weblogic.jms.MessageDrivenBeanConnectionFactory` in `config.xml`
Requirements:	Requires the server to throw a `RemoteException` when a stateful session bean instance is currently handling a method call and another (concurrent) method call arrives on the server.
Parent elements:	`weblogic-enterprise-bean` `message-driven-descriptor`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `connection-factory-jndi-name` element specifies the JNDI name of the JMS ConnectionFactory that the MessageDriven Bean should look up to create its queues and topics. If this element is not specified, the default will be the `weblogic.jms.MessageDrivenBeanConnectionFactory` in `config.xml`.

concurrency-strategy

Range of values:	`Exclusive | Database | ReadOnly`
Default value:	`Database`
Requirements:	Optional element. Valid only for entity EJBs.

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`entity-cache`

Deployment file:	`weblogic-ejb-jar.xml`

Function

This element specifies how the container should manage concurrent access to an entity bean. You can set this element to one of three possible values:

- `Exclusive` is the default locking behavior for WebLogic Server versions 3.1 through 5.1. WebLogic Server places an exclusive lock on cached entity EJB instances when the bean is associated with a transaction. Other requests for the EJB instance block until the transaction completes.

- `Database` causes WebLogic Server to defer locking requests for an entity EJB to the underlying datastore. With the `Database` concurrency strategy, WebLogic Server does not cache the intermediate results of entity EJBs involved in a transaction.

- `ReadOnly` designates an entity EJB that is never modified. WebLogic Server calls `ejbLoad()` for `ReadOnly` beans based on the `read-timeout-seconds` parameter.

db-is-shared

Range of values:	`true | false`
Default value:	`true`
Requirements:	Optional element. Valid only for entity EJBs.

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`persistence`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `db-is-shared` element applies only to entity beans. When set to `true` WebLogic Server assumes that EJB data could be modified between transactions and reloads the data at the beginning of each transaction. When set to `false` WebLogic Server assumes that it has exclusive access to the EJB data in the persistent store.

delay-updates-until-end-of-tx

Range of values:	`true | false`
Default value:	`true`
Requirements:	Optional element. Valid only for entity EJBs.

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`persistence`

Deployment file:	`weblogic-ejb-jar.xml`

Function

Set this element to `true` (the default) to update the persistent store of all beans in a transaction at the completion of the transaction. This generally improves performance by avoiding unnecessary updates. However, it does not preserve the ordering of database updates within a database transaction.

If your datastore uses an isolation level of `TRANSACTION_READ_UNCOMMITTED`, you may want to allow other database users to view the intermediate results of in-progress transactions. In this case, set `delay-updates-until-end-of-tx` to `false` to update the bean's persistent store at the conclusion of each method invoke.

Note: Setting `delay-updates-until-end-of-tx` to false does not cause database updates to be "committed" to the database after each method invoke; they are only sent to the database. Updates are committed or rolled back in the database only at the conclusion of the transaction.

description

Range of values:	n/a
Default value:	n/a
Requirements:	n/a

Parent elements:	`weblogic-enterprise-bean,`
	`transaction-isolation`
	`method`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `description` element is used to provide text that describes the parent element.

destination-jndi-name

Range of values:	Valid JNDI name
Default value:	n/a
Requirements:	Required in `message-driven-descriptor`.
Parent elements:	`weblogic-enterprise-bean` `message-driven-descriptor`
Deployment file:	`weblogic-ejb-jar.xml`

Function

`The destination-jndi-name` element specifies the JNDI name used to associate a message-driven bean with an actual JMS Queue or Topic deployed in the in WebLogic Server JNDI tree.

ejb-name

Range of values:	Name of an EJB defined in `ejb-jar.xml`
Default value:	n/a
Requirements:	Required element in `method` stanza The name must conform to the lexical rules for an NMTOKEN.

Parent elements:	`weblogic-enterprise-bean`
	`    transaction-isolation`
	`        method`

Deployment file:	`weblogic-ejb-jar.xml`

Function

`ejb-name` specifies the name of an EJB to which WebLogic Server applies isolation level properties. This name is assigned by the `ejb-jar` file's deployment descriptor. The name must be unique among the names of the enterprise beans in the same `ejb.jar` file. The enterprise bean code does not depend on the name; therefore the name can be changed during the application-assembly process without breaking the enterprise bean's function. There is no architected relationship between the ejb-name in the deployment descriptor and the JNDI name that the deployer will assign to the enterprise bean's home.

ejb-reference-description

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	Optional element.
Parent elements:	`weblogic-enterprise-bean` `reference-description`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `resource-description` stanza maps a resource reference defined in `ejb-jar.xml` to the JNDI name of an actual resource available in WebLogic Server.

- `ejb-ref-name` specifies a resource reference name. This is the reference that the EJB provider places within the `ejb-jar.xml` deployment file.
- `jndi-name` specifies the JNDI name of an actual resource factory available in WebLogic Server.

ejb-ref-name

Range of values:	n/a
Default value:	n/a
Requirements:	Optional element.

Parent elements:	`weblogic-enterprise-bean`
	`reference-description`
	`ejb-reference-description`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `resource-description` stanza maps a resource reference named in `ejb-jar.xml` to the JNDI name of an actual resource available in WebLogic Server.

ejb-local-reference-description

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	Optional element.
Parent elements:	`weblogic-enterprise-bean` `reference-descriptor`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The ejb-local-reference-description element is used to map the JNDI name in the WebLogic Server of an EJB that is referenced by the bean in an ejb-local ref.

enable-call-by-reference

Range of values:	`true | false`
Default value:	`true`
Requirements:	Optional element.

Parent elements:	`weblogic-enterprise-bean`
	`reference-description`
	`ejb-reference-description`

Deployment file:	`weblogic-ejb-jar.xml`

Function

By default, EJB methods called from within the same server pass arguments by reference. This increases the performance of method invocation because parameters are not copied.

If you set `enable-call-by-reference` to `False` parameters to EJB methods are copied (pass-by-value) in accordance with the EJB 1.1 specification. Pass by value is always necessary when the EJB is called remotely (not from within the server).

entity-cache

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	The `entity-cache` stanza is optional, and is valid only for entity EJBs.

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `entity-cache` element defines the following options used to cache entity EJB instances within WebLogic Server:

- max-beans-in-cache
- idle-timeout-seconds
- read-timeout-seconds
- concurrency-strategy

entity-clustering

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	Optional element. Valid only for entity EJBs in a cluster.

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `entity-clustering` element uses the following options to specify how an entity bean will be replicated in a WebLogic cluster:

- home-is-clusterable
- home-load-algorithm
- home call-router-class-name

entity-descriptor

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	One `entity-descriptor` stanza is required for each entity EJB in the *.jar*.
Parent elements:	`weblogic-enterprise-bean`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `entity-descriptor` element specifies the following deployment parameters that are applicable to an entity bean:

- pool
- entity-cache
- lifecycle
- persistence
- entity-clustsering

finders-load-bean

Range of values:	true | false
Default value:	`true`
Requirements:	Optional element. Valid only for CMP entity EJBs.

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`persistence`

Deployment file:	`weblogic-ejb-jar.xml`

Function

`finders-load-bean` determines whether WebLogic Server loads the EJB into the cache after a call to a finder method returns a reference to the bean. If you set this element to `true`, WebLogic immediately loads the bean into the cache if a reference to a bean is returned by the finder. If you set this element to `false`, WebLogic Server does not automatically load the bean into the cache until the first method invocation; this behavior is consistent with the EJB 1.1 specification.

home-call-router-class-name

Range of values:	Valid router class name
Default value:	null
Requirements:	Optional element. Valid only for entity EJBs and stateful session EJBs in a cluster.

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`entity-clustering`
	`and`
	`weblogic-enterprise-bean`
	`stateful-session-descriptor`
	`stateful-session-clustering`

Deployment file:	`weblogic-ejb-jar.xml`

Function

`home-call-router-class-name` specifies the name of a custom class to use for routing bean method calls. This class must implement `weblogic.rmi.extensions.CallRouter()`. If specified, an instance of this class is called before each method call. The router class has the opportunity to choose a server to route to based on the method parameters. The class returns either a server name or null, which indicates that the current load algorithm should select the server.

home-is-clusterable

Range of values:	`true | false`
Default value:	`true`
Requirements:	Optional element. Valid only for entity EJBs and stateful session EJBs in a cluster.

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`entity-clustering`
	`and`
	`weblogic-enterprise-bean`
	`stateful-session-descriptor`
	`stateful-session-clustering`

Deployment file:	`weblogic-ejb-jar.xml`

Function

When `home-is-clusterable` is `true`, the EJB can be deployed from multiple WebLogic Servers in a cluster. Calls to the home stub are load-balanced between the servers on which this bean is deployed, and if a server hosting the bean is unreachable, the call automatically fails over to another server hosting the bean.

home-load-algorithm

Range of values:	`round-robin | random | weight-based`
Default value:	Value of `weblogic.cluster.defaultLoadAlgorithm`
Requirements:	Optional element. Valid only for entity EJBs and stateful session EJBs in a cluster.

Parent elements:	`weblogic-enterprise-bean,`
	`     entity-descriptor,`
	`          entity-clustering`
	`and`
	`weblogic-enterprise-bean`
	`     stateful-session-descriptor`
	`          stateful-session-clustering`

Deployment file:	`weblogic-ejb-jar.xml`

Function

`home-load-algorithm` specifies the algorithm to use for load balancing between replicas of the EJB home. If this property is not defined, WebLogic Server uses the algorithm specified by the server property, `weblogic.cluster.defaultLoadAlgorithm`.

You can define `home-load-algorithm` as one of the following values:

- `round-robin`: Load balancing is performed in a sequential fashion among the servers hosting the bean.
- `random`: Replicas of the EJB home are deployed randomly among the servers hosting the bean.
- `weight-based`: Replicas of the EJB home are deployed on host servers according to the servers’ current workload.

idle-timeout-seconds

Range of values:	`1` to *`maxSeconds`*, where *`maxSeconds`* is the maximum value of an int.
Default value:	`600`
Requirements:	Optional element

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`entity-cache`
	`and`
	`weblogic-enterprise-bean,`
	`stateful-session-descriptor,`
	`stateful-session-cache`

Deployment file:	`weblogic-ejb-jar.xml`

Function

`idle-timeout-seconds` defines the maximum length of time a stateful EJB should remain in the cache. After this time has elapsed, WebLogic Server removes the bean instance if the number of beans in cache approaches the limit of `max-beans-in-cache`. The removed bean instances are passivated.

initial-beans-in-free-pool

Range of values:	`0` to *`maxBeans`*
Default value:	`0`
Requirements:	Optional element. Valid only for stateless session EJBs.

Parent elements:	`weblogic-enterprise-bean,`
	`pool`

Deployment file:	`weblogic-ejb-jar.xml`

Function

If you specify a value for `initial-beans-in-free-pool`, you set the initial size of the pool. WebLogic Server populates the free pool with the specified number of bean instances for every bean class at startup. Populating the free pool in this way improves initial response time for the EJB, because initial requests for the bean can be satisfied without generating a new instance.

initial-context-factory

Range of values:	`true | false`
Default value:	`weblogic.jndi.WLInitialContextFactory`
Requirements:	Requires the server to throw a `RemoteException` when a stateful session bean instance is currently handling a method call and another (concurrent) method call arrives on the server.
Parent elements:	`weblogic-enterprise-bean` `message-driven-descriptor`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `initial-context-factory` element specifies the contextFactory that the container will use to create its connection factories. If initial-context-factory is not specified, the default will be `weblogic.jndi.WLInitialContextFactory`.

is-modified-method-name

Range of values:	Valid entity EJB method name
Default value:	None
Requirements:	Optional element. Valid only for entity EJBs.

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`persistence`

Deployment file:	`weblogic-ejb-jar.xml`

Function

`is-modified-method-name` specifies a method that WebLogic Server calls when the EJB is stored. The specified method must return a `boolean` value. If no method is specified, WebLogic Server always assumes that the EJB has been modified and always saves it.

Providing a method and setting it as appropriate can improve performance for EJB 1.1-compliant beans, and for beans that use bean-managed persistence. However, any errors in the method's return value can cause data inconsistency problems.

Note: `isModified()` is no longer required for 2.0 CMP entity EJBs based on the EJB 2.0 specification However, it still applies to BMP and 1.1 CMP EJBs. When you deploy EJB 2.0 entity beans with container-managed persistence, WebLogic Server automatically detects which EJB fields have been modified, and writes only those fields to the underlying datastore.

isolation-level

Range of values:	`Serializable	ReadCommitted	ReadUncommitted	RepeatableRead`
Default value:	n/a
Requirements:	Optional element.

Parent elements:	`weblogic-enterprise-bean,`
	`transaction-isolation`

Deployment file:	`weblogic-ejb-jar.xml`

Function

`isolation-level` specifies the isolation level for all of the EJB’s database operations. The following are possible values for `isolation-level`:

- `TRANSACTION_READ_UNCOMMITTED`: The transaction can view uncommitted updates from other transactions.
- `TRANSACTION_READ_COMMITTED`: The transaction can view only committed updates from other transactions.
- `TRANSACTION_REPEATABLE_READ`: Once the transaction reads a subset of data, repeated reads of the same data return the same values, even if other transactions have subsequently modified the data.
- `TRANSACTION_SERIALIZABLE`: Simultaneously executing this transaction multiple times has the same effect as executing the transaction multiple times in a serial fashion.

Refer to your database documentation for more information on the implications and support for different isolation levels.

jndi-name

Range of values:	Valid JNDI name
Default value:	n/a
Requirements:	Required in `resource-description` and `ejb-reference-description`.

Parent elements:	`weblogic-enterprise-bean`
	`and`
	`weblogic-enterprise-bean`
	`     reference-description`
	`          resource-description`
	`and`
	`weblogic-enterprise-bean`
	`     reference-description`
	`          ejb-reference-description`

Deployment file:	`weblogic-ejb-jar.xml`

Function

`jndi-name` specifies the JNDI name of an actual EJB, resource, or reference available in WebLogic Server.

local-jndi-name

Range of values:	Valid JNDI name
Default value:	n/a
Requirements:	Required if the bean has a local home.
Parent elements:	`weblogic-enterprise-bean`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `local-jndi-name` element specifies a jndi-name for a bean's local home. If a bean has both a remote and a local home, then it must have two JNDI names; one for each home.

lifecycle

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	The `lifecycle` stanza is optional.

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor`
	`and`
	`weblogic-enterprise-bean`
	`stateful-session-descriptor`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `lifecycle` element defines options that affect the lifecycle of stateful and entity EJB instances within WebLogic Server. Currently, the `lifecycle` element includes only one element: `passivation-strategy`.

max-beans-in-cache

Range of values:	`1` to *`maxBeans`*
Default value:	`100`
Requirements:	Optional element

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`entity-cache`
	`and`
	`weblogic-enterprise-bean`
	`stateful-session-descriptor`
	`stateful-session-cache`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `max-beans-in-cache` element specifies the maximum number of objects of this class that are allowed in memory. When `max-bean-in-cache` is reached, WebLogic Server passivates some EJBs that have not been recently used by a client. `max-beans-in-cache` also affects when EJBs are removed from the WebLogic Server cache.

max-beans-in-free-pool

Range of values:	`0` to *`maxBeans`*
Default value:	*`1000`*
Requirements:	Optional element. V

Parent elements:	`weblogic-enterprise-bean,`
	`stateless-session-descriptor,`
	`pool`
	`weblogic-enterprise-bean,`
	`message-driven-descriptor,`
	`pool`
	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`pool`

Deployment file:	`weblogic-ejb-jar.xml`

Function

WebLogic Server maintains a free pool of EJBs for every stateless session bean, message-driven, and entity bean class. The `max-beans-in-free-pool` element defines the size of this pool. For more information, see "EJB Lifecycle" in *Programming WebLogic Server EJBs*

message-driven-descriptor

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	
Parent elements:	`weblogic-enterprise-bean`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `message-driven-descriptor` element associates a message-driven bean with a JMS destination in WebLogic Server. This element specifies the following deployment parameters:

- pool
- destination-jndi-name
- initial-context-factory
- provider-url
- connection-factory-jndi-name

method

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	Optional element. You can specify more than one `method` stanza to configure multiple EJB methods.
Parent elements:	`weblogic-enterprise-bean` `transaction-isolation`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `method` element defines a method of an enterprise ben's home or remote interface, or a set of methods.

method-intf

Range of values:	`Home	Remote`
Default value:	n/a
Requirements:	Optional element.

Parent elements:	`weblogic-enterprise-bean`
	`transaction-isolation`
	`method`

Deployment file:	`weblogic-ejb-jar.xml`

Function

`method-intf` specifies the EJB interface to which WebLogic Server applies isolation level properties. Use this element only if you need to differentiate between methods having the same signature in the EJB's home and remote interface.

method-name

Range of values:	Name of an EJB defined in `ejb-jar.xml | *`
Default value:	n/a
Requirements:	Required element in `method` stanza.

Parent elements:	`weblogic-enterprise-bean`
	`transaction-isolation`
	`method`

Deployment file:	`weblogic-ejb-jar.xml`

Function

`method-name` specifies the name of an individual EJB method to which WebLogic Server applies isolation level properties. Use the asterisk (*) to specify all methods in the EJB's home and remote interfaces.

If you specify a `method-name`, the method must be available in the specified `ejb-name`.

method-param

Range of values:	Fully qualified Java type of a method parameter
Default value:	n/a
Requirements:	Required element in `method-params`.

Parent elements:	`weblogic-enterprise-bean`
	`transaction-isolation`
	`method`
	`method-params`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `method-param` element specifies the fully-qualified Java type name of a method parameter.

method-params

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	Optional stanza.

Parent elements:	`weblogic-enterprise-bean`
	`transaction-isolation`
	`method`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `method-params` stanza contains one or more elements that define the Java type name of each of the method's parameters.

passivation-strategy

Range of values:	`default | transaction`
Default value:	`default`
Requirements:	Optional element. Valid only for entity EJBs.

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`lifecycle`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `passivation-strategy` element determines whether or not WebLogic Server maintains the intermediate state of entity EJBs in its cache.

persistence

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	Required only for entity EJBs that use container-managed persistence services.

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `persistence` element defines the following options that determine the persistence type, transaction commit behavior, and `ejbLoad()` and `ejbStore()` behavior for entity EJBs in WebLogic Server:

- is-modified-method-name

- delay-updates-until-end-of-tx
- finders-load-bean
- persistence-type
- db-is-shared
- persistence-use

persistence-type

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	Required only for entity EJBs that use container-managed persistence services.

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`persistence`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `persistence-type` element defines a persistence service that the entity EJB can use. You can define multiple `persistence-type` stanzas in `weblogic-ejb-jar.xml` for testing your EJB with multiple persistence services. Only the persistence type defined in `persistence-use` is actually used during deployment.

`persistence-type` includes several elements that identify the persistence types:

- type-identifier
- type-version
- type-storage

persistence-use

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	Required only for entity EJBs that use container-managed persistence services.

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`persistence`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `persistence-use` element is similar to `persistence-type`, but it defines the persistence service actually used during deployment. `persistence-use` uses the `type-identifier` and `type-version` elements defined in a `persistence-type` to identify the service.

persistent-store-dir

Range of values:	Fully qualified filesystem path
Default value:	n/a
Requirements:	Optional element.
Parent elements:	`weblogic-enterprise-bean` `stateful-session-descriptor`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `persistent-store-dir` element specifies a file system directory where WebLogic Server stores the state of passivated stateful session bean instances.

pool

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	Optional element.
Parent elements:	`weblogic-enterprise-bean` `stateless-session-descriptor`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `pool` element configures the behavior of the WebLogic Server free pool for a stateless session and message driven EJBs. The options are:

- max-beans-in-free-pool
- initial-beans-in-free-pool

principal-name

Range of values:	valid WebLogic Server principal name
Default value:	n/a
Requirements:	At least one `principal-name` is required in the `security-role-assignment` stanza. You may define more than one `principal-name` for each `role-name`.
Parent elements:	`weblogic-enterprise-bean` `security-role-assignment`
Deployment file:	`weblogic-ejb-jar.xml`

Function

`principal-name` specifies the name of an actual WebLogic Server principal to apply to the specified `role-name`.

provider-url

Range of values:	valid name
Default value:	n/a
Requirements:	Used in conjunction with `initial-context-factory` and `connection-factory-jndi-name`.
Parent elements:	`weblogic-enterprise-bean` `message-driven-descriptor`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `provider-url` element specifies the URL provider to be used by the InitialContext. Typically, this is the host : port.

read-timeout-seconds

Range of values:	`0` to *`maxSeconds`*, where *`maxSeconds`* is the maximum value of an int.
Default value:	`600`
Requirements:	Optional element. Valid only for entity EJBs.

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`entity-cache`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `read-timeout-seconds` element specifies the number of seconds between `ejbLoad()` calls on a `Read-Only` entity bean. By default, `read-timeout-seconds` is set to 0, and WebLogic Server calls `ejbLoad()` only when the bean is brought into the cache.

reference-descriptor

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	Optional element.
Parent elements:	`weblogic-enterprise-bean`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `reference-descriptor` element maps references in the `ejb-jar.xml` file to the JNDI names of actual resource factories and EJBs available in WebLogic Server.

replication-type

Range of values:	InMemory | None
Default value:	None
Requirements:	Optional element. Valid only for stateful session EJBs in a cluster.

Parent elements:	`weblogic-enterprise-bean`
	`    stateful-session-descriptor`
	`        stateful-session-clustering`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `replication-type` element determines whether or not WebLogic Server replicates the state of stateful session EJBs across WebLogic Server instances in a cluster. If you select `InMemory`, the state of the EJB is replicated. If you select `None`, the state is not replicated.

res-env-ref-name

Range of values:	A valid resource environment reference name from the `ejb-jar.xml` file
Default value:	n/a
Requirements:	n/a

Parent elements:	`weblogic-enterprise-bean`
	`reference-descriptor`
	`resource-env-description`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `res-env-ref-name` element specifies the name of a resource environment reference.

res-ref-name

Range of values:	A valid resource reference name from the `ejb-jar.xml` file
Default value:	n/a
Requirements:	Required element if the EJB specifies resource references in `ejb-jar.xml`

Parent elements:	`weblogic-enterprise-bean`
	`reference-description`
	`resource-description`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `res-ref-name` element specifies the name of a resourcefactory reference. This is the reference that the EJB provider places within the `ejb-jar.xml` deployment file.

resource-env-description

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	Optional element.
Parent elements:	`weblogic-enterprise-bean` `reference-descriptor`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `resource-env-description` element maps a resource environment reference defined in `ejb-jar.xml` to the JNDI name of an actual resource available in WebLogic Server.

resource-description

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	Optional element.
Parent elements:	`weblogic-enterprise-bean` `reference-description`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `resource-description` element maps a resource reference defined in `ejb-jar.xml` to the JNDI name of an actual resource available in WebLogic Server.

role-name

Range of values:	An EJB role name defined in `ejb-jar.xml`
Default value:	n/a
Requirements:	Required element in security-role-assignment.
Parent elements:	`weblogic-enterprise-bean` `security-role-assignment`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `role-name` element identifies an application role name that the EJB provider placed in the `ejb-jar.xml` deployment file. Subsequent `principal-name` elements in the stanza map WebLogic Server principals to the specified `role-name`.

run-as-identity-principal

Range of values:	Principal that will be used as the identity as defined in `ejb-jar.xml`
Default value:	n/a
Requirements:	Required element in `security-role-assignment`.
Parent elements:	weblogic-enterprise-bean
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `run-as-identity-principal` element specifies the principal to be used as the identity for beans that have a `security-identity.run-as-specified-identity` set in the `ejb-jar.xml`.

The principal named in this element must be one of the principals mapped to the `run-as-specified--identity` role.

security-role-assignment

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	Required element if `ejb-jar.xml` defines application roles.
Parent elements:	`weblogic-ejb-jar`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `security-role-assignment` stanza maps application roles in the `ejb-jar.xml` file to the names of security principals available in WebLogic Server.

stateful-session-cache

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	The `stateful-session-cache` stanza is optional, and is valid only for stateful session EJBs.

Parent elements:	`weblogic-enterprise-bean,`
	`stateful-session-descriptor`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `stateful-session-cache` element defines the following options used to cache stateful session EJB instances within WebLogic Server.

- home-is-clusterable
- home-load-algorithm
- home-call-router-class-name
- replication-type

stateful-session-clustering

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	Optional element. Valid only for stateful session EJBs in a cluster.

Parent elements:	`weblogic-enterprise-bean,`
	`stateful-session-descriptor`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `stateful-session-clustering` stanza element specifies the following options that determine how WebLogic Server replicates stateful session EJB instances in a cluster:

- max-beans-in-cache
- idle-timeout-seconds
- cache-type

stateful-session-descriptor

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	One `stateful-session-descriptor` stanza is required for each stateful session EJB in the `.jar`.
Parent elements:	`weblogic-enterprise-bean`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `stateful-session-descriptor` element specifies the following deployment parameters that are applicable for stateful session EJBs in WebLogic Server:

- stateful-session-cache
- lifecycle
- persistent-store-dir
- stateful-session-clustering
- allow-concurrent-calls

stateless-bean-call-router-class-name

Range of values:	Valid router class name
Default value:	n/a
Requirements:	Optional element. Valid only for stateless session EJBs in a cluster.

Parent elements:	`weblogic-enterprise-bean,`
	`    stateless-session-descriptor`
	`        stateless-clustering`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `stateless-bean-call-router-class-name` element specifies the name of a custom class to use for routing bean method calls. This class must implement `weblogic.rmi.extensions.CallRouter()`. If specified, an instance of this class is called before each method call. The router class has the opportunity to choose a server to route to based on the method parameters. The class returns either a server name or null, which indicates that the current load algorithm should select the server.

stateless-bean-is-clusterable

Range of values:	`true | false`
Default value:	`true`
Requirements:	Optional element. Valid only for stateless session EJBs in a cluster.

Parent elements:	`weblogic-enterprise-bean,`
	`stateless-session-descriptor`
	`stateless-clustering`

Deployment file:	`weblogic-ejb-jar.xml`

Function

When `stateless-bean-is-clusterable` is `true`, the EJB can be deployed from multiple WebLogic Servers in a cluster. Calls to the home stub are load-balanced between the servers on which this bean is deployed, and if a server hosting the bean is unreachable, the call automatically fails over to another server hosting the bean.

stateless-bean-load-algorithm

Range of values:	`round-robin	random	weight-based`
Default value:	Value of `weblogic.cluster.defaultLoadAlgorithm`
Requirements:	Optional element. Valid only for stateless session EJBs in a cluster.

Parent elements:	`weblogic-enterprise-bean,`
	`stateless-session-descriptor`
	`stateless-clustering`

Deployment file:	`weblogic-ejb-jar.xml`

Function

`stateless-bean-load-algorithm` specifies the algorithm to use for load balancing between replicas of the EJB home. If this property is not defined, WebLogic Server uses the algorithm specified by the server property, `weblogic.cluster.defaultLoadAlgorithm`.

You can define `stateless-bean-load-algorithm` as one of the following values:

- `round-robin`: Load balancing is performed in a sequential fashion among the servers hosting the bean.
- `random`: Replicas of the EJB home are deployed randomly among the servers hosting the bean.
- `weight-based`: Replicas of the EJB home are deployed on host servers according to the servers’ current workload.

stateless-bean-methods-are-idempotent

Range of values:	`true | false`
Default value:	`false`
Requirements:	Optional element. Valid only for stateless session EJBs in a cluster.

Parent elements:	`weblogic-enterprise-bean,`
	`stateless-session-descriptor`
	`stateless-clustering`

Deployment file:	`weblogic-ejb-jar.xml`

Function

You can set this element to either `true` or `false`. Set `stateless-bean-methods-are-idempotent` to "true" only if the bean is written such that repeated calls to the same method with the same arguments has exactly the same effect as a single call. This allows the failover handler to retry a failed call without knowing whether the call actually completed on the failed server. Setting this property to `true` makes it possible for the bean stub to recover automatically from any failure as long as another server hosting the bean can be reached.

stateless-clustering

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	Optional element. Valid only for stateless session EJBs in a cluster.

Parent elements:	`weblogic-enterprise-bean,`
	`stateless-session-descriptor`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `stateless-clustering` element specifies the following options that determine how WebLogic Server replicates stateless session EJB instances in a cluster:

- stateless-bean-is-clusterable
- stateless-bean-load-algorithm
- stateless-bean-call-router-class-name
- stateless-bean-methods-are-idempotent

stateless-session-descriptor

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	One `stateless-session-descriptor` element is required for each stateless session EJB in the *.jar*.
Parent elements:	`weblogic-enterprise-bean`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `stateless-session-descriptor` element defines deployment parameters, such as caching, clustering, and persistence for stateless session EJBs in WebLogic Server.

transaction-descriptor

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	Optional element.
Parent elements:	`weblogic-enterprise-bean`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `transaction-descriptor` element specifies options that define transaction behavior in WebLogic Server. Currently, this stanza includes only one element: `trans-timeout-seconds`.

transaction-isolation

Range of values:	n/a (XML stanza)
Default value:	n/a (XML stanza)
Requirements:	Optional element.
Parent elements:	`transaction-description`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `transaction-isolation` element defines method-level transaction isolation settings for an EJB.

trans-timeout-seconds

Range of values:	`0` to *`maxSeconds`*
Default value:	`30`
Requirements:	Optional element. Valid only for all EJBs.

Parent elements:	`weblogic-enterprise-bean,`
	`transaction-descriptor`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `trans-timeout-seconds` element specifies the maximum duration for an EJB's container-initiated transactions. If a transaction lasts longer than `trans-timeout-seconds`, WebLogic Server rolls back the transaction.

If you specify no value for `trans-timeout-seconds`, container-initiated transactions timeout after five minutes, by default.

type-identifier

Range of values:	Valid string
Default value:	n/a
Requirements:	Required only for entity EJBs that use container-managed persistence services.

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`persistence`
	`persistence-type`
	`and`
	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`persistence`
	`persistence-use`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `type-identifier` element contains text that identifies an entity EJB persistence type. WebLogic Server RDBMS-based persistence uses the identifier, `WebLogic_CMP_RDBMS`. If you use a different persistence vendor, consult the vendor's documentation for information on the correct `type-identifier`.

type-storage

Range of values:	Valid string
Default value:	n/a
Requirements:	Required only for entity EJBs that use container-managed persistence services.

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`persistence`
	`persistence-type`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `type-storage` element defines the full path of the file that stores data for this persistence type. The path must specify the file's location relative to the top level of the EJB's `.jar` deployment file or deployment directory.

WebLogic Server RDBMS-based persistence generally uses an XML file named `weblogic-cmp-rdbms-jar.xml` to store persistence data for a bean. This file is stored in the `META-INF` subdirectory of the `.jar` file.

type-version

Range of values:	Valid string
Default value:	n/a
Requirements:	Required only for entity EJBs that use container-managed persistence services.

Parent elements:	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`persistence`
	`persistence-type`
	`and`
	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`persistence`
	`persistence-use`

Deployment file:	`weblogic-ejb-jar.xml`

Function

The `type-version` element identifies the version of the specified persistence type.

Note: If you use WebLogic Server RDBMS-based persistence, the specified version must *exactly* match the RDBMS persistence version for the WebLogic Server release. Specifying an incorrect version results in the following error:

```
weblogic.ejb.persistence.PersistenceSetupException: Error
initializing the CMP Persistence Type for your bean: No installed
Persistence Type matches the signature of (identifier
‘Weblogic_CMP_RDBMS’, version ‘version_number’).
```

weblogic-ejb-jar

Range of values:	
Default value:	
Requirements:	
Parent elements:	`weblogic-enterprise-bean,`
Deployment file:	`weblogic-ejb-jar.xml`

Function

`weblogic-ejb-jar` is the root element of the weblogic component of the EJB deployment descriptor.

weblogic-enterprise-bean

Range of values:	
Default value:	
Requirements:	
Parent elements:	`weblogic-enterprise-bean,`
Deployment file:	`weblogic-ejb-jar.xml`

Function

The `weblogic-enterprise-bean` element contains the deployment information for a bean that is available in WebLogic Server.

This section describes the 6.1 WebLogic Server-specific XML elements found in `weblogic510-ejb-jar.xml` that are used to define the deployment descriptors in the WebLogic Server EJB container. These elements map to fields, of approximately the same name, in the WebLogic Server Administration Console. Use these deployment descriptors for 1.1 EJBs.

caching-descriptor

The `caching-descriptor` stanza affects the number of EJBs in the WebLogic Server cache as well as the length of time before EJBs are passivated or pooled. The entire stanza, as well as each of its elements, is optional. WebLogic Server uses default values where no elements are defined.

max-beans-in-free-pool

Note: This element is valid only for stateless session EJBs.

WebLogic Server maintains a free pool of EJBs for every bean class. This optional element defines the size of the pool. By default, `max-beans-in-free-pool` has no limit; the maximum number of beans in the free pool is limited only by the available memory.

initial-beans-in-free-pool

Note: This element is valid only for stateless session EJBs.

If you specify a value for `initial-bean-in-free-pool`, WebLogic Server populates the free pool with the specified number of bean instances at startup. Populating the free pool in this way improves initial response time for the EJB, since initial requests for the bean can be satisfied without generating a new instance.

`initial-bean-in-free-pool` defaults to 0 if the element is not defined.

max-beans-in-cache

Note: This element is valid only for stateful session EJBs and entity EJBs.

This element specifies the maximum number of objects of this class that are allowed in memory. When max-bean-in-cache is reached, WebLogic Server passivates some EJBs that have not been recently used by a client. max-beans-in-cache also affects when EJBs are removed from the WebLogic Server cache.

The default value of max-beans-in-cache is 100.

idle-timeout-seconds

idle-timeout-seconds defines the maximum length of time a stateful EJB should remain in the cache. After this time has elapsed, WebLogic Server may remove the bean instance if the number of beans in cache approaches the limit of max-beans-in-cache.

idle-timeout-seconds defaults to 600 if you do not define the element.

cache-strategy

The cache-strategy element can be one of the following:

- Read-Write
- Read-Only

The default value is Read-Write.

read-timeout-seconds

The read-timeout-seconds element specifies the number of seconds between ejbLoad() calls on a Read-Only entity bean. By default, read-timeout-seconds is set to 600 seconds. If you set this value to 0, WebLogic Server calls ejbLoad only when the bean is brought into the cache.

persistence-descriptor

The persistence-descriptor stanza specifies persistence options for entity EJBs.

is-modified-method-name

is-modified-method-name specifies a method that WebLogic Server calls when the EJB is stored. The specified method must return a boolean value. If no method is specified, WebLogic Server always assumes that the EJB has been modified and always saves it.

Providing a method and setting it as appropriate can improve performance. However, any errors in the method's return value can cause data inconsistency problems.

delay-updates-until-end-of-tx

Set this property to true (the default), to update the persistent store of all beans in a transaction at the completion of the transaction. This generally improves performance by avoiding unnecessary updates. However, it does not preserve the ordering of database updates within a database transaction.

If your datastore uses an isolation level of TRANSACTION_READ_UNCOMMITTED, you may want to allow other database users to view the intermediate results of in-progress transactions. In this case, set delay-updates-until-end-of-tx to false to update the bean's persistent store at the conclusion of each method invoke.

Note: Setting delay-updates-until-end-of-tx to false does not cause database updates to be "committed" to the database after each method invoke; they are only sent to the database. Updates are committed or rolled back in the database only at the conclusion of the transaction.

persistence-type

A persistence-type defines a persistence service that can be used by an EJB. You can define multiple persistence-type entries in weblogic-ejb-jar.xml for testing with multiple persistence services. Only the persistence type defined in persistence-use is used during deployment.

persistence-type includes several elements that define the properties of a service:

- type-identifier contains text that identifies the specified persistence type. For example, WebLogic Server RDBMS persistence uses the identifier, WebLogic_CMP_RDBMS.
- type-version identifies the version of the specified persistence type.

Note: The specified version must *exactly* match the RDBMS persistence version for the WebLogic Server release. Specifying an incorrect version results in the error:

```
weblogic.ejb.persistence.PersistenceSetupException: Error
initializing the CMP Persistence Type for your bean: No installed
Persistence Type matches the signature of (identifier
‘Weblogic_CMP_RDBMS’, version ‘version_number’).
```

- `type-storage` defines the full path of the file that stores data for this persistence type. The path must specify the file’s location relative to the top level of the EJB’s *.jar* deployment file or deployment directory.

  WebLogic Server RDBMS-based persistence generally uses an XML file named `weblogic-cmp-rdbms-jar.xml` to store persistence data for a bean. This file is stored in the `META-INF` subdirectory of the *.jar* file.

db-is-shared

The `db-is-shared` element applies only to entity beans. When set to `true` (the default value), WebLogic Server assumes that EJB data could be modified between transactions and reloads data at the beginning of each transaction. When set to `false`, WebLogic Server assumes that it has exclusive access to the EJB data in the persistent store.

stateful-session-persistent-store-dir

`stateful-session-persistent-store-dir` specifies the file system directory where WebLogic Server stores the state of passivated stateful session bean instances.

persistence-use

The `persistence-use` property is similar to `persistence type`, but it defines the persistence service actually used during deployment. `persistence-use` uses the `type-identifier` and `type-version` elements defined in a `persistence-type` to identify the service.

clustering-descriptor

The `clustering-descriptor` stanza defines the replication properties and behavior for EJBs deployed in a WebLogic Server cluster. The `clustering-descriptor` stanza and each of its elements are optional, and are not applicable to single-server systems.

home-is-clusterable

You can set this element to either `true` or `false`. When `home-is-clusterable` is true, the EJB can be deployed from multiple WebLogic Servers in a cluster. Calls to the home stub are load-balanced between the servers on which this bean is deployed, and if a server hosting the bean is unreachable, the call automatically fails over to another server hosting the bean.

home-load-algorithm

`home-load-algorithm` specifies the algorithm to use for load balancing between replicas of the EJB home. If this property is not defined, WebLogic Server uses the algorithm specified by the server property, `weblogic.cluster.defaultLoadAlgorithm`.

You can define `home-load-algorithm` as one of the following values:

- `round-robin`: Load balancing is performed in a sequential fashion among the servers hosting the bean.
- `random`: Replicas of the EJB home are deployed randomly among the servers hosting the bean.
- `weight-based`: Replicas of the EJB home are deployed on host servers according to the servers' current workload.

home-call-router-class-name

`home-call-router-class-name` specifies the custom class to use for routing bean method calls. This class must implement `weblogic.rmi.extensions.CallRouter()`. If specified, an instance of this class is called before each method call. The router class has the opportunity to choose a server to route to based on the method parameters. The class returns either a server name or null, which indicates that the current load algorithm should select the server.

stateless-bean-is-clusterable

This property is similar to `home-is-clusterable`, but it is applicable only to stateless session EJBs.

stateless-bean-load-algorithm

This property is similar to `home-load-algorithm`, but it is applicable only to stateless session EJBs.

stateless-bean-call-router-class-name

This property is similar to `home-call-router-class-name`, but it is applicable only to stateless session EJBs.

stateless-bean-methods-are-idempotent

You can set this element to either `true` or `false`. Set `stateless-bean-methods-are-idempotent` to `true` only if the bean is written such that repeated calls to the same method with the same arguments has exactly the same effect as a single call. This allows the failover handler to retry a failed call without knowing whether the call actually completed on the failed server. Setting this property to `true` makes it possible for the bean stub to automatically recover from any failure as long as another server hosting the bean can be reached.

Note: This property is applicable only to stateless session EJBs.

transaction-descriptor

The `transaction-descriptor` stanza contains elements that define transaction behavior in WebLogic Server. Currently, this stanza includes only one element:

trans-timeout-seconds

The `trans-timeout-seconds` element specifies the maximum duration for the EJB's container-initiated transactions. If a transaction lasts longer than `trans-timeout-seconds`, WebLogic Server rolls back the transaction.

If you specify no value for `trans-timeout-seconds`, container-initiated transactions timeout after five minutes, by default.

reference-descriptor

The `reference-descriptor` stanza maps references in the `ejb-jar.xml` file to the JNDI names of actual resource factories and EJBs available in WebLogic Server.

The `reference-descriptor` stanza contains one or more additional stanzas to define resource factory references and EJB references.

resource-description

The following elements define an individual `resource-description`:

- `res-ref-name` specifies a resource reference name. This is the reference that the EJB provider places within the `ejb-jar.xml` deployment file.
- `jndi-name` specifies the JNDI name of an actual resource factory available in WebLogic Server.

ejb-reference-description

The following elements define an individual `ejb-reference-description`:

- `res-ref-name` specifies an EJB reference name. This is the reference that the EJB provider places within the `ejb-jar.xml` deployment file.
- `jndi-name` specifies the JNDI name of an actual EJB available in WebLogic Server.

transaction-isolation

The `transaction-isolation` stanza specifies the transaction isolation level for EJB methods. The stanza consists of one or more `isolation-level` elements that apply to a range of EJB methods.

isolation-level

`isolation-level` defines a valid transaction isolation level to apply to specific EJB methods. The following are possible values for `isolation-level`:

- `TRANSACTION_READ_UNCOMMITTED`: The transaction can view uncommitted updates from other transactions.
- `TRANSACTION_READ_COMMITTED`: The transaction can view only committed updates from other transactions.
- `TRANSACTION_REPEATABLE_READ`: Once the transaction reads a subset of data, repeated reads of the same data return the same values, even if other transactions have subsequently modified the data.
- `TRANSACTION_SERIALIZABLE`: Simultaneously executing this transaction multiple times has the same effect as executing the transaction multiple times in a serial fashion.

Refer to your database documentation for more information on the implications and support for different isolation levels.

method

The `method` stanza defines the EJB methods to which an isolation level applies. `method` defines a range of methods using the following elements:

- `description` is an optional element that describes the method.
- `ejb-name` identifies the EJB to which WebLogic Server applies isolation level properties.
- `method-intf` is an optional element that indicates whether the specified method(s) reside in the EJB’s home or remote interface. The value of this

element must be “Home” or “Remote”. If you do not specify `method-intf`, you can apply an isolation to methods in both interfaces.

- `method-name` specifies either the name of an EJB method or an asterisk (*) to designate all EJB methods.
- `method-params` is an optional stanza that lists the Java types of each of the method’s parameters. The type of each parameter must be listed in order, using individual `method-param` elements within the `method-params` stanza.

security-role-assignment

The `security-role-assignment` stanza maps application roles in the `ejb-jar.xml` file to the names of security principals available in WebLogic Server.-

`security-role-assignment` can contain one or more pairs of the following elements:

- `role-name` is the application role name that the EJB provider placed in the `ejb-jar.xml` deployment file.
- `principal-name` specifies the name of an actual WebLogic Server principal.

enable-call-by-reference

By default, EJB methods called from within the same server pass arguments by reference. This increases the performance of method invocation since parameters are not copied.

If you set `enable-call-by-reference` to `false`, parameters to EJB methods are copied (pass by value) in accordance with the EJB 1.1 specification. Pass by value is always necessary when the EJB is called remotely (not from within the server).

25 EJB Home Runtime

By default, this pane allows the user to sort the objects by the following criteria:

- Application
- Name
- URI

26 Entity EJB Runtime

You can monitor your entity EJBs using the following statistics:

Statistic	Description
Idle Beans Count	Reports the number of idle beans in the free pool that are available for use.
Beans In Use Count	Reports the number of beans currently in use from the free pool. This information is useful for tracking demand for your EJB. For example, this can be important when investigating an abnormal pool miss ratio. The `beans-in-use` number may slightly exceed the `beans-in-cache` number in certain situations. WebLogic Server uses anonymous instances to execute finder and home methods. These anonymous instances would cause the `beans-in-use` count to go up while not changing the `beans-in-cache` count. However, the difference between these numbers should be relatively small.
Waiter Total Count	Reports the number of times a thread requested and had to wait for a bean from the free pool.
Timeout Total Count	Reports the total number of transactions that have timed out. Every EJB request uses valuable server resources such as threads and bean instances. A timed out transaction means that server resources were tied up in vain, suggesting a problem with the application.
Cached Beans Current Count	Reports the total number of beans from this EJB Home currently in the EJB cache. Use this information to calculate the current percentage of the configured cache capacity being used.

Statistic	Description
Cache Access Count	Reports the total number of attempts to access a bean from the cache. This information is useful for giving context to other counts such as cache hits.
Cache Hit Count	Reports the total number of times an attempt to access a bean from the cache succeeded. This information is useful for determining the effectiveness of the EJB cache.
Activation Count	Reports the total number of beans from this EJB Home that have been activated.
Passivation Count	Reports the total number of beans from this EJB Home that have been passivated.
Lock Entries Current Count	Reports the current number of lock entries in the lock manager. This information may be helpful in detecting stale lock entries.
Lock Manager Access Count	Reports the total number of attempts to obtain a lock on a bean. This includes attempts to obtain a lock on a bean that is already locked on behalf of the client. This information is useful for giving context to the waiter and timeout total counts.

27 Execute Queue

Create a New Execute Queue

1. Select the server for which the Execute Queue will be created from the list of servers available in the left pane of the Administration Console.
2. Click Monitoring.
3. On the General Tab, click Monitor All Active Queues.
4. Click Configure Execute Queues.
5. Click Configure a new Execute Queue.
6. On the Configuration Tab, accept the default values or edit them as desired.
7. Click Create.

General

Attribute	Description	Range of Values	Default Value
Name	This attribute sets the name of the executequeue.		
Thread Priority	This attribute sets the priority for the threads in this queue.		
Thread Count	This attribute sets the maximum number of threads for this queue.		

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute allows optional user-supplied information.	string	Null

28 Execute Queue Runtime

By default, this pane allows the user to sort the objects by the following criteria:

- Name
- Idle execue threads
- Oldest pending request
- Requests serviced
- Pending requests

29 Execute Threads

By default, this pane allows the user to sort the objects by the following criteria:

- Queue
- Name
- User
- Transaction
- Total requests

30 File T3

These procedures describe how to use the Administration Console to set the attributes for configuring and managing FileT3s. For more information, see Configuring WebLogic Servers and Clusters in the Administration Guide.

Configure a New FileT3

1. Click the FileT3 node in the left pane. The FileT3 table displays in the right pane showing all the FileT3s defined in the domain.
2. Click the Configure a New FileT3 text link. A dialog displays in the right pane showing the tabs associated with configuring a new FileT3.
3. Enter values in the Name and Path attribute fields.
4. Click Create to create a FileT3 instance with the name you specified in the Name field. The new instance is added under the FileT3 node in the left pane.
5. Click Apply to save changes.

Clone a FileT3

1. Click the FileT3 node in the left pane. The FileT3 table displays in the right pane showing all the FileT3s defined in the domain.

2. Click the Clone icon in the row of the FileT3 you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a FileT3.
3. Enter values in the Name and Path attribute fields.
4. Click Clone to create a FileT3 instance with the name you specified in the Name field. The new instance is added under the FileT3 node in the left pane.
5. Click Apply to save changes.

Delete a FileT3

1. Click the FileT3 node in the left pane. The FileT3 table displays in the right pane showing all the FileT3s defined in the domain.
2. Click the Delete icon in the row of the FileT3 you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click Yes to delete the FileT3. The FileT3 icon under the FileT3 node is deleted.

Assign a FileT3

1. Click the instance node in the left pane under FileT3 for the file you want to assign. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Targets tab.
3. Complete the following steps for the Servers and Clusters tabs:
   a. Select one or more targets in the Available column that you want to assign to the FileT3.
   b. Click the mover control to move the targets you selected to the Chosen column.
   c. Click Apply to save your assignments.

Configuration

Attribute	Description	Range of Values	Default Value
Name	This attribute returns the name of the File T3 component.	The name can be up to 256 alphanumeric characters, but may not contain commas or spaces.	MyFileT3
Path	This attribute allows the user to set the path to the File T3 component.	Valid path	Null

Target

Servers

Attribute	Description	Range of Values	Default Value
Target Servers	This attribute allows the user to select the servers to be used as targets for the File T3 component.	List	Null

Clusters

Attribute	Description	Range of Values	Default Value
Target Clusters	This attribute allows the user to select the clusters to be used as targets for the File T3 component.	List	Null

Notes

Attribut e	Description	Range of Values	Default Value
Notes	This attribute provides a space for optional user-supplied information.	The value must be an alphanumeric string.	Null

31 Group

The following procedures describe how to use the Administration Console to set the attributes for creating and managing Groups. Read about Groups in Managing Security in the *Administration Guide.*

Create a New Group

1. Click the Groups node in the left pane. The Groups table displays in the right pane showing all the groups in the domain.
2. Enter a value in the New Group Name attribute field.
3. Click Create to create the group. A dialog displays in the right pane showing additional controls for working with the new group.
4. Enter values in the Add Users and Add Groups attribute fields to add users and groups respectively. Separate multiple users and multiple groups in each field with a space. Note that any users and groups you add must already exist in the database.
5. Click Update Group. The dialog in the right pane refreshes showing the new users and groups in the Members field.

Delete a Group

1. Click the Groups node in the left pane. The Groups table displays in the right pane showing all the groups in the domain.
2. Click the Delete icon in the row of the group you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click Yes to delete the group. The group icon under the Groups node is deleted.

Remove Users from a Group

1. Click the Groups node in the left pane. The Groups table displays in the right pane showing all the groups in the domain.
2. Click the name of the group that has users you want to remove. A dialog displays in the right pane showing additional controls for working with the selected group.
3. Enter the names of the users you want to remove in the Remove Users attribute field.
4. Click Update Group. The dialog in the right pane refreshes showing that the users you selected have been removed from the Members field.

Remove Groups from a Group

1. Click the Groups node in the left pane. The Groups table displays in the right pane showing all the groups in the domain.
2. Click the name of the group that has groups you want to remove. A dialog displays in the right pane showing additional controls for working with the selected group.

3. Enter the names of the groups you want to remove in the Remove Groups attribute field.
4. Click Update Group. The dialog in the right pane refreshes showing that the groups you selected have been removed from the Members field.

Configuration

Attribute	Description	Range of Values	Default Value
Name	This attribute returns the name of the group.	The name can be up to 256 alphanumeric characters, but may not contain commas or spaces.	Null
Members	This attribute returns the members in the group.		Null

Notes

Attribut e	Description	Range of Values	Default Value
Notes	This attribute provides a space for user supplied information.	The value must be an alphanumeric string.	Null

Read about Groups in Managing Security in the *Administration Guide.*

32 JDBC Connection Pool

These procedures describe how to use the Administration Console to set the attributes for configuring and managing connection pools, which are an essential part of database connectivity.

For more information, see

- Managing JDBC Connectivity in the *Administration Guide*, and
- *Programming WebLogic JDBC*, the developers' guide.

Configure a JDBC Connection Pool

1. Click to expand the JDBC node.
2. Click the Connection Pools node. The Connection Pool table displays in the right pane showing all the connection pools defined in your domain.
3. Click the Configure a New JDBC Connection Pool text link. A dialog displays in the right pane showing the tabs associated with configuring a new connection pool.
4. Enter values in the Name, URL, Driver Classname, Properties, Password, and Open String Password attribute fields.
5. Click Create to create a connection pool instance with the name you specified in the Name field. The new instance is added under the Connection Pools node in the left pane.
6. Click the Connections and Testing tabs and change the attribute fields or accept the default values as assigned.

7. Click Apply to save any changes you made.

For what’s next in setting connectivity, see the Configuration Procedures in Managing JDBC Connectivity.

Clone a JDBC Connection Pool

1. Click to expand the JDBC node.
2. Click the Connection Pools node. The Connection Pool table displays in the right pane showing all the connection pools defined in your domain.
3. Click the Clone icon in the row of the connection pool you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a connection pool.
4. Enter values in the Name, URL, Driver Classname, Properties, and Password attribute fields.
5. Click Create to create a connection pool instance with the name you specified in the Name field. The new instance is added under the Connection Pools node in the left pane.
6. Click the Connections and Testing tabs and change the attribute fields or accept the default values as assigned.
7. Click Apply to save any changes you made.

Delete a JDBC Connection Pool

1. Click to expand the JDBC node.
2. Click the Connection Pools node. The Connection Pool table displays in the right pane showing all the connection pools defined in your domain.

3. Click the Delete icon in the row of the connection pool you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
4. Click Yes to delete the connection pool. The connection pool icon under the Connection Pools node is deleted.

Assign a JDBC Connection Pool to One or More Servers or Clusters

1. Click the instance node in the left pane under Connection Pools for the pool you want to assign. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Targets tab.
3. Complete the following steps for the Servers or Clusters tabs:
   a. Select one or more targets in the Available column to which you want to assign the connection pool.
   b. Click the mover control to move the targets you selected to the Chosen column.
   c. Click Apply to save your assignments.

Configuration

These tables describe the attributes you set in the Administration Console to configure and manage connection pools, which are an essential part of database connectivity. For the procedures you need to follow to set connectivity, see the Configuration Procedures in Managing JDBC Connectivity and *Programming WebLogic JDBC*, the developers' guide.

General

Attribute	Description	Range of Values	Default Value
Name	This attribute returns the name of the connection pool.	The name can be up to 256 alphanumeric characters, but may not contain commas or spaces.	Null
URL	The URL for the database, as specified in the JDBC driver documentation. The URL is passed to the driver to create the physical database connections.		
Driver Class Name	The string containing the name of the JDBC driver class. This is the full package name of the JDBC 2-tier driver class used to create the physical connections between the WebLogic Server and the DMBS for this connection pool. It must be the name of a c lass that implements the `java.sql.Driver` interface. Check the documentation for the JDBC driver to find the full path name.		
Properties	The list of the properties passed to this JDBC driver.		

Attribute	Description	Range of Values	Default Value
Password	This value overrides any password defined in Properties (as a name/value pair). This attribute is passed to the 2-tier JDBC driver when creating physical database connections. The value is stored in an encrypted form in the config.xml and can be used to avoid storing cleartext passwords in that file.		Null
Open String Password	If set, this value overrides the password in the open string. This password is used in the open string for creating an XA physical database connection. The value is stored in an encrypted form in the config.xml.		

Connections

Attribute	Description	Range of Values	Default Value
Login Delay Seconds	The number of seconds to delay before creating each physical database connection. This delay takes place both during initial pool creation and during the lifetime of the pool whenever a physical database connection is created. Some database servers cannot handle multiple requests for connections in rapid succession. This property allows you to build in a small delay to let the database server catch up. This delay takes place both during initial pool creation and during the lifetime of the pool whenever a physical database connection is created	Integer in seconds	Default = 0
Initial Capacity	The number of physical database connections to create when configuring the pool. If unable to create this number of connections, creation of this Connection Pool will fail. This is also the minimum number of physical connections the pool will keep available.	Integer. Minimum = 0.	Default = 1.

Attribute	Description	Range of Values	Default Value
Maximum Capacity	This attribute sets maximum number of physical database connections that this connection pool can contain. Different JDBC drivers and database servers will likely limit the number of possible physical connections.	Integer. Legal Minimum =1.	Default = 1.
Capacity Increment	The increment by which the pool capacity is expanded. When there are no more available physical connections to service requests, the pool will create this number of additional physical database connections and add them to the pool. The pool will ensure that it does not exceed the maximum number of physical connections as set by `MaxCapacity`.	Integer. Minimum = 0.	Default = 1.
Allow Shrinking	Set to true to enable pool shrinking. Indicates whether or not the pool can shrink back to its `InitialCapacity` when connections are detected to not be in use.	Boolean Selected = true Not Selected = false	Default = true

Attribute	Description	Range of Values	Default Value
Shrink Period	This attribute sets the number of minutes to wait before shrinking a connection pool that has incrementally increased to meet demand. ShrinkingEnabled must be true for shrinking to take place.	Integer. Units in minutes. Minimum = 1.	Default = 15.

Refresh Period	This attribute sets the connection refresh interval. Every unused connection will be tested using TestTableName. Connections that do not pass the test will be closed and reopened in an attempt to reestablish a valid physical database connection. If TestTableName is not set then the test will not be performed.	Integer. Units in minutes.
		Minimum = 0.
		Maximum = 35791394
			Default = 0.

Supports Local Transaction	This attribute applies to XA connection pools only, and is ignored for non-XA driver. True if XA driver supports SQL with no global transaction.	Boolean Selected = true Not Selected = false	Not selected (default) = false.

Testing

Attribute	Description	Range of Values	Default Value
Test Table Name	This attribute sets the name of the table that will be used when testing a physical database connection. The default SQL used to test a connection is "select count" from TestTableName.The TestTableName must exist and be accessible to the database user for the connection. Most database servers optimize this SQL to avoid a table scan, but it is still a good idea to set TestTableName to the name of a table that is known to have few rows, or even no rows. If TestTableName begins with "SQL ", then the rest of the string, after that leading token, will be taken as a literal sql statement that will be used to test a connection.	String	null

Attribute	Description	Range of Values	Default Value
Test Connections On Reserve	This is attribute is used to determine if the connections on reserve will be tested. When set to true, the WebLogic Server tests a connection after giving it to the client. The test adds a small delay in serving the client's request for a connection from the pool, but ensures that the client receives a working connection (assuming that the DBMS is available and accessible).	Boolean Selected = true Not Selected = false	Default = false.
Test Connections On Release	This attribute is used to determine if connections that are being released are tested. When set to true, the WebLogic Server tests a connection before returning it to the connection pool. If all connections in the pool are already in use and a client is waiting for a connection, the client's wait will be slightly longer while the connection is tested.	Boolean Selected = true Not Selected = false	Default = false.

Targets

Servers

Attribute	Description	Range of Values	Default Value
Targets	This attribute sets the target servers for this deployment	String	The default is [Lweblogic.management.configuration.TargetMBean;@2c84d9

Clusters

Attribute	Description	Range of Values	Default Value
Targets	This attribute sets the target clusters for this deployment	String	The default is [Lweblogic.management.configuration.TargetMBean;@2c84d9

Monitoring

Attribute	Description	Range of Values	Default Value
Configured to Deploy On	This attribute is an integer that shows the number of servers on which the pool is configured to deploy.	Integer.	
Instances Currently Active	This attribute is an integer that shows the number of servers that are currently active for this pool.	Integer.	

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for user supplied information.	The value must be an alphanumeric string.	Null

For more information, see Managing JDBC Connectivity in the *Administration Guide* and *Programming WebLogic JDBC*, the developers' guide.

33 JDBC Connection Pool Runtime

By default, this pane allows the user to sort the objects by the following criteria:

- Server
- Machine
- Connections High
- Wait Seconds High

34 JDBC Data Sources

The following procedures describe how to use the Administration Console to set the attributes for configuring and managing Data Sources. Read about Data Sources in Managing JDBC Connectivity.

Configure a JDBC Data Source

1. Click to expand the JDBC node.
2. Click the Data Sources node. The Data Sources table displays in the right pane showing all the data sources defined in your domain.
3. Click the Configure a New JDBC Data Source text link. A dialog displays in the right pane showing the tabs associated with configuring a new data source.
4. Enter values in the Name, JNDI Name, and Pool Name attribute fields.
5. Click Create to create a data source instance with the name you specified in the Name field. The new instance is added under the Data Sources node in the left pane.

For what's next in the configuration process, see the Configuration Procedures in Managing JDBC Connectivity.

Clone a JDBC Data Source

1. Click to expand the JDBC node.
2. Click the Data Sources node. The Data Sources table displays in the right pane showing all the data sources defined in your domain.
3. Click the Clone icon in the row of the data source you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a data source.
4. Enter values in the Name, JNDI Name, and Pool Name attribute fields.
5. Click Create to create a data source instance with the name you specified in the Name field. The new instance is added under the Data Sources node in the left pane.

Monitor All Instances of a JDBC Data Source

1. Click to expand the JDBC node.
2. Click the Data Sources node in the left pane. The Data Sources table displays in the right pane showing all the data sources defined in the domain.
3. Click the Monitor All Instances icon in the row of the data source you want to monitor. A dialog displays in the right pane showing all instances of the data source deployed across the server domain.

Assign a JDBC Data Source

1. Click the instance node in the left pane under Data Sources for the source you want to assign. A dialog displays in the right pane showing the tabs associated with this instance.

2. Click the Targets tab.
3. Complete the following steps for the Servers, Groups, and Clusters tabs:
   a. Select one or more targets in the Available column that you want to assign to the data source.
   b. Click the mover control to move the targets you selected to the Chosen column.
   c. Click Apply to save your assignments.

Configuration

The following tables describe the attributes you use to configure and manage Data Sources. Read about Data Sources in Managing JDBC Connectivity.

General

Attribute	Description	Range of Values	Default Value
Name	This attribute reports the name of the data source.	The name can be up to 256 alphanumeric characters, but may not contain commas or spaces.	Null
JNDIName	This attribute sets the JNDI name associated with this data source. Sets the JNDI path to where this DataSource is bound.	String	Null
PoolName	Sets the name of the JDBC connection pool that is associated with this DataSource. DataSource will return a connection from the associated connection pool.	String	Null

Attribute	Description	Range of Values	Default Value
Row Prefetch Enabled	Set to true to enable row prefetching between client and WebLogic Server for each ResultSet. When an external client does JDBC access through Weblogic Server, row prefetching improves performance by fetching multiple rows from the server to the client in one server access. The WebLogic Server will ignore this setting and not use row prefetching when the client and WebLogic Server are in the same JVM.	Boolean Selected = true Not Selected = false	Not selected (default) = false.
Row Prefetch Size	The number of rows that will be prefetched between the client and WebLogic Server for each `ResultSet`. The optimal value is very dependent on the particulars of the query. Generally, increasing this number will increase performance, until a particular value is reached. At that point further increases do not result in any significant performance increase. Very rarely will increased performance result from exceeding 100 rows. The default value should be reasonable for most situations.	Legal Minimum = 2 Legal Maximum = 65536	Default = 48

Attribute	Description	Range of Values	Default Value
Stream Chunk Size	The number of rows that will be prefetched between client and WebLogic Server for each ResultSet. The optimal value is very dependent on the particulars of the query. Generally, increasing this number will increase performance, until a particular value is reached. At that point further increases do not result in any significant performance increase. Very rarely will increased performance result from exceeding 100 rows. The default value should be reasonable for most situations.	Integer. Minimum = 1. Maximum = 65,536.	Default = 256.

Target

Servers

Attribute	Description	Range of Values	Default Value
Target Servers	This attribute allows the user to select the servers to be used as targets for this datasource.	List	Null

Clusters

Attribute	Description	Range of Values	Default Value
Target Clusters	This attribute allows the user to select the clusters to be used as targets for this datasource.	List	Null

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for user supplied information.	The value must be an alphanumeric string.	Null

Read about Data Sources in Managing JDBC Connectivity.

35 JDBC MultiPools

The following procedures describe how to use the Administration Console to set the attributes for configuring and managing MultiPools. Read about configuring and managing MultiPools in Managing JDBC Connectivity.

Configure a JDBC MultiPool

1. Click to expand the JDBC node.
2. Click the MultiPools node. The MultiPools table displays in the right pane showing all the MultiPools defined in your domain.
3. Click the Configure a New JDBC MultiPool text link. A dialog displays in the right pane showing the tabs associated with configuring a new MultiPool.
4. On the General tab do the following:
   - Enter a value in the Name attribute field, then
   - Select either Load Balancing or High Availability from the drop-down list to chose how you want the connection pools selected from the MultiPool.
5. Click Create to create a MultiPool instance with the name you specified in the Name field. The new instance is added under the MultiPools node in the left pane.
6. On the Pools tab do the following:

   a. Select the connection pools from the Pool List in the Available column that you want to assign to the MultiPool.

b. Click the mover control to move the connection pools you selected to the Chosen column.

7. Click Apply to save your assignments.

For what's next in the configuration process, see the Configuration Procedures in Managing JDBC Connectivity.

Clone a JDBC MultiPool

1. Click to expand the JDBC node.
2. Click the MultiPools node. The MultiPools table displays in the right pane showing all the MultiPools defined in your domain.
3. Click the Clone icon in the row of the MultiPool you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a MultiPool.
4. Enter a value in the Name attribute field. Click either the Load Balancing or the High Availability checkbox to to chose how you want the connection pools selected from the MultiPool.
5. Click Create to create a MultiPool instance with the name you specified in the Name field. The new instance is added under the MultiPools node in the left pane.

Monitor All Instances of a JDBC MultiPool

1. Click to expand the JDBC node.
2. Click the MultiPools node. The MultiPools table displays in the right pane showing all the MultiPools defined in the domain.
3. Click the Monitor All Instances icon in the row of the MultiPool you want to monitor. A dialog displays in the right pane showing all instances of the MultiPool deployed across the connection-pool domain.

Assign a JDBC MultiPool to One or More Servers or Clusters

1. Click the instance node in the left pane under MultiPools for the MultiPool you want to assign. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Targets tab.
3. Complete the following steps for the Servers or Clusters tab:
   a. Select the server in the Available column to which you want to assign the MultiPool.
   b. Click the mover control to move the target you selected to the Chosen column.
   c. Click Apply to save your assignments.

Configuration

Attribute	Description	Range of Values	Default Value
Name	This attribute sets the name of the JDBC Multi pool.	The name can be up to 256 alphanumeric characters, but may not contain commas or spaces.	Null

Attribute	Description	Range of Values	Default Value
Algorithm Type	Chose how you want the connection pools selected from the MultiPool:		

	Load Balancing. Distributes the connection requests evenly (round-robin from the next available pool) to its member pools.	Boolean
		True = selected
		False = not selected
			Not Selected

	High Availability. Distributes the connection requests sequentially from an ordered list.
	That is, every time the Multipool is asked for a connection, it tries to get a connection from the first pool in its list. If unable to get a valid connection, it tries the next pool in its list. Process is repeated until a valid connection is obtained, or until the end of the list is reached, in which case an exception is thrown.
	Note that the Multipool will only move to the next pool in the list when there is a real problem with the pool, for example the database is down or the pool disabled. For the cases where all connections are busy, the Multipool behaves as a single pool and an exception is thrown.
		Boolean
		True = selected
		False = not selected
			Selected

Pools

Attribute	Description	Range of Values	Default Value
Pool List	This attribute retrieves the list of connection pools in the multi pool	List	

Targets

Attribute	Description	Range of Values	Default Value
Targets Servers			

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for user supplied information.	The value must be an alphanumeric string.	Null

Read about configuring and managing MultiPools in Managing JDBC Connectivity.

36 JDBC Transaction Data Sources

This table describes the attributes you set in the Administration Console to configure and manage Transaction Data Sources. Read about Transaction Data Sources in Managing JDBC Connectivity.

Configuration

Attribute	Description	Range of Values	Default Value
Name	This attribute returns the name of this data source.	String	MyJDBC Tx Data Source
JNDI Name	Sets the JNDI path to where this TxDataSource is bound. Applications that look up the JNDI path will get a `javax.sql.DataSource` instance that corresponds to this TxDataSource. Note that the old style usage of `DriverManager.getConnection`() or `Driver.Connect()` has been deprecated in favor of looking up a datasource to obtain a connection.	Valid JNDI name	Null
Pool Name	This attribute allows the user to set the name of the connection pool that is associated with this TxDataSource. Calls to `getConnection()` on this TxDataSource will return a connection from the associated connection pool.	Valid connection pool name	Null

Attribute	Description	Range of Values	Default Value
Enable Two-Phase Commit	This attribute allows the user to enable or disable the two phase commitment of transactions. Enable Two-Phase Commit allows non-XA JDBC drivers to pretend to participate in a the JDBC connection is the only participant in the transaction, but it is asking for trouble to have more than one resource participating in a transaction where one of them (the JDBC driver) is pretending to be an XA resource. There is a possibility of heuristic failures in such a situation. Set Enable Two-PhaseCommit (true) should be used only if you have no option (no decent XA driver available for a particular database, for example). If this TxDataSource is associated with an XA connection pool, or if there is only one resource participating in the distributed transaction, then this setting is ignored.	Boolean Enabled = Selected Not Enabled = Not Selected	Not Selected

Attribute	Description	Range of Values	Default Value
Row Prefetch Enabled	Set to True to enable row prefetching between client and WebLogic Server for each ResultSet. When an external client does JDBC access through WebLogic Server, row prefetching improves performance by fetching multiple rows from the server to the client in one server access. The WebLogic Server will ignore this setting and not use row prefetching when the client and WebLogic Server are in the same JVM.	Boolean Enabled = Selected Not Enabled = Not Selected	Not Selected
Row Prefetch Size	The number of rows that will be prefetched between the client and WebLogic Server for each `ResultSet`. The optimal value is very dependent on the particulars of the query. Generally, increasing this number will increase performance, until a particular value is reached. At that point further increases do not result in any significant performance increase. Very rarely will increased performance result from exceeding 100 rows. The default value should be reasonable for most situations.	Legal Minimum = 2 Legal Maximum = 65536	Default = 48

Attribute	Description	Range of Values	Default Value
Stream Chunk Size	Determines the data chunk size for streaming datatypes. Streaming datatypes (for example resulting from a call to `getBinarySTream()`)will be pulled in Stream Chunk Size-sized chunks from the WebLogic Server to the client as needed.	Legal Minimun = 1 Legal Maximum= 65,536 Units in bytes.	Default = 256

Targets

Servers

Attribute	Description	Range of Values	Default Value
Target Servers	This attribute allows the user to select the servers to be used as targets for this datasource.	List	Null

Clusters

Attribute	Description	Range of Values	Default Value
Target Clusters	This attribute allows the user to select the clusters to be used as targets for this datasource.	List	Null

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for optional user-supplied information.	The value must be an alphanumeric string.	Null

37 JMS Connection Consumer

Create a JMS Connection Consumer

1. Click to expand the JMS node.
2. Click to expand the JMS Servers node.
3. Click to expand a server instance under JMS Servers.
4. Click to expand the JMS Session Pools node.
5. Click to expand a session pool instance under JMS Session Pools.
6. Click the JMS Consumers node. The JMS Consumers table displays in the right pane showing all the connection consumers.
7. Click the Configure a new JMS Connection Consumer text link. A dialog displays in the right pane showing the tabs associated with configuring a new connection consumer.
8. Enter values in the attribute fields.
9. Click Create to create a connection consumer instance with the name you specified in the Name field. The new instance is added under the JMS Consumers node in the left pane.

Clone a JMS Connection Consumer

1. Click to expand the JMS node.
2. Click to expand the JMS Servers node.
3. Click to expand a server instance under JMS Servers.
4. Click to expand the JMS Session Pools node.
5. Click to expand a session pool instance under JMS Session Pools.
6. Click the JMS Consumers node. The JMS Consumers table displays in the right pane showing all the connection consumers.
7. Click the Clone icon in the row of the connection consumer you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a connection consumer.
8. Enter values in the attribute fields.
9. Click Create to create a connection consumer instance with the name you specified in the Name field. The new instance is added under the JMS Consumers node in the left pane.

Delete a JMS Connection Consumer

1. Click to expand the JMS node.
2. Click to expand the JMS Servers node.
3. Click to expand a server instance under JMS Servers.
4. Click to expand the JMS Session Pools node.
5. Click to expand a session pool instance under JMS Session Pools.
6. Click the JMS Consumers node. The JMS Consumers table displays in the right pane showing all the connection consumers.

7. Click the Delete icon in the row of the connection consumer that you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.

Click Yes to delete the connection consumer. The connection consumer icon under the JMS Consumers node is deleted.

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	Name of the connection consumer. This attribute is not dynamically configurable.	Valid Java identifier, unique within a specific JMS Server sessionpool.	MyJMS Connection Consumer[-*n*]
Messages Maximum	Maximum number of messages that can be accumulated by the connection consumer. A value of -1 indicates that there is no maximum. This attribute is dynamically configurable, but the change does not take effect until the connection consumer's session pool is restarted.	-1, and 1 through $2^{63} – 1$	-1
Selector	JMS selector expression used to filter messages. For information on defining selectors, see *Programming WebLogic JMS*. This attribute is not dynamically configurable.	Valid JMS message selector expression	N/A
Destination	Destination on which the connection consumer will listen. This attribute is not dynamically configurable.	Valid JNDI name or null	null

Notes

Attribut e	Description	Range of Values	Default Value
Notes	This attribute provides a space for optional user-supplied information.	The value must be an alphanumeric string.	Null

38 JMS Connection Factories

Create a JMS Connection Factory

1. Click to expand the JMS node.
2. Click the JMS Connection Factories node. The JMS Connection Factories table displays in the right pane showing all the connection factories defined in your domain.
3. Click the Configure a new JMS Connection Factory text link. A dialog displays in the right pane showing the tabs associated with configuring a new connection factory.
4. Enter values in the attribute fields.
5. Click Create to create a connection factory instance with the name you specified in the Name field. The new instance is added under the JMS Connection Factories node in the left pane.
6. Assign the JMS Connection Factory to a WebLogic Server.

For more information see “Assign a JMS Connection Factory”.

Clone a JMS Connection Factory

1. Click to expand the JMS node.
2. Click the JMS Connection Factories node. The JMS Connection Factories table displays in the right pane showing all the connection factories defined in your domain.
3. Click the Clone icon in the row of the connection factory you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a connection factory.
4. Enter values in the attribute fields.
5. Click Create to create a connection factory instance with the name you specified in the Name field. The new instance is added under the JMS Connection Factories node in the left pane.
6. Assign the JMS Connection Factory to a WebLogic Server.

For more information see “Assign a JMS Connection Factory”.

Delete a JMS Connection Factory

1. Click to expand the JMS node.
2. Click the JMS Connection Factories node. The JMS Connection Factories table displays in the right pane showing all the connection factories defined in your domain.
3. Click the Delete icon in the row of the connection factory you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.
4. Click Yes to delete the connection factory. The connection factory icon under the JMS Connection Factories node is deleted.

Assign a JMS Connection Factory

1. Click the instance node in the left pane under JMS Connection Factories for the connection factory you want to assign. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Targets tab.
3. Complete the following steps for the Servers and Clusters tabs:

   a. Select one or more targets in the Available column that you want to assign to the connection factory.

   b. Click the mover control to move the targets you selected to the Chosen column.

   c. Click Apply to save your assignments.

Configuration

General

Attribut e	Description	Range of Values	Default Value
Name	Name of the connection factory. The JNDI name is configured separately. This attribute is not dynamically configurable.	Java identifier, unique within specific cluster	MyJMS Connection Factory[-*n*]
JNDI Name	Name that is assigned to and used to look up the connection factory within the JNDI namespace. The connection factory name is configured separately. This attribute is not dynamically configurable.	Java identifier, unique within JNDI namespace	null
Client Id	Client ID that can be used for clients with durable subscribers. For more information about durable subscribers, see *Programming WebLogic JMS*. This attribute is not dynamically configurable.	Java identifier	null

Attribute	**Description**	**Range of Values**	**Default Value**
Default Priority	Default priority used for messages for which a priority is not explicitly defined. For information about defining the priority when sending messages, see *Programming WebLogic JMS*. This attribute is dynamically configurable, but changes will only affect new connections; existing connections will not be impacted.	0 through 9	4
Default Time to Live	Default maximum length of time, in milliseconds, that a message will exist. Used for messages for which a priority was not explicitly defined. A value of 0 indicates that the message has an infinite amount time to live. For information about defining the time-to-live when sending messages, see *Programming WebLogic JMS*. This attribute is dynamically configurable, but changes will only affect new connections; existing connections will not be impacted.	0 through $2^{63} – 1$	0

Attribute	Description	Range of Values	Default Value
Default Delivery Mode	Default delivery mode used for messages for which a delivery mode is not explicitly defined. For more information about delivery modes and defining them when sending messages, see *Programming Weblogic JMS*. This attribute is dynamically configurable, but changes will only affect new connections; existing connections will not be impacted.	Persistent or Non-Persistent	Persistent
Default Redelivery Delay	Default time delay, in milliseconds, before rolled back or recovered messages are redelivered. For more information about delivery modes and defining them when sending messages, see *Programming Weblogic JMS*. This attribute is dynamically configurable, but changes will only affect new connections; existing connections will not be impacted.	-1 through 2^{63} – 1	0

Attribute	Description	Range of Values	Default Value
Default Time To Deliver	Default time delay, in milliseconds, between when a message is produced and when it is made visible on its destination. For more information about delivery modes and defining them when sending messages, see *Programming Weblogic JMS*. This attribute is dynamically configurable, but changes will only affect new connections; existing connections will not be impacted.	-1 through $2^{63} – 1$	0

Attribute	**Description**	**Range of Values**	**Default Value**
Messages Maximum	Maximum number of messages that may exist for an asynchronous session and that have not yet been passed to the message listener. A value of -1 indicates that there is no limit on the number of messages. In this case, however, the limit is set to the amount of remaining virtual memory. When the number of messages reaches the Messages Maximum value: ■ For multicast sessions, new messages are discarded according the policy specified by the Overrun Policy attribute and a DataOverrunException is thrown. ■ For non-multicast sessions, new messages are flow-controlled, or retained on the server until the application can accommodate the messages. For multicast sessions, when a connection is stopped, messages will continue to be delivered but only until the Messages Maximum value is reached. Once this value is reached, messages will be discarded based on the Overrun policy. For more information about asynchronous messages and multicasting, see *Programming WebLogic*	-1, 1 through $2^{31} - 1$	10

Attribut e	Description	Range of Values	Default Value
Overrun Policy	Overrun policy for multicast sessions. When the number of outstanding messages reaches the Messages Maximum attribute value, messages are discarded based on the specified policy. If set to KeepNew, the most recent messages are given priority over the oldest messages, and the oldest messages are discarded, as needed. If set to KeepOld, the oldest messages are given priority over the most recent messages, and the most recent messages are discarded, as needed. Message age is defined by the order of receipt, not by the `JMSTimestamp` value. For more information about asynchronous messages and multicasting, see *Programming WebLogic JMS*. This attribute is dynamically configurable, but changes will only affect new connections; existing connections will not be impacted.	KeepNew, KeepOld	KeepOld

Attribute	Description	Range of Values	Default Value

Allow Close In OnMessage	Flag for specifying whether or not a connection factory creates message consumers that allow a `close()` method to be issued within its `onMessage()` method call.
	If selected, a `close()` method call from within an `onMessage()` method call will succeed instead of blocking forever.
	If the acknowledge mode of the session is set to `AUTO_ACKNOWLEDGE`, the current message will still be acknowledged automatically when the `onMessage()` call completes.
	The `onMessage()` method is implemented to support asynchronous message delivery as described in *Programming WebLogic JMS*.
		Boolean
		Enabled = selected
		Disabled = not selected
			Not Selected

Attribute	Description	Range of Values	Default Value
Acknowledge Policy	The message acknowledge policy for the connection factory. This attribute only applies to implementations that use the `CLIENT_ACKNOWLEDGE` acknowledge mode for a non-transacted session. - `All` — acknowledge all messages ever received by a given session, regardless of which message calls the acknowledge method. - `Previous` — acknowledge all messages received by a given session, but only up to and including the message that calls the acknowledge method. For more information about message acknowledge modes, see *Programming Weblogic JMS.* **Note:** The `All` default represents a change from versions of JMS prior to WebLogic release 6.1, which defaulted to `Previous`. For more information, see “Migrating WebLogic JMS Applications” in Programming WebLogic JMS.	All or Previous	All

Transactions

Attribute	Description	Range of Values	Default Value
Transaction Timeout	Timeout value (in seconds) for transacted sessions. If a transacted session is still active after the timeout has elapsed, the transaction is rolled back. A value of 0 indicates that the default value will be used. If you have long-running transactions, you might want to adjust the value of this attribute to allow transactions to complete. This attribute is dynamically configurable, but changes will only affect new connections; existing connections will not be impacted.	0 through $2^{31} - 1$	3600

Attribute	Description	Range of Values	Default Value

User Transactions Enabled	Flag for specifying whether or not a connection factory creates sessions that are JTA aware. If set, the associated message producers and message consumers look into the running thread for a transaction context. Otherwise, the current JTA transaction will be ignored.
	However, if the XA Connection Factory Enabled flag is set, the User Transactions Enabled attribute is ignored (since it is always considered true).
	Note: Transacted sessions ignore the current threads transaction context in favor of their own internal transaction, regardless of the setting. This setting only affects non-transacted sessions.
	This attribute is dynamically configurable, but changes will only affect new connections; existing connections will not be impacted.
		Boolean
		Enabled = selected
		Disabled = not selected
			Not Selected

Attribute	Description	Range of Values	Default Value

XA Connection Factory Enabled	Flag for specifying whether or not a XA queue or XA topic connection factory is returned, instead of a queue or topic connection factory. It can be used to return a XA queue session or XA topic session, which has the `getXAResource` method.
	This attribute is dynamically configurable, but changes will only affect new connections; existing connections will not be impacted.
		Boolean
		Enabled = selected
		Disabled = not selected
			Not Selected

Targets

Servers

Attribute	Description	Range of Values	Default Value
Target Servers	This attribute provides a list of potential targets for servers. When a JMS client creates a connection, it will connect to one of these targets, using this factory.	List of available and chosen targets.	No targets are chosen

Clusters

Attribute	Description	Range of Values	Default Value
Target Clusters	This attribute provides a list of potential targets for clusters. Only one target can be assigned to each JMS Server.	List of available and chosen targets.	No targets are chosen

Notes

Attribut e	Description	Range of Values	Default Value
Notes	This attribute provides a space for optional user-supplied information.	The value must be an alphanumeric string.	Null

39 JMS Connection Runtime

By default, this pane allows the user to sort the objects by the following criteria:

- Client ID
- Active sessions
- Most sessions
- Total sessions

40 JMS Consumer Runtime

By default, this pane allows the user to sort the objects by the following criteria:

- Name
- Messages Maximum
- Selector
- Destination

41 JMS Destinations

Create a JMS Queue

1. Click to expand the JMS node.
2. Click to expand the JMS Servers node.
3. Click to expand a server instance under JMS Servers.
4. Click the JMS Destinations node. The JMS Destinations table displays in the right pane showing all the JMS queues.
5. Click the Configure a new JMS Queue text link. A dialog displays in the right pane showing the tabs associated with configuring a new queue.
6. Enter values in the attribute fields.
7. Click Create to create a queue instance with the name you specified in the Name field. The new instance is added under the JMS Destinations node in the left pane.
8. Click the remaining tabs individually and change the attribute fields or accept the default values as assigned.
9. Click Apply to save any changes you made.

Clone a JMS Queue

1. Click to expand the JMS node.

2. Click to expand the JMS Servers node.
3. Click to expand a server instance under JMS Servers.
4. Click the JMS Destinations node. The JMS Destinations table displays in the right pane showing all the JMS queues.
5. Click the Clone icon in the row of the queue you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a queue.
6. Enter values in the attribute fields.
7. Click the Create button in the lower right corner to create a queue instance with the name you specified in the Name field. The new instance is added under the JMS Destinations node in the left pane.
8. Click the remaining tabs individually and change the attribute fields or accept the default values as assigned.
9. Click Apply to save any changes you made.

Delete a JMS Queue

1. Click to expand the JMS node.
2. Click to expand the JMS Servers node.
3. Click to expand a server instance under JMS Servers.
4. Click the JMS Destinations node. The JMS Destinations table displays in the right pane showing all the JMS queues.
5. Click the Delete icon in the row of the queue you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.
6. Click Yes to delete the queue. The queue icon under the JMS Destinations node is deleted.

Create a JMS Topic

1. Click to expand the JMS node.
2. Click to expand the JMS Servers node.
3. Click to expand a server instance under JMS Servers.
4. Click the JMS Destinations node. The JMS Destinations table displays in the right pane showing all the JMS topics.
5. Click the Create a new JMS Topic text link. A dialog displays in the right pane showing the tabs associated with configuring a new topic.
6. Enter values in the attribute fields.
7. Click the Create button in the lower right corner to create a topic instance with the name you specified in the Name field. The new instance is added under the JMS Destinations node in the left pane.
8. Click the remaining tabs individually and change the attribute fields or accept the default values as assigned.
9. Click Apply to save any changes you made.

Clone a JMS Topic

1. Click to expand the JMS node.
2. Click to expand the JMS Servers node.
3. Click to expand a server instance under JMS Servers.
4. Click the JMS Destinations node. The JMS Destinations table displays in the right pane showing all the JMS topics.
5. Click the Clone icon in the row of the topic you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a topic.

6. Enter values in the attribute fields.
7. Click the Create button in the lower right corner to create a topic instance with the name you specified in the Name field. The new instance is added under the JMS Destinations node in the left pane.
8. Click the remaining tabs individually and change the attribute fields or accept the default values as assigned.
9. Click Apply to save any changes you made.

Delete a JMS Topic

1. Click to expand the JMS node.
2. Click to expand the JMS Servers node.
3. Click to expand a server instance under JMS Servers.
4. Click the JMS Destinations node. The JMS Destinations table displays in the right pane showing all the JMS topics.
5. Click the Delete icon in the row of the topic you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.
6. Click Yes to delete the topic. The topic icon under the JMS Destinations node is deleted.

Monitor All Active JMS Destinations

See "Monitor All Active JMS Destinations" in the JMS Server online help for more information.

Configuration

General

Attribute	Description	Range of Values	Default
Name	Name of the destination The JNDI name is configured separately. This attribute is not dynamically configurable.	Java identifier, unique within the JMSServer	My JMS Destination[-*n*]
JNDI Name	Name used to look up the destination within the JNDI namespace. The destination name is configured separately. If not specified, the destination name is not advertised through the JNDI namespace and cannot be looked up and used. This attribute is not dynamically configurable.	Java identifier, unique within the JNDI namespace scope	null

Attribute	Description	Range of Values	Default
Enable Store	Flag specifying whether or not the destination uses the persistent store specified by the JMS server. ■ If this flag is enabled, but no persistent store is defined for the JMS Server, then the configuration fails and WebLogic JMS does not boot. ■ If this flag is disabled, then the destination does not support persistent messages. ■ If this flag is set to Default, then the destination uses the persistent store—if one is defined for the JMS server—and supports persistent messaging. This attribute is not dynamically configurable.	false, true, default	default
Template	The JMS template from which the destination is derived. If this attribute is not defined, then the attributes for the destination must be specified as part of the destination. The Template attribute setting per destination is static. The template attributes, however, can be modified dynamically. **Note:** Attributes that are set to their default values will inherit their destination values from the JMS template at run time.	Existing JMS template name or none	none

Attribute	Description	Range of Values	Default
Destination Keys	This attribute provides a list of potential destination keys for sorting the messages that arrive on the destination. They are ordered from most significant to least significant. If more than one key is specified, a key based on the JMSMessageID can only be the last key in the list. **Note:** If JMSMessageID is not defined in the key, it is implicitly assumed to be the last key and set as "Ascending" (FIFO) for the sort order.	List	Null

Thresholds and Quotas

Attribute	Description	Range of Values	Default
Bytes Maximum	Maximum number of bytes that may be stored in the destination. A value of -1 specifies that there is no limit on the number of bytes that can be stored in the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through 2^{63}–1, Greater than or equal to Bytes Threshold High	-1
Bytes Threshold High	Upper threshold value based on the number of bytes stored in the destination. If the number of bytes exceeds this threshold—and if bytes paging is enabled and a paging store has been configured for the JMS Server—then destination-level bytes paging is started, and a message is logged on the server indicating a high threshold condition. A value of -1 specifies that destination bytes paging and threshold log messages are disabled for the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted. **Note:** Bytes paging cannot be dynamically disabled by resetting the Bytes High Threshold to -1. To disable paging, you could set the Bytes High Threshold to a very large number (max is 2^{63}–1), so that paging would not be triggered.	-1, 0 through 2^{63}–1, Less than or equal to Bytes Maximum, >Bytes Threshold Low	-1

Attribute	Description	Range of Values	Default
Bytes Threshold Low	Lower threshold value based on the number of bytes stored in the destination. If the number of bytes falls below this threshold, then destination-level bytes paging is stopped (if paging is occurring), and a message is logged on the server indicating that the threshold condition has cleared. A value of -1 specifies that destination bytes paging and threshold log messages are disabled for the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through 2^{63}–1, <Bytes Threshold High	-1

Attribute	Description	Range of Values	Default
Bytes Paging Enabled	Flag for specifying whether or not bytes paging is enabled on the destination. ■ If this attribute is set to False, then destination-level byte paging is explicitly disabled for this destination. ■ If this attribute is set to True, a paging store has been configured for the JMS Server, and both the Bytes Threshold Low and Bytes Threshold High attribute values are greater than -1, then destination-level bytes paging is enabled for this destination. ■ If this attribute is set to Default, then this value inherits the JMS template's value—if a JMS template is specified. If no template is configured for the destination, then the Default value is equivalent to False. **Note:** If server-level bytes paging is enabled, and destination-level paging is disabled for a given destination, than messages on the destination can still be paged if server-level paging is started. However, when destination-level paging is disabled for a given destination, then the destination-level high thresholds will not force the destination to page out messages when they are exceeded.	String True False Default	Default
Messages Maximum	Maximum number of messages that may be stored in the destination. A value of -1 specifies that there is no limit on the number of messages that can be stored in the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through 2^{63}–1, Greater than or equal to Messages Threshold High	-1

Attribute	Description	Range of Values	Default
Messages Threshold High	Upper threshold value based on the number of messages stored in the destination. If the number of messages exceeds this threshold—and if messages paging is enabled and a paging store has been configured for the JMS Server—then destination-level paging is started, and a message is logged on the server indicating a high threshold condition. A value of -1 specifies that destination messages paging and threshold log messages are disabled for the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted. **Note:** Messages paging cannot be dynamically disabled by resetting the Messages High Threshold to -1. To disable paging, you could set the Messages High Threshold to a very large number (max is 2^{63} –1), so that paging would not be triggered.	-1, 0 through 2^{63}–1, Less than or equal to Messages Maximum, >Bytes Threshold Low	-1
Messages Threshold Low	Lower threshold value based on the number of messages stored in the destination. If the number of messages falls below this threshold, then destination-level messages paging is stopped (if paging is occurring) and a message is logged on server, indicating that the threshold condition has cleared. A value of -1 specifies that destination messaes paging and threshold log messages are disabled for the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through 2^{63}–1, <Messages Threshold High	-1

Attribute	Description	Range of Values	Default
Messages Paging Enabled	Flag for specifying whether or not messages paging is enabled on the destination. ■ If this attribute is set to False, then destination-level message paging is explicitly disabled for this destination. ■ If this attribute is set to True, a paging store has been configured for the JMS Server, and both the Message Threshold Low and Message Threshold High attribute values are greater than -1, then destination-level messages paging is enabled for this destination. ■ If this attribute is set to Default, then this value inherits the JMS template's value—if a JMS template is specified. If no template is configured for the destination, then the Default value is equivalent to False. **Note:** If server-level messages paging is enabled, and destination-level paging is disabled for a given destination, than messages on the destination can still be paged if server-level paging is started. However, when destination-level paging is disabled for a given destination, then the destination-level high thresholds will not force the destination to page out messages when they are exceeded.	String True False Default	Default

Overrides

Attribute	Description	Range of Values	Default
Priority Override	The priority value assigned to all messages that arrive at the destination, regardless of the priority specified by the message producer. The default value (-1) specifies that the destination will not override the Priority setting. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through 9	-1
Time To Live Override	The time-to-live value assigned to all messages that arrive at the destination, regardless of the time-to-live specified by the message producer. The default value (-1) specifies that the destination will not override the Time To Live setting. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through 2^{63}–1	-1
Time To Deliver Override	Defines the default delay, in milliseconds, between when a message is produced and when it is made visible on its destination, regardless of the delivery time specified by the producer and/or connection factory. The default value (-1) specifies that the destination will not override the Time To Deliver setting. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	A long integer greater than or equal to -1, or string syntax for a delivery schedule	-1

Attribute	Description	Range of Values	Default
Delivery Mode Override	Delivery mode assigned to all messages that arrive at the destination regardless of the delivery mode specified by the message producer. A value of `No-Delivery` specifies that the delivery mode will not be overridden. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	Persistent, Non-Persistent, or No-Delivery	No-Delivery

Redelivery

Attribute	Description	Range of Values	Default
Redelivery Delay Override	Defines the delay, in milliseconds, before rolled back or recovered messages are redelivered, regardless of the redelivery delay specified by the consumer and/or connection factory. The default value (-1) specifies that the destination will not override the Redelivery Delay setting. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through 2^{63}–1	-1

Attribute	Description	Range of Values	Default
Redelivery Limit	The number of redelivery tries a message can have before it is placed in the error destination. Depending on whether an error destination is configured, the following occurs when the redelivery limit is reached: ■ If no error destination is configured, or the quota for the error destination would be exceeded, then persistent and non-persistent messages are simply dropped. ■ If an error destination is configured and the error destination is at quota, then an error message is logged and the message is dropped. However, if the message is persistent, it remains in the persistent store. This ensures that a persistent message will be redelivered when WebLogic Server is rebooted. The default value (-1) specifies that the destination will not override the Redelivery Limit setting. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through 2^{63}–1	-1
Error Destination	A destination for messages that have reached their redelivery limit. If the error destination is null, then such messages are simply dropped. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	Existing destination or none	none

Multicast

Attribute	Description	Range of Values	Default
Multicast Address	IP address used for multicasting. This address is used to transmit messages to multicast consumers. This attribute is valid for topics only and is not dynamically configurable.	A well-formed IP address (Class D)	N/A
Multicast TTL	Time-to-live value used for multicasting, specifying the number of routers that the message can traverse en route to the consumers. A value of 0 indicates that the message will not traverse any routers, and is limited to one subnet. This value is independent of the `JMSExpirationTime` value. This attribute is valid for topics only and is not dynamically configurable.	0 to 255	0
Multicast Port	IP port used for multicasting. This port is used to transmit messages to multicast consumers. This attribute is valid for topics only and is not dynamically configurable.	1024 to 65535	6001

Monitor Destinations

Attribute	Description	Range of Values	Default
Destination	Destination name.		N/A
Server	Associated server name.		N/A
Consumers	Current number of registered message consumers.		N/A
Consumers High	Greatest number of registered message consumers at any given time.		N/A
Consumers Total	Total number of registered message consumers.		N/A
Bytes Current	Current number of bytes stored.		N/A
Bytes Pending	Number of unacknowledged and uncommitted transactional bytes stored.		N/A
Bytes Received	Number of bytes received.		N/A
Bytes Threshold Time	Amount of time that the destination has spent in a bytes threshold condition since the last time the server was booted.		N/A
Messages	Current number of messages stored on this destination.		N/A
Messages High	Greatest number of messages stored at any given time.		N/A

Attribute	Description	Range of Values	Default
Messages Pending	Number of unacknowledged and uncommitted transactional messages stored.		N/A
Messages Received	Number of messages received.		N/A
Messages Threshold Time	Amount of time that the destination spent in a message threshold condition.		N/A

Notes

Attribut e	Description	Range of Values	Default
Notes	This attribute provides a space for user-supplied information.	The value must be an alphanumeric string.	Null

42 JMS Destination Key

Create a JMS Destination Key

1. Click to expand the JMS node.
2. Click the JMS Destination Keys node. The JMS Destinations Keys table displays in the right pane showing all the destination keys.
3. Click the Create a new JMS Destination Key text link. A dialog displays in the right pane showing the tabs associated with configuring a new destination key.
4. Enter values in the attribute fields.
5. Click Create to create a destination key instance with the name you specified in the Name field. The new instance is added under the JMS Destination Keys node in the left pane.

Clone a JMS Destination Key

1. Click to expand the JMS node.
2. Click the JMS Destination Keys node. The JMS Destinations Keys table displays in the right pane showing all the destination keys.
3. Click the Clone icon in the row of the destination key that you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a destination key.

4. Enter values in the attribute fields.
5. Click Create to create a destination key instance with the name you specified in the Name field. The new instance is added under the JMS Destination Keys node in the left pane.

Delete a JMS Destination Key

1. Click to expand the JMS node.
2. Click the JMS Destination Keys node. The JMS Destinations Keys table displays in the right pane showing all the destination keys.
3. Click the Delete icon in the row of the destination key that you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.

Click Yes to delete the destination key. The destination key icon under the JMS Destination Keys node is deleted.

Configuration

General

Attribut e	Description	Range of Values	Default Value
Name	Name of the key. This attribute is not dynamically configurable.	Java identifier, unique within a specific domain.	MyJMS Destination Key[-*n*]
Property	Property name on which to sort. This value indicates a message property name or the name of a message header field on which to sort. Message header field keys ignore the key type *and* reference message header fields rather than message properties. **Note:** To improve performance, we recommend using message header fields as sorting keys, rather than message properties. This attribute is not dynamically configurable.	JMS Property name (including user properties), or message header fields that can be sorted on, as follows: ■ JMSMessageID ■ JMSTimestamp ■ JMSCorrelationID ■ JMSPriority ■ JMSExpiration ■ JMSType ■ JMSRedelivered ■ JMSDeliveryTime	N/A
Key Type	Expected property type. This attribute is not dynamically configurable.	Boolean, Byte, Double, Float, Int, Long, Short, String	String

Attribute	Description	Range of Values	Default Value
Direction	Direction in which to sort the destinations. Choosing Ascending for JMSMessageID implies a FIFO sort order (the default for destinations). Set the value to Descending for a LIFO sort order. This attribute is not dynamically configurable.	Ascending or Descending	Ascending

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for user-supplied information.	The value must be an alphanumeric string.	Null

43 JMS Destination Runtime

Monitor Destinations

Attribute	Description	Range of Values	Default
Destination	Destination name.		N/A
Server	Associated server name.		N/A
Consumers	Current number of registered message consumers.		N/A
Consumers High	Greatest number of registered message consumers at any given time.		N/A
Consumers Total	Total number of registered message consumers.		N/A
Bytes Current	Current number of bytes stored.		N/A
Bytes Pending	Number of unacknowledged and uncommitted transactional bytes stored.		N/A

Attribute	Description	Range of Values	Default
Bytes Received	Number of bytes received.		N/A
Bytes Threshold Time	Amount of time that the destination has spent in a bytes threshold condition since the last time the server was booted.		N/A
Messages	Current number of messages stored on this destination.		N/A
Messages High	Greatest number of messages stored at any given time.		N/A
Messages Pending	Number of unacknowledged and uncommitted transactional messages stored.		N/A
Messages Received	Number of messages received.		N/A
Messages Threshold Time	Amount of time that the destination spent in a message threshold condition.		N/A

44 JMS Durable Subscriber Runtime

By default, this pane allows the user to monitor a JMS durable subscriber by the following criteria:

Attribute	Description	Range of Values	Default Value
Client ID	The client ID for this durable subscriber.		N/A
Subscription Name	The subscription name for this durable subscriber.		N/A
No Local	The `noLocal` Boolean for this durable subscriber.		N/A
Active	Determines whether this subscription is being used by a durable subscriber.		N/A
Selector	A durable subscriber selector.		N/A
Messages Pending Count	The number of messages pending (uncommitted and unacknowledged) by this durable subscriber.		N/A
Messages Current Count	The number of messages still available by this durable subscriber.		N/A
Bytes Pending Count	The number of bytes still pending by this durable subscriber.		N/A

Attribute	Description	Range of Values	Default Value
Bytes Current Count	The number of bytes received by this durable subscriber.		N/A

45 JMS File Store

Create a JMS File Store

1. Click to expand the JMS node.
2. Click the JMS Stores node. The JMS Stores table displays in the right pane showing all the JMS file stores.
3. Click the Create a new JMS File Store text link. A dialog displays in the right pane showing the tabs associated with configuring a new file store.
4. Enter values in the attribute fields.
5. Click Create to create a file store instance with the name you specified in the Name field. The new instance is added under the JMS Stores node in the left pane.

Clone a JMS File Store

1. Click to expand the JMS node.
2. Click the JMS Stores node. The JMS Stores table displays in the right pane showing all the JMS file stores.
3. Click the Clone icon in the row of the file store you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a file store.
4. Enter values in the attribute fields.

5. Click Create to create a file store instance with the name you specified in the Name field. The new instance is added under the JMS Stores node in the left pane.

Delete a JMS File Store

1. Click to expand the JMS node.
2. Click the JMS Stores node. The JMS Stores table displays in the right pane showing all the JMS file stores.
3. Click the Delete icon in the row of the file store you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.
4. Click Yes to delete the file store. The file store icon under the JMS Stores node is deleted.

Configuration

General

Attribut e	Description	Range of Values	Default Value
Name	Name of the backing store (or stores). This name will be used as a prefix for names of the store file(s). This attribute is not dynamically configurable.	Java identifier, unique within the JMS server	MyJMSFile Store[-*n*]
Directory	Specifies the directory within which the file backing store exists. This attribute is not dynamically configurable.	Valid directory on system on which backing store is instantiated	N/A

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for optional user-supplied information.	The value must be an alphanumeric string.	Null

46 JMSJDBCStore

Create a JMS JDBC Store

1. Click to expand the JMS node.
2. Click the JMS Stores node. The JMS Stores table displays in the right pane showing all the JMS JDBC stores.
3. Click the Create a new JMS JDBC Store text link. A dialog displays in the right pane showing the tabs associated with configuring a new JDBC store.
4. Enter values in the attribute fields.
5. Click Create to create a JDBC store instance with the name you specified in the Name field. The new instance is added under the JMS Stores node in the left pane.

Clone a JMS JDBC Store

1. Click to expand the JMS node.
2. Click the JMS Stores node. The JMS Stores table displays in the right pane showing all the JMS JDBC stores.
3. Click the Clone icon in the row of the JDBC store you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a JDBC store.
4. Enter values in the attribute fields.

5. Click Create to create a JDBC store instance with the name you specified in the Name field. The new instance is added under the JMS Stores node in the left pane.

Delete a JMS JDBC Store

1. Click to expand the JMS node.
2. Click the JMS Stores node. The JMS Stores table displays in the right pane showing all the JMS file stores.
3. Click the Delete icon in the row of the JDBC store you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.
4. Click Yes to delete the JDBC store. The JDBC store icon under the JMS Stores node is deleted.

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	Name of the JMS JDBC store. This attribute is not dynamically configurable.	Java identifier, unique within the JMSServer	MyJMSJDBC Store[-*n*]
Connection Pool	Name of the JDBC Connection Pool for this backing store. This attribute is not dynamically configurable.	Existing JDBC ConnectionPool name or none	none

Attribute	Description	Range of Values	Default Value
Prefix Name	The prefix name of JMS tables in the backing store. Specifying unique prefixes allows multiple stores to exist in the same database. The prefix is prepended to table names when: ■ The DBMS requires fully qualified names. ■ You must differentiate between JMS tables for two WebLogic servers, enabling multiple tables to be stored on a single DBMS. The prefix should be specified using the following format and will result in a valid table name when prepended to the JMS table name: `[[catalog.]schema.]prefix` **Note:** No two JMS stores should be allowed to use the same database tables, as this will result in data corruption.	Valid table prefix for JMS tables	N/A

Notes

Attribute	**Description**	**Range of Values**	**Default Value**
Notes	This attribute provides a space for user-supplied information.	The value must be an alphanumeric string.	Null

47 JMS Producer Runtime

By default, this pane allows the user to sort the objects by the following criteria:

- Destination
- Server
- Consumers
- Messages
- Messages Received
- Bytes Current
- Bytes Pending

48 JMS Queue

Create a JMS Queue

1. Click to expand the JMS node.
2. Click to expand the JMS Servers node.
3. Click to expand a server instance under JMS Servers.
4. Click the Destinations node. The JMS Destinations table displays in the right pane showing all the JMS queue and topic destinations.
5. Click the Configure a new JMS Queue text link. A dialog displays in the right pane showing the tabs associated with configuring a new destination queue.
6. Enter values in the attribute fields.
7. Click Create to create a destination queue instance with the name you specified in the Name field. The new instance is added under the Destinations node in the left pane.

Clone a JMS Queue

1. Click to expand the JMS node.
2. Click to expand the JMS Servers node.
3. Click to expand a server instance under JMS Servers.

4. Click the Destinations node. The Destinations table displays in the right pane showing all the destinations.
5. Click the Clone icon in the row of the destination topic you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a destination queue.
6. Enter values in the attribute fields.
7. Click Create to create a destination queue instance with the name you specified in the Name field. The new instance is added under the Destinations node in the left pane.

Delete a JMS Queue

1. Click to expand the JMS node.
2. Click to expand the JMS Servers node.
3. Click to expand a server instance under JMS Servers.
4. Click the Destination node. The JMS Destinations table displays in the right pane showing all the session pools.
5. Click the Delete icon in the row of the destination queue you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.
6. Click Yes to delete the destination queue. The topic icon under the Destinations node is deleted.

Monitor All Active JMS Destinations

See “Monitor All Active JMS Destinations” in the JMS Server online help for more information.

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	Name of the queue. The JNDI name is configured separately. This attribute is not dynamically configurable.	Java identifier, unique within the JMSServer	MyJMSQueue [-*n*]
JNDI Name	Name used to look up the destination within the JNDI namespace. The destination name is configured separately. If not specified, the destination name is not advertised through the JNDI namespace. This attribute is not dynamically configurable.	Java identifier, unique within the JNDI namespace scope	null

Attribute	**Description**	**Range of Values**	**Default Value**
Enable Store	Flag specifying whether or not the queue uses the persistent store specified by the JMS server. ■ If this flag is enabled, but no persistent store is defined for the JMS Server, then the configuration fails and WebLogic JMS does not boot. ■ If this flag is disabled, then the queue does not support persistent messages. ■ If this flag is set to Default, then the queue uses the persistent store—if one is defined for the JMS server—and supports persistent messaging. This attribute is not dynamically configurable.	false, true, default	default
Template	The JMS template from which the queue is derived. If this attribute is not defined, then the attributes for the queue must be specified as part of the destination. The Template attribute setting per destination is static. The template attributes, however, can be modified dynamically.	Existing JMS template name or none	none
Destination Keys	This attribute provides a list of potential destination keys for sorting the messages that arrive on the queue. They are ordered from most significant to least significant. A key that is based on JMSMessageID can only be the last key. **Note:** If JMSMessageID is not defined in the key, it is implicitly assumed to be the last key and set as “Ascending” (FIFO) for the sort order.	List	Null

Thresholds & Quotas

Attribute	Description	Range of Values	Default Value
Bytes Maximum	Maximum number of bytes that may be stored in the destination. A value of -1 specifies that there is no limit on the number of bytes that can be stored in the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through $2^{63} - 1$, Less than or equal to Bytes Threshold High	-1

Attribute	Description	Range of Values	Default Value
Bytes Threshold High	Upper threshold value based on the number of bytes stored in the destination. If the number of bytes exceeds this threshold—and if bytes paging is enabled and a paging store has been configured for the JMS Server—then destination-level bytes paging is started, and a message is logged on the server indicating a high threshold condition. A value of -1 specifies that destination bytes paging and threshold log messages are disabled for the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted. **Note:** Bytes paging cannot be dynamically disabled by resetting the Bytes High Threshold to -1. To disable paging, you could set the Bytes High Threshold to a very large number (max is 2^{63}–1), so that paging would not be triggered.	-1, 0 through $2^{63} - 1$, Less than or equal to Bytes Maximum, >Bytes Threshold Low	-1
Bytes Threshold Low	Lower threshold value based on the number of bytes stored in the destination. If the number of bytes falls below this threshold, then destination-level bytes paging is stopped (if paging is occurring), and a message is logged on the server indicating that the threshold condition has cleared. A value of -1 specifies that destination bytes paging and threshold log messages are disabled for the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through $2^{63} - 1$, <Bytes Threshold High	-1

Attribute	Description	Range of Values	Default Value
Bytes Paging Enabled	Flag for specifying whether or not bytes paging is enabled on the destination. ■ If this attribute is set to False, then destination-level byte paging is explicitly disabled for this destination. ■ If this attribute is set to True, a paging store has been configured for the JMS Server, and both the Bytes Threshold Low and Bytes Threshold High attribute values are greater than -1, then destination-level bytes paging is enabled for this destination. ■ If this attribute is set to Default, then this value inherits the JMS template's value—if a JMS template is specified. If no template is configured for the destination, then the Default value is equivalent to False. **Note:** If server-level bytes paging is enabled, and destination-level paging is disabled for a given destination, than messages on the destination can still be paged if server-level paging is started. However, when destination-level paging is disabled for a given destination, then the destination-level high thresholds will not force the destination to page out messages when they are exceeded.	String True False Default	Default

Attribute	Description	Range of Values	Default Value
Messages Maximum	Maximum number of messages that may be stored in the destination. A value of -1 specifies that there is no limit on the number of messages that can be stored in the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through $2^{63} - 1$, Less than or equal to Bytes Threshold High	-1
Messages Threshold High	Upper threshold value based on the number of messages stored in the destination. If the number of messages exceeds this threshold—and if messages paging is enabled and a paging store has been configured for the JMS Server—then destination-level paging is started, and a message is logged on the server indicating a high threshold condition. A value of -1 specifies that destination messages paging and threshold log messages are disabled for the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted. **Note:** Messages paging cannot be dynamically disabled by resetting the Messages High Threshold to -1. To disable paging, you could set the Messages High Threshold to a very large number (max is 2^{63} –1), so that paging would not be triggered.	-1, 0 through $2^{63} - 1$, Less than or equal to Messages Maximum, >Messages Threshold Low	-1

Attribute	Description	Range of Values	Default Value
Messages Threshold Low	Lower threshold value based on the number of messages stored in the destination. If the number of messages falls below this threshold, then destination-level messages paging is stopped (if paging is occurring) and a message is logged on server, indicating that the threshold condition has cleared. A value of -1 specifies that destination messaes paging and threshold log messages are disabled for the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through $2^{63} - 1$, <Messages Threshold High	-1

Attribute	Description	Range of Values	Default Value
Messages Paging Enabled	Flag for specifying whether or not messages paging is enabled on the destination. ■ If this attribute is set to False, then destination-level message paging is explicitly disabled for this destination. ■ If this attribute is set to True, a paging store has been configured for the JMS Server, and both the Message Threshold Low and Message Threshold High attribute values are greater than -1, then destination-level messages paging is enabled for this destination. ■ If this attribute is set to Default, then this value inherits the JMS template's value—if a JMS template is specified. If no template is configured for the destination, then the Default value is equivalent to False. **Note:** If server-level messages paging is enabled, and destination-level paging is disabled for a given destination, than messages on the destination can still be paged if server-level paging is started. However, when destination-level paging is disabled for a given destination, then the destination-level high thresholds will not force the destination to page out messages when they are exceeded.	String True False Default	Default

Overrides

Attribute	**Description**	**Range of Values**	**Default Value**
Priority Override	Priority assigned to all messages that arrive at the destination, regardless of the priority specified by the message producer. The default value (-1) specifies that the destination will not override the Priority setting. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through 9	-1
Time To Live Override	Time-to-live assigned to all messages that arrive at the destination, regardless of the time-to-live specified by the message producer. The default value (-1) specifies that the destination will not override the Time To Live setting. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through $2^{63} - 1$	-1
Time To Deliver Override	Defines the default delay, in milliseconds, between when a message is produced and when it is made visible on its destination, regardless of the delivery time specified by the producer and/or connection factory. The default value (-1) specifies that the destination will not override the Time To Deliver setting. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through $2^{63} - 1$, or string syntax for a delivery schedule	-1

Attribute	Description	Range of Values	Default Value
Delivery Mode Override	Delivery mode assigned to all messages that arrive at the destination regardless of the delivery mode specified by the message producer. The default `No-Delivery` value specifies that the delivery mode will not be overridden. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	Persistent, Non-Persistent, or No-Delivery	No-Delivery

Monitoring

Monitor All Active JMS Destinations

Attribute	Description	Range of Values	Default Value
Destination	Destination name.		N/A
Server	Associated server name.		N/A
Consumers	Current number of registered message consumers.		N/A
Consumers High	Greatest number of registered message consumers at any given time.		N/A
Consumers Total	Total number of registered message consumers.		N/A
Bytes Current	Current number of bytes stored.		N/A
Bytes Pending	Number of unacknowledged and uncommitted transactional bytes stored.		N/A
Bytes Received	Number of bytes received.		N/A

Attribute	Description	Range of Values	Default Value
Bytes Threshold Time	Amount of time that the destination has spent in a bytes threshold condition since the last time the server was booted.		N/A
Messages	Current number of messages stored at this destination.		N/A
Messages High	Greatest number of messages stored at any given time.		N/A
Messages Pending	Number of unacknowledged and uncommitted transactional messages stored.		N/A
Messages Received	Number of messages received.		N/A
Messages Threshold Time	Amount of time that the destination spent in a message threshold condition.		N/A

For additional information about monitoring JMS Queues, please see Monitor All Active JMS Services.

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for user-supplied information.	The value must be an alphanumeric string.	Null

49 JMS Runtime

By default, this pane allows the user to sort the objects by the following criteria:

- Server
- Connections
- Connections high
- Connections total
- Servers
- Servers high
- Servers total

50 JMS Server

Create a JMS Server

1. Click to expand the JMS node.
2. Click the JMS Servers node. The JMS Servers table displays in the right pane showing all the servers defined in your domain.
3. Click the Configure a new JMS Server text link. A dialog displays in the right pane showing the tabs associated with configuring a new server.
4. Enter values in the attribute fields.
5. Click Create to create a server instance with the name you specified in the Name field. The new instance is added under the JMS Servers node in the left pane. A JMS Destinations node and a Session Pools node are automatically added under the new server instance by default.
6. Assign the JMS Server to a WebLogic Server.

For more information, see Assign a JMS Server.

Clone a JMS Server

1. Click to expand the JMS node.
2. Click the JMS Servers node. The JMS Servers table displays in the right pane showing all the servers defined in your domain.

3. Click the Clone icon in the row of the server you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a server.
4. Enter values in the attribute fields.
5. Click Create to create a server instance with the name you specified in the Name field. The new instance is added under the JMS Servers node in the left pane. A JMS Destinations node and a Session Pools node are automatically added under the new server instance by default.
6. Assign the JMS Server to a WebLogic Server.

For more information, see “Assign a JMS Server”.

Delete a JMS Server

1. Click to expand the JMS node.
2. Click the JMS Servers node. The JMS Servers table displays in the right pane showing all the servers defined in your domain.
3. Click the Delete icon in the row of the server you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.
4. Click Yes to delete the server. The server icon under the JMS Servers node is deleted.

Monitor All Active JMS Services

1. Click to expand the JMS node.
2. Click the JMS Servers node. The JMS Servers table displays in the right pane showing all the servers defined in your domain.
3. Click the Monitor all Active JMS Services text link. A dialog displays in the right pane showing all active JMS services for the current domain

Monitor All Instances of a JMS Server

1. Click to expand the JMS node.
2. Click the JMS Servers node. The JMS Servers table displays in the right pane showing all the servers defined in your domain.
3. Click the Monitor All Instances icon in the row of the server you want to monitor. A dialog displays in the right pane showing all instances of the server deployed across the server domain.

Monitor All Active JMS Destinations

1. Click to expand the JMS node.
2. Click the JMS Servers node. The JMS Servers table displays in the right pane showing all the servers defined in your domain.
3. Click the Monitor all Active JMS Destinations text link. A dialog displays in the right pane showing all active JMS destinations for the current domain.

Monitor All Active JMS Session Pools

1. Click to expand the JMS node.
2. Click the JMS Servers node. The JMS Servers table displays in the right pane showing all the servers defined in your domain.
3. Click the Monitor all Active JMS Session Pools Runtime text link. A dialog displays in the right pane showing all active JMS session pools for the current domain.

Assign a JMS Server

1. Click the instance node in the left pane under JMS Servers for the server you want to assign. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Targets tab.
3. Complete the following steps for the Servers and Clusters tabs:
    a. Select one or more targets in the Available column that you want to assign to the data source.

    b. Click the mover control to move the targets you selected to the Chosen column.

    c. Click Apply to save your assignments.

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	Name of the JMS server. This attribute is not dynamically configurable.	Java identifier, unique within specific cluster	MyJMS Server[-*n*]
Store	The existing persistent store for the JMS server. A persistent store may only be used by one JMS server. A value of none specifies that no persistent messaging is supported. If no persistent store is specified, then destinations on this server will not support persistent messages or durable subscribers. This attribute is not dynamically configurable.	Existing persistent store name or none	none
Paging Store	The name of the persistent store where non-persistent messages are paged for the JMS server. A paging store cannot be the same store used for persistent messages or durable subscribers. A paging store may only be used by one JMS server. A value of none specifies that message paging is not supported. If no paging store is specified, then the server and destinations will not support message paging. **Note:** Using a JDBC store is not recommended for paging since the amount of traffic and subsequent lack of performance would make such a configuration undesirable.	Existing persistent store or none	none

Attribute	Description	Range of Values	Default Value
Temporary Template	Name of an existing JMS template used to create all temporary destinations, including temporary queues and temporary topics. The attribute values for a temporary destination are derived from the this JMS template. If provided as part of the template, the Store attribute values are ignored because temporary destinations do not support persistent messaging. This attribute is dynamically configurable. **Note:** If this attribute is set to none, attempts to create a temporary destination (queue or topic) will fail.	JMS template name or none	none (The default values for the destination attribute are used.)

Thresholds and Quotas

Attribute	Description	Range of Values	Default Value
Bytes Maximum	Maximum number of bytes that may be stored in the JMS server. A value of -1 specifies that there is no limit on the number of bytes that can be stored in the JMS server. This attribute is dynamically configurable.	-1, 0 through 2^{63}–1, Greater than or equal to Bytes Threshold High	-1
Bytes Threshold High	Upper threshold value based on the number of bytes stored in the JMS server. If the number of bytes exceeds this threshold—and if bytes paging is enabled and a paging store has been configured—then server bytes paging is started, and a message is logged on the server indicating a high threshold condition. A value of -1 specifies that server bytes paging and threshold log messages are disabled for the JMS server. This attribute is dynamically configurable. **Note:** Bytes paging cannot be dynamically disabled by resetting the Bytes High Threshold to -1. To disable paging, you could set the Bytes High Threshold to a very large number (max is 2^{63}–1), so that paging would not be triggered.	-1, 0 through 2^{63}–1, Less than or equal to Bytes Maximum, >Bytes Threshold Low	-1

Attribute	Description	Range of Values	Default Value
Bytes Threshold Low	Lower threshold value based on the number of bytes stored in the JMS server. If the number of bytes falls below this threshold, then server bytes paging is stopped (if paging is occurring), and a message is logged on the server indicating that the threshold condition has cleared. A value of -1 specifies that server paging and threshold log messages are disabled for the JMS server. This attribute is dynamically configurable.	-1, 0 through 2^{63}–1, <Bytes Threshold High	-1

Bytes Paging Enabled	Flag for specifying whether or not bytes paging is enabled on the JMS server.
	■ If this flag is not selected, then server bytes paging is explicitly disabled.
	■ If this flag is selected, a paging store has been configured, and both the Bytes Threshold Low and Bytes Threshold High attribute values are greater than -1, then server bytes paging is enabled.
	■ If either the Bytes Threshold Low or Bytes Threshold High attribute is undefined, or defined as -1, then server bytes paging is implicitly disabled—even though this flag is selected.
		Boolean
		Enabled = selected
		Disabled = not selected
			Not selected

Messages Maximum	Maximum number of messages that may be stored in the JMS server. A value of -1 specifies that there is no limit on the number of messages that can be stored in the JMS server. This attribute is dynamically configurable.	-1, 0 through 2^{63}–1, Greater than or equal to Messages Threshold High	-1

Attribute	Description	Range of Values	Default Value
Messages Threshold High	Upper threshold value based on the number of messages stored in the JMS server. If the number of messages exceeds this threshold—and if messages paging is enabled and a paging store has been configured—then server messages paging is started, and a message is logged on the server indicating a high threshold condition. A value of -1 specifies that server messages paging and threshold log messages are disabled for the JMS server. This attribute is dynamically configurable. **Note:** Messages paging cannot be dynamically disabled by resetting the Messages High Threshold to -1. To disable paging, you could set the Messages High Threshold to a very large number (max is 2^{63} –1), so that paging would not be triggered.	-1, 0 through 2^{63}–1, Less than or equal to Messages Maximum, >Messages Threshold Low	-1
Messages Threshold Low	Lower threshold value based on the number of messages stored in the JMS server. If the number of messages falls below this threshold, then server messages paging stops (if paging is occurring), and a message is logged on the server, indicating that the threshold condition has cleared. A value of -1 specifies that server paging and threshold log messages are disabled for the JMS server. This attribute is dynamically configurable.	-1, 0 through 2^{63}–1, <Messages Threshold High	-1

Attribute	Description	Range of Values	Default Value

Messages Paging Enabled	Flag for specifying whether or not messages paging is enabled on the JMS server.
	■ If this flag is not selected, then server-level messages paging is explicitly disabled.
	■ If this flag is selected, a paging store has been configured, and both the Messages Threshold Low and Messages Threshold High attribute values are greater than -1, then serve-level messages paging is enabled.
	■ If either the Messages Threshold Low or Messages Threshold High attribute is undefined, or defined as -1, then server-level messages paging is implicitly disabled–even though this flag is selected.
		Boolean
		Enabled = selected
		Disabled = not selected
			Not selected

Monitoring

Monitor All Active JMS Services

Attribute	Description	Range of Values	Default Value
Server	Service name	N/A	N/A
Connections	Current number of connections	N/A	N/A
Connections High	Greatest number of connections at any given time	N/A	N/A
Connections Total	Total number of connections	N/A	N/A
Servers	Current number of servers	N/A	N/A
Servers High	Greatest number of servers at any given time	N/A	N/A
Servers Total	Total number of servers	N/A	N/A

Monitor All Instances of JMS Server

Attribute	Description	Range of Values	Default Value
Name	Name of JMS server		N/A
Destinations	List of destinations.		N/A
Destinations High	Greatest number of instantiated destinations at any given time.		N/A
Destinations Total	Total number of instantiated destinations.		N/A
Bytes Current	Current number of bytes stored across all destinations in the JMS server.		N/A
Bytes Pending	Number of bytes sent and received that are pending the result of an uncommitted transaction or an acknowledgement.		N/A
Bytes Received Count	Total number of bytes received for all destinations.		N/A
Bytes Threshold Time	Amount of time spent in a bytes threshold condition.		N/A
Messages Current	Current number of messages stored across all destinations.		N/A

Messages High	Greatest number of messages stored across all destinations at a given time.	N/A
Messages Received	Total number of messages received for all destinations.	N/A
Messages Threshold Time	Amount of time spent in a messages threshold condition.	N/A
Messages Total	Total number of messages stored across all destinations	N/A
Session Pools	List of session pools defined for the JMS server.	N/A
Session Pools High	Greatest number of session pools at any given time.	N/A
Session Pools Total	Total number of instantiated sessions pools.	N/A

Monitor Destinations

Attribute	Description	Range of Values	Default Value
Destination	Destination name.		N/A
Server	Associated server name.		N/A
Consumers	Current number of registered message consumers.		N/A
Consumers High	Greatest number of registered message consumers at any given time.		N/A
Consumers Total	Total number of registered message consumers.		N/A
Bytes Current	Current number of bytes stored.		N/A
Bytes Pending	Number of unacknowledged and uncommitted transactional bytes stored.		N/A
Bytes Received	Number of bytes received.		N/A
Bytes Threshold Time	Amount of time that the destination has spent in a bytes threshold condition since the last time the server was booted.		N/A
Messages	Current number of messages stored at this destination.		N/A

Attribute	Description	Range of Values	Default Value
Messages High	Greatest number of messages stored at any given time.		N/A
Messages Pending	Number of unacknowledged and uncommitted transactional messages stored.		N/A
Messages Received	Number of messages received.		N/A
Messages Threshold Time	Amount of time that the destination spent in a message threshold condition.		N/A

Monitor Server Session Pools

Attribute	Description	Range of Values	Default Value
Session Pool	Name of session pool.		
Server	Name of associated server.		
Consumers	List of consumers.		
Consumers High	Greatest number of consumers on the session pool at any given time.		
Consumers Total	Total number of consumers.		

Targets

Attribute	Description	Range of Values	Default Value
Targets	This attribute provides a list of potential targets. Currently a JMS server can have a maximum of one target.	List	No targets are chosen.

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for user-supplied information.	The value must be an alphanumeric string.	Null

51 JMS Server Runtime

Monitor JMS Servers

Attribute	Description	Range of Values	Default
Destination	Destination name.		N/A
Session Pools	Associated session pools names.		N/A
Consumers	Current number of registered message consumers.		N/A
Bytes Current	Current number of bytes stored.		N/A
Bytes Pending	Number of unacknowledged and uncommitted transactional bytes stored.		N/A
Bytes Received	Number of bytes received.		N/A
Bytes Threshold Time	Amount of time that the destination has spent in a bytes threshold condition since the last time the server was booted.		N/A
Messages	Current number of messages stored on this destination.		N/A

Attribute	Description	Range of Values	Default
Messages High	Greatest number of messages stored at any given time.		N/A
Messages Pending	Number of unacknowledged and uncommitted transactional messages stored.		N/A
Messages Received	Number of messages received.		N/A
Messages Threshold Time	Amount of time that the destination spent in a message threshold condition.		N/A

52 JMS Session Pools

Create a JMS Session Pool

1. Click to expand the JMS node.
2. Click to expand the JMS Servers node.
3. Click to expand a server instance under JMS Servers.
4. Click the Session Pools node. The Session Pools table displays in the right pane showing all the session pools.
5. Click the Configure a new JMS Session Pool text link. A dialog displays in the right pane showing the tabs associated with configuring a new session pool.
6. Enter values in the attribute fields.
7. Click Create to create a session pool instance with the name you specified in the Name field. The new instance is added under the Session Pools node in the left pane.

Clone a JMS Session Pool

1. Click to expand the JMS node.
2. Click to expand the JMS Servers node.
3. Click to expand a server instance under JMS Servers.

4. Click the Session Pools node. The Session Pools table displays in the right pane showing all the session pools.
5. Click the Clone icon in the row of the session pool you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a session pool.
6. Enter values in the attribute fields.
7. Click Create to create a session pool instance with the name you specified in the Name field. The new instance is added under the Session Pools node in the left pane.

Delete a JMS Session Pool

1. Click to expand the JMS node.
2. Click to expand the JMS Servers node.
3. Click to expand a server instance under JMS Servers.
4. Click the Session Pools node. The Session Pools table displays in the right pane showing all the session pools.
5. Click the Delete icon in the row of the session pool you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.
6. Click Yes to delete the session pool. The session pool icon under the Session pools node is deleted.

Monitor All Active JMS Session Pools

See “Monitor All Active JMS Session Pools” in the JMS Server online help for more information.

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	Name of the session pool. This attribute is not dynamically configurable.	Java identifier, unique within a specific cluster	MyJMS Session Pool[-*n*]
Connection Factory	Connection factory associated with the session pool.	Existing connection factory name or none	N/A
Listener Class	Name of the message listener class that provides an `onMessage()` method that, in turn, processes a message. This attribute is not dynamically configurable.	Existing message listener class name or java class in your CLASSPATH	N/A

Attribute	Description	Range of Values	Default Value
Acknowledge Mode	Acknowledge mode used by non-transacted sessions within a session pool. For transacted sessions, messages are acknowledged automatically when the session is committed, and this field is ignored. For more information about acknowledge modes, see *Programming WebLogic JMS* This attribute is not dynamically configurable.	Auto, Client, Dups OK, and None	Auto
Sessions Maximum	Number of concurrent sessions in the session pool. A value of -1 indicates that there is no maximum. This attribute is dynamically configurable, however, it does not take effect until the session pool is restarted.	-1, and 1 through $2^{63} – 1$	-1

Transacted	Flag specifying whether or not the session pool creates transacted sessions.
	This attribute is not dynamically configurable.
		Enabled = selected
		Disabled = not selected
			Not Selected

Monitoring

Monitor All JMS Session Pool Runtimes

Attribute	Description	Range of Values	Default Value
Session Pool	Name of session pool.		N/A
Server	Name of associated server.		N/A
Consumers	List of consumers.		N/A
Consumers High	Greatest number of consumers on the session pool at any given time.		N/A
Consumers Total	Total number of consumers.		N/A

For additional information about monitoring JMS Session Pools, please see Monitor All Active JMS Session Pools.

Notes

Attribut e	Description	Range of Values	Default Value
Notes	This attribute provides a space for optional user-supplied information.	The value must be an alphanumeric string.	Null

53 JMS Sessionpool Runtime

By default, this pane allows the user to sort the objects by the following criteria:

- Session Pool
- Server
- Consumers
- Consumers High
- Consumers Total

54 JMS Stores

JMS stores information can be found in the following files:

- JMS File Store
- JMSJDBCStore

55 JMS Templates

Create a JMS Template

1. Click the JMS node in the left pane.
2. Click the JMS Templates node. The JMS Templates table displays in the right pane showing all the templates defined in the domain.
3. Click the Configure a new JMS Template text link. A dialog displays in the right pane showing the tabs associated with configuring a new template.
4. Enter a value in the Name attribute field.
5. Click Create to create a template instance with the name you specified in the Name field. The new instance is added under the JMS Templates node in the left pane.
6. Click the remaining tabs individually and change the attribute fields or accept the default values as assigned.
7. Click Apply to save any changes you made.

Clone a JMS Template

1. Click the JMS node in the left pane.
2. Click the JMS Templates node. The JMS Templates table displays in the right pane showing all the templates defined in the domain.

3. Click the Clone icon in the row of the template you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a new template.
4. Enter a value in the Name attribute field.
5. Click the Create button in the lower right corner to create a template instance with the name you specified in the Name field. The new instance is added under the JMS Templates node in the left pane.
6. Click the remaining tabs individually and change the attribute fields or accept the default values as assigned.
7. Click Apply to save any changes you made.

Delete a JMS Template

1. Click the JMS node in the left pane.
2. Click the JMS Templates node. The JMS Templates table displays in the right pane showing all the templates defined in the domain.
3. Click the Delete icon in the row of the server you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.
4. Click Yes to delete the template. The template icon under the JMS Templates node is deleted.

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	Name of the JMS template. This attribute is not dynamically configurable.	Java identifier, unique within specific cluster	MyJMS Template[-*n*]
Destination Keys	This attribute provides a list of potential destination keys for sorting the messages as they arrive on the destination. They are ordered from most significant to least significant. If more than one key is specified, a key based on the JMSMessageID can only be the last key in the list. **Note:** If JMSMessageID is not defined in the key, it is implicitly assumed to be the last key and set as “Ascending” (FIFO) for the sort order.	List	Null

Thresholds and Quotas

Attribute	Description	Range of Values	Default Value
Bytes Maximum	Maximum number of bytes that may be stored in the destination. A value of -1 specifies that there is no limit on the number of bytes that can be stored in the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through 2^{63}–1, Greater than or equal to Bytes Threshold High	-1
Bytes Threshold High	Upper threshold value based on the number of bytes stored in the destination. If the number of bytes exceeds this threshold—and if bytes paging is enabled and a paging store has been configured for the JMS Server—then destination-level bytes paging is started, and a message is logged on the server indicating a high threshold condition. A value of -1 specifies that bytes paging and threshold log messages are disabled for the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted. **Note:** Bytes paging cannot be dynamically disabled by resetting the high threshold to -1. To disable paging, you could set the high threshold to very large number (max is 2^{63} -1), so that paging would not be triggered.	-1, 0 through 2^{63}–1, Less than or equal to Bytes Maximum, >Bytes Threshold Low	-1

Attribute	Description	Range of Values	Default Value
Bytes Threshold Low	Lower threshold value based on the number of bytes stored in the destination. If the number of bytes falls below this threshold, then destination-level bytes paging is stopped (if paging is occurring), and a message is logged on the server indicating that the threshold condition has cleared. A value of -1 specifies that destination bytes paging and threshold log messages are disabled for the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through 2^{63}–1, <Bytes Threshold High	-1

Bytes Paging Enabled	Flag for specifying whether or not bytes paging is enabled on the destination.
	■ If this flag is not selected, then bytes paging is disabled for the template's destinations—unless the destination setting overrides the template.
	■ If this flag is selected, a paging store has been configured for the JMS Server, and both the Bytes Threshold Low and Bytes Threshold High attribute values are greater than -1, then bytes paging is enabled for the template's destinations—unless the destination setting overrides the template.
	Note: If no value is defined in the JMS Template MBean, then it defaults to False and bytes paging is disabled for the template's destinations—unless the destination setting overrides the template.
		Boolean
		Enabled = selected
		Disabled = not selected
			Not selected

Attribute	Description	Range of Values	Default Value
Messages Maximum	Maximum number of messages that may be stored in the destination. A value of -1 specifies that there is no limit on the number of messages that can be stored in the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through 2^{63}–1, Greater than or equal to Messages Threshold High	-1
Messages Threshold High	Upper threshold value based on the number of messages stored in the destination. If the number of messages exceeds this threshold—and if messages paging is enabled and a paging store has been configured for the JMS Server—then destination-level messages paging is started, and a message is logged on the server indicating a high threshold condition. A value of -1 specifies that destination messages paging and threshold log messages are disabled for the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted. **Note:** Messages paging cannot be dynamically disabled by resetting the High Threshold to -1. To disable paging, you could set the High Threshold to a very large number (max is 2^{63} -1), so that paging would not be triggered.	-1, 0 through 2^{63}–1, Less than or equal to Messages Maximum, >Messages Threshold Low	-1

Attribute	Description	Range of Values	Default Value
Messages Threshold Low	Lower threshold value based on the number of messages stored in the destination. If the number of messages falls below this threshold, then destination-level messages paging is stopped (if paging is occurring), and a message is logged on the server indicating that the threshold condition has cleared. A value of -1 specifies that destination mesages paging and threshold log messages are disabled for the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through 2^{63}–1, <Bytes Threshold High	-1

Messages Paging Enabled	Flag for specifying whether or not messages paging is enabled on the destination.
	■ If this flag is not selected, then messages paging is disabled for the template's destinations–unless the destination setting overrides the template.
	■ If this flag is selected, a paging store has been configured for the JMS Server, and both the Messages Threshold Low and Messages Threshold High attribute values are greater than -1, then messages paging is enabled for the template's destinations—unless the destination setting overrides the template.
	Note: If no value is defined in the JMS Template MBean, then it defaults to False and messages paging is disabled for the template's destinations—unless the destination setting overrides the template.
		Boolean
		Enabled = selected
		Disabled = not selected
			Not selected

Override

Attribute	Description	Range of Values	Default Value
Priority Override	The priority assigned to all messages that arrive at the destination regardless of the priority specified by the message producer. The default value (-1) specifies that the destination will not override the Priority setting. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through 9	-1
Time to Live Override	The time-to-live value assigned to all messages that arrive at the destination, regardless of the time-to-live specified by the message producer. The default value (-1) specifies that the destination will not override the Time To Live setting. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through 2^{63}–1	-1
Time To Deliver Override	Defines the default delay, in milliseconds, between when a message is produced and when it is made visible on its destination, regardless of the delivery time specified by the producer and/or connection factory. The default value (-1) specifies that the destination will not override the Time To Deliver setting. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through 2^{63}–1, or string syntax for a delivery schedule	-1

Attribute	Description	Range of Values	Default Value
Delivery Mode Override	Delivery mode assigned to all messages that arrive at the destination regardless of the delivery mode specified by the message producer. A value of `No-Delivery` specifies that the delivery mode will not be overridden. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	Persistent, Non-Persistent, or No-Delivery	No-Delivery

Redelivery

Attribute	Description	Range of Values	Default
Redelivery Delay Override	Defines the delay, in milliseconds, before rolled back or recovered messages are redelivered, regardless of the redelivery delay specified by the consumer and/or connection factory. The default value (-1) specifies that the destination will not override the Redelivery Delay setting. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through 2^{63}–1	-1

Attribute	Description	Range of Values	Default
Redelivery Limit	The number of redelivery tries a message can have before it is placed in the error destination. Depending on whether an error destination is configured, the following occurs when the redelivery limit is reached: ■ If no error destination is configured, or the quota for the error destination would be exceeded, then persistent and non-persistent messages are simply dropped. ■ If an error destination is configured and the error destination is at quota, then an error message is logged and the message is dropped. However, if the message is persistent, it remains in the persistent store. This ensures that a persistent message will be redelivered when WebLogic Server is rebooted. The default value (-1) specifies that the destination will not override the redelivery limit setting. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through 2^{63}–1	-1
Error Destination	A destination for messages that have reached their redelivery limit. If the error destination is null, then such messages are simply dropped. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	Existing destination or none	none

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for optional user-supplied information.	The value must be an alphanumeric string.	Null

56 JMS Topic

Create a JMS Topic

1. Click to expand the JMS node.
2. Click to expand the JMS Servers node.
3. Click to expand a server instance under JMS Servers.
4. Click the Destinations node. The JMS Destinations table displays in the right pane showing all the JMS topic and queue destinations.
5. Click the Configure a new JMS Topic text link. A dialog displays in the right pane showing the tabs associated with configuring a new destination topic.
6. Enter values in the attribute fields.
7. Click Create to create a destination topic instance with the name you specified in the Name field. The new instance is added under the Destinations node in the left pane.

Clone a JMS Topic

1. Click to expand the JMS node.
2. Click to expand the JMS Servers node.
3. Click to expand a server instance under JMS Servers.

4. Click the Destinations node. The Destinations table displays in the right pane showing all the destinations.
5. Click the Clone icon in the row of the destination topic you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a destination topic.
6. Enter values in the attribute fields.
7. Click Create to create a destination topic instance with the name you specified in the Name field. The new instance is added under the Destinations node in the left pane.

Delete a JMS Topic

1. Click to expand the JMS node.
2. Click to expand the JMS Servers node.
3. Click to expand a server instance under JMS Servers.
4. Click the Destination node. The JMS Destinations table displays in the right pane showing all the session pools.
5. Click the Delete icon in the row of the destination topic you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.
6. Click Yes to delete the destination topic. The topic icon under the Destinations node is deleted.

Monitor All Active JMS Destinations

See “Monitor All Active JMS Destinations” in the JMS Server online help for more information.

Configuration

General

Attribute	Description	Range of Values	Default
Name	Name of the topic. The JNDI name is configured separately. This attribute is not dynamically configurable.	Java identifier, unique within the JMS Server	MyJMS Destination[-*n*]
JNDI Name	Name used to look up the destination within the JNDI namespace. The destination name is configured separately. If not specified, the destination name is not advertised through the JNDI namespace. This attribute is not dynamically configurable.	Java identifier, unique within the JNDI namespace scope	null
Destination Keys	This attribute provides a list of potential destination keys for sorting the messages that arrive on the destination. They are ordered from most significant to least significant. A key that is based on JMSMessageID can only be the last key. **Note:** If JMSMessageID is not defined in the key, it is implicitly assumed to be the last key and set as “Ascending” (FIFO) for the sort order.	List	Null

Attribute	Description	Range of Values	Default
Enable Store	Flag specifying whether or not the destination uses the persistent store specified by the JMS server. ■ If this flag is enabled, but no persistent store is defined for the JMS Server, then the configuration fails and WebLogic JMS does not boot. ■ If this flag is disabled, then the destination does not support persistent messages. ■ If this flag is set to Default, then the destination uses the persistent store—if one is defined for the JMS server—and supports persistent messaging. This attribute is not dynamically configurable.	false, true, default	default
Template	The JMS template from which the destination is derived. If this attribute is not defined, then the attributes for the destination must be specified as part of the destination. The Template attribute setting per destination is static. The template attributes, however, can be modified dynamically.	Existing JMS template name or none	none

Thresholds & Quotas

Attribute	Description	Range of Values	Default
Bytes Maximum	Maximum number of bytes that may be stored in the destination. A value of -1 specifies that there is no limit on the number of bytes that can be stored in the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through $2^{63} - 1$, Greater than or equal to Bytes Threshold High	-1
Bytes Threshold High	Upper threshold value based on the number of bytes stored in the destination. If the number of bytes exceeds this threshold—and if bytes paging is enabled and a paging store has been configured for the JMS Server—then destination-level bytes paging is started, and a message is logged on the server indicating a high threshold condition. A value of -1 specifies that destination bytes paging and threshold log messages are disabled for the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted. **Note:** Bytes paging cannot be dynamically disabled by resetting the Bytes High Threshold to -1. To disable paging, you could set the Bytes High Threshold to a very large number (max is 2^{63}–1), so that paging would not be triggered.	-1, 0 through $2^{63} - 1$, Less than or equal to Bytes Maximum, >Bytes Threshold Low	-1

Attribute	Description	Range of Values	Default
Bytes Threshold Low	Lower threshold value based on the number of bytes stored in the destination. If the number of bytes falls below this threshold, then destination-level bytes paging is stopped (if paging is occurring), and a message is logged on the server indicating that the threshold condition has cleared. A value of -1 specifies that destination bytes paging and threshold log messages are disabled for the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through $2^{63} - 1$, <Bytes Threshold High	-1

Attribute	Description	Range of Values	Default
Bytes Paging Enabled	Flag for specifying whether or not bytes paging is enabled on the destination. ■ If this attribute is set to False, then destination-level byte paging is explicitly disabled for this destination. ■ If this attribute is set to True, a paging store has been configured for the JMS Server, and both the Bytes Threshold Low and Bytes Threshold High attribute values are greater than -1, then destination-level bytes paging is enabled for this destination. ■ If this attribute is set to Default, then this value inherits the JMS template's value—if a JMS template is specified. If no template is configured for the destination, then the Default value is equivalent to False. **Note:** If server-level bytes paging is enabled, and destination-level paging is disabled for a given destination, than messages on the destination can still be paged if server-level paging is started. However, when destination-level paging is disabled for a given destination, then the destination-level high thresholds will not force the destination to page out messages when they are exceeded.	String True False Default	Default
Messages Maximum	Maximum number of messages that may be stored in the destination. A value of -1 specifies that there is no limit on the number of messages that can be stored in the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through $2^{63} – 1$, Greater than or equal to Bytes Threshold High	-1

Attribute	Description	Range of Values	Default
Messages Threshold High	Upper threshold value based on the number of messages stored in the destination. If the number of messages exceeds this threshold—and if messages paging is enabled and a paging store has been configured for the JMS Server—then destination-level paging is started, and a message is logged on the server indicating a high threshold condition. A value of -1 specifies that destination messages paging and threshold log messages are disabled for the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted. **Note:** Messages paging cannot be dynamically disabled by resetting the Messages High Threshold to -1. To disable paging, you could set the Messages High Threshold to a very large number (max is 2^{63}–1), so that paging would not be triggered.	-1, 0 through $2^{63} – 1$, Less than or equal to Messages Maximum, >Messages Threshold Low	-1
Messages Threshold Low	Lower threshold value based on the number of messages stored in the destination. If the number of messages falls below this threshold, then destination-level messages paging is stopped (if paging is occurring) and a message is logged on server, indicating that the threshold condition has cleared. A value of -1 specifies that destination messaes paging and threshold log messages are disabled for the destination. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through $2^{63} – 1$, <Messages Threshold High	-1

Attribute	Description	Range of Values	Default
Messages Paging Enabled	Flag for specifying whether or not messages paging is enabled on the destination. ■ If this attribute is set to False, then destination-level message paging is explicitly disabled for this destination. ■ If this attribute is set to True, a paging store has been configured for the JMS Server, and both the Message Threshold Low and Message Threshold High attribute values are greater than -1, then destination-level messages paging is enabled for this destination. ■ If this attribute is set to Default, then this value inherits the JMS template's value—if a JMS template is specified. If no template is configured for the destination, then the Default value is equivalent to False. **Note:** If server-level messages paging is enabled, and destination-level paging is disabled for a given destination, than messages on the destination can still be paged if server-level paging is started. However, when destination-level paging is disabled for a given destination, then the destination-level high thresholds will not force the destination to page out messages when they are exceeded.	String True False Default	Default

Overrides

Attribute	Description	Range of Values	Default
Priority Override	Priority assigned to all messages that arrive at the destination, regardless of the priority specified by the message producer. The default value (-1) specifies that the destination will not override the Priority setting. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through 9	-1
Time To Live Override	Time-to-live assigned to all messages that arrive at the destination, regardless of the time-to-live specified by the message producer. The default value (-1) specifies that the destination will not override the Time To Live setting. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through $2^{63} - 1$	-1
Time To Deliver Override	Defines the default delay, in milliseconds, between when a message is produced and when it is made visible on its destination, regardless of the time-to-deliver specified by the producer and/or connection factory. The default value (-1) specifies that the destination will not override the Time To Deliver setting. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through $2^{63} - 1$, or string syntax for a delivery schedule	-1

Attribute	Description	Range of Values	Default
Delivery Mode Override	Delivery mode assigned to all messages that arrive at the destination regardless of the delivery mode specified by the message producer. The default value of `No-Delivery` specifies that the delivery mode will not be overridden. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	Persistent, Non-Persistent, or No-Delivery	No-Delivery

Redelivery

Attribute	Description	Range of Values	Default
Redelivery Delay Override	Defines the delay, in milliseconds, before rolled back or recovered messages are redelivered, regardless of the redelivery-delay specified by the consumer and/or connection factory. The default value (-1) specifies that the destination will not override the Redelivery Delay setting. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through $2^{63} – 1$	-1

Attribute	Description	Range of Values	Default
Redelivery Limit	The number of redelivery tries a message can have before it is placed in the error destination. Depending on whether an error destination is configured, the following occurs when the redelivery limit is reached: ■ If no error destination is configured, or the quota for the error destination would be exceeded, then persistent and non-persistent messages are simply dropped. ■ If an error destination is configured and the error destination is at quota, then an error message is logged and the message is dropped. However, if the message is persistent, it remains in the persistent store. This ensures that a persistent message will be redelivered when WebLogic Server is rebooted. The default value (-1) specifies that the destination will not override the Redelivery Limit setting. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	-1, 0 through $2^{63} - 1$	-1
Error Destination	A destination for messages that have reached their redelivery limit. If the error destination is null, then such messages are simply dropped. This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.	Existing destination or none	none

Multicast

Attribute	Description	Range of Values	Default
Multicast Address	IP address used for multicasting. This address is used to transmit messages to multicast consumers. This attribute is valid for topics only and is not dynamically configurable.	A well-formed IP address (Class D)	N/A
Multicast TTL	Time-to-live value used for multicasting, specifying the number of routers that the message can traverse en route to the consumers. A value of 0 indicates that the message will not traverse any routers, and is limited to one subnet. This value is independent of the `JMSExpirationTime` value. This attribute is valid for topics only and is not dynamically configurable.	0 to 255	0
Multicast Port	IP port used for multicasting. This port is used to transmit messages to multicast consumers. This attribute is valid for topics only and is not dynamically configurable.	1024 to 65535	6001

Monitoring

Monitor All Active JMS Destinations

Attribute	Description	Range of Values	Default Value
Destination	Destination name.		N/A
Server	Associated server name.		N/A
Consumers	Current number of registered message consumers.		N/A
Consumers High	Greatest number of registered message consumers at any given time.		N/A
Consumers Total	Total number of registered message consumers.		N/A
Bytes Current	Current number of bytes stored.		N/A
Bytes Pending	Number of unacknowledged and uncommitted transactional bytes stored.		N/A
Bytes Received	Number of bytes received.		N/A

Attribute	Description	Range of Values	Default Value
Bytes Threshold Time	Amount of time that the destination has spent in a bytes threshold condition since the last time the server was booted.		N/A
Messages	Current number of messages stored at this destination.		N/A
Messages High	Greatest number of messages stored at any given time.		N/A
Messages Pending	Number of unacknowledged and uncommitted transactional messages stored.		N/A
Messages Received	Number of messages received.		N/A
Messages Threshold Time	Amount of time that the destination spent in a message threshold condition.		N/A

For additional information about monitoring JMS Queues, please see Monitor All Active JMS Destinations.

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for user-supplied information.	The value must be an alphanumeric string.	Null

57 Messaging Bridge

A messaging bridge is responsible for transferring messages between two JMS providers. The WebLogic Messaging Bridge feature allows you to configure a store and forward mechanism between any two JMS providers—including separate implementations of WebLogic JMS.

Specifically, you can configure a source/target mapping of a JMS destination (queue or topic) to another JMS destination, and vice versa. Messages that are sent to the source destination are then automatically forwarded to the target destination. The messaging bridge also allows the configuration of quality of service (QOS), message filtering, transaction semantics, and reconnection policies.

For each source/target JMS destination to be mapped, whether it is a WebLogic JMS destination or a 3rd-party JMS provider, you must configure a JMS Bridge Destination instance. If the source/target destination is for a non-JMS messaging product, you must configure a General Bridge Destination instance.

Creating a Messaging Bridge

To configure a messaging bridge, follow these steps:

1. Click to expand the Messaging Bridge node.

2. Click the Bridges node to open the Bridges tab in the right pane.

3. Click the Configure a new Messaging Bridge link in the right pane. A Configuration dialog displays in the right pane showing the tabs associated with configuring a new messaging bridge.

4. On the Configuration General tab, enter values in the Name, Source Destination, Target Destination, Selector, Quality of Service, QOS Degradation Allowed, Maximum Idle Time, Asynchronous Mode Enable, Durability Enabled, and Started attribute fields. For more information about messaging bridge general attributes, see General.

5. Click Create to create an instance of the messaging bridge with the name you specified in the Name field. The new instance is added under the Bridges node in the left pane.

6. Click the Connection Retry tab and change the attribute fields or accept the default values as assigned. For connection retry attribute descriptions, see Connection Retry. Click Apply to save your changes.

7. Click the Transactions tab and change the attribute fields or accept the default values as assigned. For transaction attribute descriptions, see Transactions. Click Apply to save your changes.

8. Click the Targets tab and assign the messaging bridge to servers, clusters, or a migratable target server. See Assigning a Messaging Bridge to One or More Servers, or Clusters, or a Migratable Target for instructions. Click Apply to save your changes.

9. Click Apply to save your changes.

Cloning a Messaging Bridge

1. Click to expand the Bridges node. The Messaging Bridges table displays in the right pane showing all the Messaging Bridge instances.

2. Click the Clone icon in the row of the messaging bridge you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a messaging bridge.

3. Enter values in the attribute fields.

4. Click Clone to create a messaging bridge with the name you specified in the Name field. The messaging bridge is added to the Messaging Bridges table in the right pane.

Deleting a Messaging Bridge

1. Click to expand the Messaging Bridge node. The Messaging Bridges table displays in the right pane showing all the messaging bridge instances.
2. Click the Delete icon in the row of the messaging bridge that you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.
3. Click Yes to delete the messaging bridge, and then click Continue to redisplay the Messaging Bridges table in the right pane. The messaging bridge is deleted from the Messaging Bridges table.

Assigning a Messaging Bridge to One or More Servers, or Clusters, or a Migratable Target

1. In the left pane, click the Bridges node to expand it.
2. Click the Messaging Bridge node to expand it and show the list of messaging bridges defined in your domain.
3. Click the messaging bridge that you want to assign. A dialog displays in the right pane showing the tabs associated with the messaging bridge instance.
4. Click the Targets tab to display the following targeting tabs.
   - Servers—you can select a server or servers where the messaging bridge will be deployed. The messaging bridge will be available on all the selected servers.
   - Clusters—you can select a cluster where the messaging bridge will be deployed. The messaging bridge will be available on all servers in the selected cluster.
   - Migratable Targets—you can select a WebLogic Server migratable target where the messaging bridge will be deployed. When WebLogic Server is first booted, the

messaging bridge is initially available only on the user-preferred server. After that, the bridge can be migrated to another server listed in the migratable target

5. On the Servers, Clusters, or Migratable Targets tabs, assign the messaging bridge, as follows:
   a. In the Available column, select one or more targets to assign to the messaging bridge.
   b. Move the targets into the Chosen column by clicking the mover control.
   c. Click Apply to save your assignments.

Adding a Note to a Messaging Bridge

1. In the left pane, click the Bridges node to expand it.
2. Click the Messaging Bridge node to expand it and show the list of messaging bridges defined in your domain.
3. Click the messaging bridge that you want to add a note to. A dialog displays in the right pane showing the tabs associated with the messaging bridge instance.
4. Click the Notes tab. Type a note in the Notes field.
5. Click Apply to save your changes.

Stopping and Restarting a Messaging Bridge

To temporarily suspend a messaging bridge, follow these steps:

1. Click to expand the Messaging Bridge node.
2. Select the messaging bridge instance that you want to stop.

3. On the Configuration General tab, clear the Start check box to stop the bridge.
4. To restart the bridge, select the Start check box.

Monitoring All Active Messaging Bridges

1. Click the server node in the left pane.
2. Select a specific server in the left pane. A dialog displays in the right pane showing the tabs associated with the server instance
3. Select the Services tab.
4. Select the Bridge tab.
5. Click the Monitor all Messaging Bridge Runtimes text link. A table displays showing all active messaging bridges for the server.

Configuring the Execute Thread Pool Size

You can configure the default execute thread pool size for your messaging bridges. For example, you may want to increase or decrease the default size to reduce competition from the WebLogic Server default thread pool. Entering a value of -1 disables this thread pool and forces the bridge to use the WebLogic Server default thread pool.

1. Click the Servers node in the left pane to expand it.
2. Select a specific server instance.
3. Select the Services tab in the right pane.
4. Select the Bridge tab.
5. Enter a new value in the Messaging Bridge Thread Pool Size field.
6. Click Apply to save your changes.

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	A messaging bridge name that is unique across the WebLogic Server domain. This attribute is not dynamically configurable.	Java identifier, unique within specific domain.	MyMessaging Bridge [-*n*]
Source Destination	The source destination *from which* messages are received by the messaging bridge. This attribute is not dynamically configurable.	Existing source bridge destination name.	Null
Target Destination	The target destination *to which* messages are sent from the messaging bridge. This attribute is not dynamically configurable.	Existing target bridge destination name.	Null

Attribute	**Description**	**Range of Values**	**Default Value**
Selector	Allows you to filter the messages that are sent across the messaging bridge. Only messages that match the selection criteria are sent across the messaging bridge. For queues, messages that do not match the selection criteria are left behind and accumulate in the queue. For topics, messages that do not match the connection criteria are dropped. **Note:** For more information on using selectors to filter messages, see "Developing a WebLogic JMS Application" in *Programming WebLogic JMS*. This attribute is not dynamically configurable.	The value must be an alphanumeric string.	None

Attribute	Description	Range of Values	Default Value
Quality of Service	Defines the guarantees for forwarding a message across a messaging bridge. The valid qualities of service are: *Exactly-once*—Each message will be sent exactly once. This is the highest quality of service. *Atmost-once*—Each message is sent at most one time. Some messages may not be delivered to the target destination. *Duplicate-okay*—Each message is sent at least one time. Duplicate messages can be delivered to the target destination. **Note:** In order to use *Exactly-once*, the source and target connection factories must be configured to use `XAConnectionFactory`. Also, the `jms-xa-adp.rar` adapter must be deployed and identified in the Adapter JNDI Name attribute for the source and target destinations. This attribute is not dynamically configurable.	Exactly-once, Atmost-once, and Duplicate-okay	Exactly-once

QOS Degradation Allowed	When selected, the messaging bridge will automatically degrade the requested QOS (quality of service) when the configured one is not available. If this occurs, a message is delivered to the WebLogic startup window (or log file). If this option is not selected (false), and the messaging bridge cannot satisfy the requested QOS, it will result in an error and the messaging bridge will not start.
	This attribute is not dynamically configurable.
		Boolean
		Enabled = selected
		Disabled = not selected
			Not selected

Attribute	**Description**	**Range of Values**	**Default Value**
Maximum Idle Time (seconds)	For bridges running in a asynchronous mode, this is the maximum amount of time, in seconds, the messaging bridge will sit idle before checking the health of its connections. For bridges running in a synchronous mode, this dictates the amount of time the messaging bridge can block on a receive call if no transaction is involved.	0 through 2^{31} - 1	60

Asynchronous Mode Enabled	Defines whether a messaging bridge works in asynchronous mode. Messaging bridges that work in asynchronous mode (true) are driven by the source target. The messaging bridge listens for messages and forwards them as they arrive. When the value is false, the bridge is forced to work in synchronous mode, even if the source supports asynchronous receiving.
	Note: For a messaging bridge with a QOS of *Exactly-once* to work in asynchronous mode, the source destination has to support the `MDBTransaction` interface described in the `weblogic.jms.extensions` Javadoc. Otherwise, the bridge will automatically switch to synchronous mode if it detects that `MDBTransactions` are not supported by the source destination. For more information about `MDBTransactions`, see "Using Message-Driven Beans" in *Programming WebLogic EJB*.
	This attribute is not dynamically configurable.
		Boolean
		Enabled = selected
		Disabled = not selected
			Selected

Attribute	Description	Range of Values	Default Value

Durability Enabled	A messaging bridge that has this attribute set to *true* creates a durable subscription for the source destination—providing the source destination type is a topic. This allows the source JMS implementation to save messages that are sent to it when the bridge is not running. The bridge will then forward these messages to the target destination once it is restarted. If the value is set to *false*, messages that are sent to the source JMS topic while the bridge is down cannot be forwarded to the target destination.
	Note: This attribute is used only for JMS topics or when a destination has similar characteristics.
	Note: If a bridge must be taken permanently offline, you must delete any durable subscriptions that use the bridge. For information on deleting durable subscribers, see "Deleting Durable Subscriptions" in *Programming WebLogic JMS*.
	This attribute is not dynamically configurable.
		Boolean
		Enabled = selected
		Disabled = not selected
			Not selected

Attribute	Description	Range of Values	Default Value

Started	Defines the initial state of the messaging bridge at run-time (that is, the state when it boots). The default value is on (*true*). By clearing this check box (*false*), you can stop the messaging bridge. Conversely, reselecting the check box, restarts the bridge.
	Note: This does not indicate the run-time state of the bridge. For information on monitoring a bridge's state, see "Monitoring All Active Messaging Bridges" on page 57-23
		Boolean
		Enabled = selected
		Disabled = not selected
			Selected

Connection Retry

Attribute	Description	Range of Values	Default Value
Minimum Delay (seconds)	The minimum delay, in seconds, between reconnection attempts. When a messaging bridge boots and cannot connect to a destination, or a connection is lost and the messaging bridge is first attempting to reconnect, it will attempt to reconnect in this specified amount of seconds. This attribute is not dynamically configurable.	0 through 2^{31} - 1	15
Incremental Delay (seconds)	The delay increment, in seconds, between reconnection attempts. Each time a bridge fails to reconnect, it will add this amount of seconds to the delay before it makes its next reconnection attempt. This attribute is dynamically configurable.	0 through 2^{31} - 1	5
Maximum Delay (seconds)	The maximum delay, in seconds, between reconnection attempts. Each reconnection attempt is delayed further by the Incremental Delay amount of seconds, but it is never delayed by more than this value. This attribute is dynamically configurable.	0 through 2^{31} - 1	60

Transactions

Attribute	Description	Range of Values	Default Value
Transaction Timeout	Defines the number of seconds the transaction manager will wait for each transaction before timing it out. Transaction timeouts are used when a bridge’s quality of service requires transactions. This attribute is not dynamically configurable.	0 through 2^{31} - 1	30
Batch Size	Defines the number of messages that the messaging bridge transfers within one transaction. This attribute only applies to bridges that work in synchronous mode and whose quality of service require two-phase transactions. This attribute is dynamically configurable.	0 through 2^{31} - 1	10
Batch Interval (milliseconds)	Defines the maximum time, in milliseconds, that the bridge will wait before sending a batch of messages in one transaction, regardless of whether the Batch Size amount has been reached or not. This attribute only applies to bridges that work in synchronous mode and whose quality of service require two-phase transactions. This attribute is dynamically configurable.	0 through 2^{31} - 1	-1

Targets

Attribute	Description	Range of Values	Default Value
Migratable Targets	Defines a WebLogic Server migratable target where the messaging bridge will be deployed. When WebLogic Server is first booted, the messaging bridge will initially be available only on the *user-preferred* server. Afterwards, the bridge can be migrated to another server in the migratable target using either the Administration Console or the command-line tool.	List of available and chosen targets.	No targets are chosen.
Clusters	Defines a WebLogic Server cluster where the messaging bridge will be deployed. The messaging bridge will be available on all servers in the selected cluster.	List of available and chosen targets.	No targets are chosen.
Servers	Defines the WebLogic Servers where the messaging bridge will be deployed. The messaging bridge will be available on all the selected WebLogic Servers.	List of available and chosen targets	No targets are chosen.

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for user-supplied information.	The value must be an alphanumeric string.	Null

58 JMS Bridge Destination

For each source/target JMS destination to be mapped, whether it is a WebLogic JMS implementation or a 3rd-party JMS provider, you must configure a JMS Bridge Destination instance.

Creating a JMS Bridge Destination

1. Click to expand the Messaging Bridge node.
2. Click the JMS Bridge Destinations node to open the JMS Bridge Destinations tab in the right pane.
3. In the right pane, click the Configure a New JMS Bridge Destination link. A Configuration dialog displays in the right pane showing the tabs associated with configuring a new JMS bridge destination.
4. On the Configuration tab, enter values in the Name, Adapter JNDI Name, Adapter Class Path, Connection URL, Initial Context Factory, Connection Factory JNDI Name, Destination JNDI Name, and Destination Type attribute fields. For more information about these general attributes, see Configuration.
5. Optionally, fill out the User Name and User Password attributes. For more information about these attributes, see Configuration.

   Note: All operations done to the specified destination are done using that user name and password. Therefore, the User Name/Password for the source and target destinations must have permission to access the underlying destinations in order for the messaging bridge to work.

6. Click Create to create an instance the bridge destination with the name you specified in the Name field. The new instance is added under the JMS Bridge Destination node in the left pane.
7. When you finish defining attributes for a source JMS bridge destination, repeat these steps to configure a target JMS bridge destination, or vice versa.

For what’s next in configuring a messaging bridge, see “Messaging Bridge” on page 57-19.

Cloning a JMS Bridge Destination

1. Click to expand the Messaging Bridge node.
2. Click to expand the JMS Bridge Destinations node. The JMS Bridge Destinations table displays in the right pane showing all the JMS bridge destination instances.
3. Click the Clone icon in the row of the JMS bridge destination instance you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a JMS bridge destination.
4. Enter values in the attribute fields.
5. Click Clone to create a JMS bridge destination with the name you specified in the Name field. The JMS bridge destination is added to the JMS Bridge Destination table in the right pane.

Deleting a JMS Bridge Destination

1. Click to expand the Messaging Bridge node.
2. Click to expand the JMS Bridge Destinations node. The JMS Bridge Destinations table displays in the right pane showing all the JMS bridge destination instances.

3. Click the Delete icon in the row of the JMS bridge destination that you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.
4. Click Yes to delete the JMS bridge destination, and then click Continue to redisplay the JMS Bridge Destinations table in the right pane. The bridge destination is deleted from the JMS Bridge Destinations table.

Adding a Note to a JMS Bridge Destination

1. In the left pane, click the Bridges node to expand it.
2. Click the JMS Bridge Destinations node to expand it and show the list of JMS bridge destinations defined in your domain.
3. Click the JMS bridge destination that you want to add a note to. A dialog displays in the right pane showing the tabs associated with the JMS bridge destination instance.
4. Click the Notes tab. Type a note in the Notes field.
5. Click Apply to save your changes.

Configuration

Configuration

Attribute	Description	Range of Values	Default Value
Name	A bridge destination name that is unique across a WebLogic Server domain. This attribute is not dynamically configurable.	Java identifier, unique within specific domain.	MyJMS Bridge Destination [-*n*]
Adapter JNDI Name	The JNDI name of the adapter. This is the string defined in the adapter's descriptor file, and is used to bind the adapter in WebLogic Server JNDI. A bridge destination must supply the JNDI name of the adapter used to communicate with the specified destination. This attribute is not dynamically configurable.	Java identifier, unique within JNDI namespace.	eis.jms.WLS Connection FactoryJNDI XA

Attribute	Description	Range of Values	Default Value
Adapter Classpath	The `CLASSPATH` of the bridge destination, which is mainly used to connect to a different release of WebLogic JMS. When connecting to a destination that is running on WebLogic Server 6.0 or earlier, the bridge destination must supply a `CLASSPATH` that indicates the locations of the classes for the earlier WebLogic Server implementation. This attribute is not dynamically configurable.	The value must be an alphanumeric string.	Null
Connection URL	The URL of the JNDI provider used to look up the connection factory and destination. This attribute is not dynamically configurable.	The value must be an alphanumeric string.	Null
Initial Context Factory	The factory used to get the JNDI context. This attribute is not dynamically configurable.	The value must be an alphanumeric string.	weblogic.jndi.WLInitialContextFactory
Connection Factory JNDI Name	The JMS connection factory used to create a connection. **Note:** In order to use the *Exactly-once* QOS, the connection factory has to be a XAConnection Factory. This attribute is not dynamically configurable.	Java identifier, unique within JNDI namespace.	Null
Destination JNDI Name	The JNDI name of the JMS destination. This attribute is not dynamically configurable.	Java identifier, unique within JNDI namespace.	Null

Attribute	Description	Range of Values	Default Value
Destination Type	Either a Queue or Topic. This attribute is not dynamically configurable.	Queue or Topic	Queue
User Name	The optional user name that the messaging bridge will give to the bridge adapter. **Note:** All operations done to the specified destination are done using the User Name and User Password. Therefore, the User Name/Password for the source and target bridge destinations must have permission to access the underlying source and target destinations in order for the messaging bridge to work. This attribute is not dynamically configurable.	The value must be an alphanumeric string.	Null
User Password	The password that the messaging bridge will give to the bridge adapter. This attribute is not dynamically configurable.	The value must be an alphanumeric string.	Null

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for user-supplied information.	The value must be an alphanumeric string.	Null

59 General Bridge Destination

For non-JMS messaging products, you need to configure a generic BridgeDestination instance for each source and target destination to be mapped by a messaging bridge.

Note: Although WebLogic JMS includes a provisional General Bridge Destination framework for accessing non-JMS messaging products, WebLogic Server does not provide supported adapters for such products. Therefore, you must obtain a custom adapter from a third-party OEM vendor and consult their documentation for configuration instructions. You could also contact BEA Professional Services more information about obtaining a custom non-JMS adapter.

Creating a General Bridge Destination

1. Expand the Messaging Bridge node.
2. Click the General Bridge Destinations node to open the General Bridge Destinations tab in the right pane.
3. In the right pane, click the Configure a New General Bridge Destination link. A Configuration dialog shows the tabs associated with configuring a new general bridge destination.
4. On the Configuration General tab, enter values in the Name, Adapter JNDI Name, Adapter Class Path, and Properties attribute fields. For more information about these general attributes, see Configuration.

5. Optionally, fill out the User Name and User Password attributes. For more information about these attributes, see Configuration.

   Note: All operations done to the specified destination are done using that user name and password. Therefore, the User Name/Password for the source and target destinations must have permission to access the underlying destinations in order for the messaging bridge to work.

6. Click Create to create an instance the general bridge destination with the name you specified in the Name field. The new instance is added under the General Bridge Destination node in the left pane.

7. When you finish defining attributes for a source general bridge destination, repeat these steps to configure a target general bridge destination, or vice versa.

For what's next in configuring a messaging bridge, see "Messaging Bridge" on page 57-19.

Cloning a General Bridge Destination

1. Expand the Messaging Bridge node and the General Bridge Destinations node.

2. The General Bridge Destinations table displays in the right pane showing all the general bridge destination instances.

3. Click the Clone icon in the row of the general bridge destination that you want to clone. A dialog shows the tabs associated with cloning a general bridge destination.

4. Enter values in the attribute fields.

5. Click Clone to create a general bridge destination with the name you specified in the Name field. The general bridge destination is added to the General Bridge Destination table in the right pane.

Deleting a General Bridge Destination

1. Expand the Messaging Bridge node and the General Bridge Destinations node.
2. The General Bridge Destinations table displays in the right pane showing all the general bridge destination instances.
3. Click the Delete icon in the row of the general bridge destination instance that you want to delete. A dialog prompts you to confirm your deletion request.
4. Click Yes to delete the general bridge destination, and then click Continue to redisplay the General Bridge Destinations table in the right pane. The general bridge destination is deleted from the General Bridge Destinations table.

Adding a Note to a General Bridge Destination

1. Expand the Message Bridge node to expand it.
2. Click the General Bridge Destinations node to expand it and show the list of general bridge destinations defined in your domain.
3. Click the general bridge destination that you want to add a note to. A dialog shows the tabs associated with the general bridge destination instance.
4. Click the Notes tab. Type a note in the Notes field.
5. Click Apply to save your changes.

Configuration

Configuration

Attribute	Description	Range of Values	Default Value
Name	A bridge destination name that is unique across a WebLogic domain. For the source destination, you may want to change the default name to "Source Bridge Destination". For the target, use something like "Target Bridge Destination". Once configured, these names are listed as options in the Source Destination and Target Destination attributes on the Bridges --> General tab. This attribute is not dynamically configurable.	Java identifier, unique within specific domain.	MyGeneral Bridge Destination [-*n*]
Adapter JNDI Name	A bridge destination must supply the JNDI name of the adapter used to communicate with the specified destination. WebLogic Server does not provide adapters for non-JMS messaging products. Therefore, you must use a specialized adapter from a third-party OEM vendor or contact BEA Professional services. This attribute is dynamically configurable.	Java identifier, unique within JNDI namespace	eis.jms.WLS Connection FactoryJNDI XA

Attribute	Description	Range of Values	Default Value
Adapter Classpath	The CLASSPATH of the bridge destination, which is mainly used to connect to a different release of WebLogic JMS. When connecting to a destination that is running on WebLogic Server 6.0 or earlier, the bridge destination must supply a CLASSPATH that indicates the locations of the classes for the earlier WebLogic Server implementation. When connecting to a third-party messaging product, you must supply the product's CLASSPATH in the WebLogic Server CLASSPATH. This attribute is dynamically configurable.	The value must be an alphanumeric string.	Null

Properties (key=value)	Specifies all the properties defined for the bridge destination. Each property must be separated by a semicolon (for example, `DestinationJNDIName=myTopic;DestinationType=topic;`).
	For non-JMS messaging products that use adapters provided by a third-party OEM vendor, you should consult the vendor's documentation for property configuration instructions.
	The following properties are required for all JMS implementations:
	`ConnectionURL` – URL used to establish a connection to the destination.
	`InitialContextFactory` – Factory used to get the JNDI context.
	`ConnectionFactoryJNDIName` – the JMS connection factory used to create a connection.
	`DestinationJNDIName` – JNDI name of the JMS destination.
	`DestinationType` – Queue or Topic.
	This attribute is dynamically configurable.
		The value must be an alphanumeric string proceeded by the following property identifiers:
		`ConnectionURL=`
		`InitialContextFactory=`
		`ConnectionFactoryJNDIName=`
		`DestinationJNDIName=`
		`DestinationType=`
			Null

Attribute	Description	Range of Values	Default Value
User Name	The optional user name that the messaging bridge will give to the bridge adapter. **Note:** All operations done to the specified destination are done using the User Name and User Password. Therefore, the User Name/Password for the source and target bridge destinations must have permission to access the underlying source and target destinations in order for the messaging bridge to work. This attribute is dynamically configurable.	The value must be an alphanumeric string.	Null
User Password	The password that the messaging bridge will give to the bridge adapter. This attribute is dynamically configurable.	The value must be an alphanumeric string.	Null

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for user-supplied information.	The value must be an alphanumeric string.	Null

60 JNDIBinding

WebLogic Server EJBs, RMI, JMS, JDBC, and Mail objects can be bound into the JNDI tree.

You can use the Administration Console to bind JMS connection factories or JDBC Data Source objects to the JNDI tree when you create those objects.

For more information on JNDI, see Managing JNDI in the Administration Guide and Programming WebLogic JNDI.

61 JNDI Context

You can view the JNDI tree from the Administration Console.

1. Right click on the node for your WebLogic Server in the right pane. This opens a pop-up menu.
2. Click View JNDI Tree. The information appears in the right pane of a new window.

For more information on JNDI, see Managing JNDI in the Administration Guide and Programming WebLogic JNDI.

62 Jolt

These procedures describe how to use the Administration Console to set the attributes for configuring and managing Jolt connection pools.

Configure a Jolt Connection Pool

1. Click the Jolt node in the left pane. The Jolt Connection Pools table displays in the right pane showing all the Jolt connection pools defined in the domain.
2. Click the Configure a New Jolt Connection Pool text link. A dialog displays in the right pane showing the tabs associated with configuring a new connection pool.
3. Enter values in the Name, Minimum Pool Size, Maximum Pool Size, and the Recv Timeout attribute fields. Click the Security Context Enabled checkbox to enable security context.
4. Click Create to create a connection pool instance with the name you specified in the Name field. The new instance is added under the Jolt node in the left pane.
5. Click the Addresses and the User tabs individually and change the attribute fields or accept the default values as assigned.
6. Click Apply to save any changes you made.

Clone a Jolt Connection Pool

1. Click the Jolt node in the left pane. The Jolt Connection Pools table displays in the right pane showing all the Jolt connection pools defined in the domain.
2. Click the Clone icon in the row of the connection pool you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a new connection pool.
3. Enter values in the Name, Minimum Pool Size, Maximum Pool Size, and the Recv Timeout attribute fields. Click the Security Context Enabled checkbox to enable security context.
4. Click Create to create a connection pool instance with the name you specified in the Name field. The new instance is added under the Jolt node in the left pane.
5. Click the Addresses and the User tabs individually and change the attribute fields or accept the default values as assigned.
6. Click Apply to save any changes you made.

Delete a Jolt Connection Pool

1. Click the Jolt node in the left pane. The Jolt Connection Pools table displays in the right pane showing all the Jolt connection pools defined in the domain.
2. Click the Delete icon in the row of the connection pool you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click Yes to delete the connection pool. The connection pool icon under the Jolt node is deleted.

Assign a Jolt Connection Pool

1. Click the instance node in the left pane under Jolt to select a connection pool for assignment. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Targets tab.
3. Complete the following steps for the Servers, Groups, and Clusters tabs:
   a. Select one or more targets in the Available column that you want to assign to the Jolt connection factory.
   b. Click the mover control to move the targets you selected to the Chosen column.
   c. Click Apply to save your assignments.

Monitor All Instances of a Jolt Connection Pool

1. Click the instance node in the left pane under Jolt to select a connection pool for monitoring. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Monitoring tab.
3. Click the Monitor All Instances of a Jolt Connection Pool text link. The Jolt Connection Pools table displays in the right pane showing all connection pool instances deployed on a server.

63 Jolt Connection Pool

Configuration

General

Attribute	Description	Range of Values	Default
Name	This attribute returns the name of this connection pool.	String	MyJolt Connection Pool
Minimum Pool Size	This attribute allows the user to set the minimum pool size for this connection pool.	Integer	0
Maximum Pool Size	This attribute allows the user to set the maximum pool size for this connection pool.	Integer	1
Recv Timeout	This attribute allows the user to set the amount of time the client should wait to receive a response before timing out.	Integer	0

Attribute	Description	Range of Values	Default
Security Context Enabled	This attribute enables or disables the security context for this connection pool.	Boolean Enabled = Selected Not Enabled = Not Selected	Not Selected

Addresses

Attribute	Description	Range of Values	Default
Primary Addresses	This attribute allows the user to set the primary addresses	Valid Addresses	Null
Failover Addresses	This attribute allows the user to set the addresses to be used in the event of a failure.	Valid Addresses	Null

User

Attribute	Description	Range of Values	Default
User Name	This attribute sets the user name for this connection pool.	Valid user name as set by the administrator.	Null
User Role	This attribute sets the user role for this connection pool.	Valid user role as set by the administrator.	Null
User Password	This attribute sets the user password for this connection pool.	Valid user password as set by the administrator.	Null
Application Password	This attribute allows the user to set the application password for this connection pool.	Valid password	Null

Targets

Servers

Attribute	Description	Range of Values	Default
Target Servers	This attribute allows the user to set the target servers for the connection pool.	Valid server names	Null

Clusters

Attribute	Description	Range of Values	Default
Target Clusters	This attribute allows the user to set the target clusters for the connection pool.	Valid server names	Null

Notes

Attribute	Description	Range of Values	Default
Notes	This attribute provides space for optional user supplied information.	The value must be an alphanumeric string.	Null

64 Jolt Connection Pool Runtime

By default, this pane allows the user to sort the objects by the following criteria:

- Server
- Machine
- Name
- Pool state
- Connections
- MaxCapacity

65 Web Application Deployment Descriptor Editor Help

Configuring the JSP Descriptor

The JSP Descriptor configures JSP-related parameters.

1. If it does not exist in the left pane, create the WebApp Ext node:

   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

   b. Select Configure a new WebAppExtDescriptor

   c. In the right pane, enter a description of the Web Application in the Description field (optional).

   d. Enter the version of WebLogic Server that your application is running on (optional).

   e. Click Create.

   f. Expand the WebApp Ext node in the left panel.

2. Click on the JSP Descriptor node in the left panel.

3. Fill in the fields in the right panel according to the information in JSP Descriptor table.

4. Click Apply.

Configuring Security Role Assignments

Security role assignments map a security role to one or more principals. The principals must be defined in your security realm.

1. If it does not exist in the left pane, create the WebApp Ext node:
    a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)
    b. Select Configure a new WebAppExtDescriptor
    c. In the right pane, enter a description of the Web Application in the Description field (optional).
    d. Enter the version of WebLogic Server that your application is running on (optional).
    e. Click Create.
    f. Expand the WebApp Ext node in the left panel.
2. Right click on the Security Role Assignment node in the left panel and select Configure a new SecurityRoleAssignment.
3. Select a security role from the Role drop-down list.
4. Add one or more Principal Names to the text box, one per line. The principals must be valid in your security realm.
5. Click Create.

Configuring Character Set Parameters

You can define codeset behavior for non-unicode operations by configuring Character Set Parameters. You can also specify a mapping of Java character sets to IANA character sets. For more information, see Determining the Encoding of an HTTP

Request at `http://e-docs.bea.com/wls/docs61/webapp/components.html#encoding-http`, and Mapping IANA Character Sets to Java Character Sets at `http://e-docs.bea.com/wls/docs61/webapp/components.html#map-iana`.

1. If it does not exist in the left pane, create the WebApp Ext node:

   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

   b. Select Configure a new WebAppExtDescriptor

   c. In the right pane, enter a description of the Web Application in the Description field (optional).

   d. Enter the version of WebLogic Server that your application is running on (optional).

   e. Click Create.

   f. Expand the WebApp Ext node in the left panel.

2. Right click on the WebApp Ext node in the left panel and select Configure a new Charset Params.

3. Create Input Charset Descriptors

   a. Expand the Charset Params node.

   b. Right click on the Input Charset Descriptors node and select Configure a new InputCharsetDescriptor.

   c. Enter the path that applies to this Charset Descriptor in the Resource Path field.

   d. Enter a valid Java character set name in the Java Charset Name field.

   e. Click Create

4. Create a Character Set Mapping

   a. Expand the Charset Params node.

   b. Right click on the Charset Mappings node and select Configure a new CharsetMapping.

   c. Enter a valid IANA character set name in the IANA Charset Name field.

d. Enter a valid Java character set name in the Java Charset Name field.

e. Click Create

Configuring a Reference Descriptor

The Reference Descriptor maps the JNDI name of a server resource to a name used in the Web application. The Resource Description panel maps a resource, for example, a DataSource, to its JNDI name. The Ejb Reference panel maps an EJB to its JNDI name.

1. If it does not exist in the left pane, create the WebApp Ext node:

   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

   b. Select Configure a new WebAppExtDescriptor

   c. In the right pane, enter a description of the Web Application in the Description field (optional).

   d. Enter the version of WebLogic Server that your application is running on (optional).

   e. Click Create.

   f. Expand the WebApp Ext node in the left panel.

2. Right click on the WebApp Ext node in the left panel and select Configure a new Reference Descriptor.

3. Create Resource Descriptions. You can create one or more Resource Descriptions.

   a. Expand the Resource Descriptor node.

   b. Right click on the Resource Descriptions node and select Configure a new ResourceDescription.

   c. Enter the name of an object from the JNDI tree in the Jndi Name field.

   d. Select a Resource Reference from the Resource Reference drop-down list. The Resource Reference should be already configured under the Resource Refs tab.

   e. Click Create

4. Create an EJB Reference Description

   a. Expand the Resource Descriptor node.

   b. Right click on the Ejb Reference Descriptions node and select Configure a new EjbReferenceDescription.

   c. Enter the name of an object from the JNDI tree in the Jndi Name field.

   d. Select an EJB Reference from the Ejb Reference drop-down list. The EJB Reference should be already configured under the Ejb refs tab.

   e. Click Create

Configuring a Container Descriptor

Enable Check Auth on Forward when you want to require authentication of forwarded requests from a servlet or JSP.

1. If it does not exist in the left pane, create the WebApp Ext node:

   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

   b. Select Configure a new WebAppExtDescriptor

   c. In the right pane, enter a description of the Web Application in the Description field (optional).

   d. Enter the version of WebLogic Server that your application is running on (optional).

   e. Click Create.

   f. Expand the WebApp Ext node in the left panel.

2. Right click on the WebApp Ext node and select Configure a new Container Descriptor.

3. To enable authentication of forwarded requests, check the Check Auth on Forward Enabled box.

4. To configure redirect behavior, check the redirect-with-absoute-url box.

5. Click Create.

Web App Descriptor

Attribute	Description	Range of Values	Default Value
Description	A text description of the servlet.	String	none
Display Name	This element is not used by WebLogic Server.	String	none
Large Icon File Name	Specifies the location for a large (32x32 pixel) `.gif` or `.jpg` image used to represent the Web application in a GUI tool. Currently, this element is not used by WebLogic Server.	String	none
Small Icon File Name	Specifies the location for a small (16x16 pixel) `.gif` or `.jpg` image used to represent the Web application in a GUI tool. Currently, this is not used by WebLogic Server.	String	none
Distributable	This element is not used by WebLogic Server.	Empty	none

Filters

Attribute	Description	Range of Values	Default Value
Filter Name	The name for this filter	String	MyFilter
Description	A text description of the servlet.	String	none
Display Name	This element is not used by WebLogic Server.	String	none
Small Icon File Name	This element is not used by WebLogic Server.	String	none
Large Icon File Name	This element is not used by WebLogic Server.	String	none
Filter Class	The fully-qualified class name of the filter.	Class name	none

Filter Mappings

Attribute	Description	Range of Values	Default Value
Filter	The name of the filter to which you are mapping a URL pattern. This name corresponds to the name you assigned a filter in a `<filter-name>` element.	String	none

Url Pattern	Describes a pattern used to resolve URLs. The portion of the URL after the `http://host:port` + `WebAppName` is compared to the `<url-pattern>` by WebLogic Server. If the patterns match, the filter mapped in this element will be called.
	Example patterns:
	`/soda/grape/*`
	`/foo/*`
	`/contents`
	`*.foo`
	The URL must follow the rules specified in Section 10 of the Servlet 2.2 Specification.
		String	none

Attribute	Description	Range of Values	Default Value
Servlet	The name of a servlet which, if called, causes this filter to execute.	A valid servlet name defined in this deployment descriptor.	none

Listeners

Attribute	Description	Range of Values	Default Value
Listener Class Name	Name of the class that responds to a Web Application event.	Class name	none

Servlets

Attribute	Description	Range of Values	Default Value
Description	A text description of the servlet.	String	None
Display Name	This element is not used by WebLogic Server.	String	None
Small Icon File Name	This element is not used by WebLogic Server.	String	None
Large Icon File Name	This element is not used by WebLogic Server.	String	None
Servlet Name	Defines the canonical name of the servlet, used to reference the servlet definition elsewhere in the deployment descriptor.	String	MyServlet
Servlet Class	The fully-qualified class name of the servlet. You may use only one of either the `<servlet-class>`tags or `<jsp-file>` tags in your servlet body.	Class name	None

Attribute	Description	Range of Values	Default Value
Jsp File	The full path to a JSP file within the Web application, relative to the Web application root directory. You may use only one of either the `<servlet-class>`tags or `<jsp-file>` tags in your servlet body.	String	None
Load On Startup	WebLogic Server initializes this servlet when WebLogic Server starts up. The optional contents of this element must be a positive integer indicating the order in which the servlet should be loaded. Lower integers are loaded before higher integers.	Positive Integer If no value is specified, or if the value specified is not a positive integer, WebLogic Server can load the servlet in any order in the startup sequence.	0

Parameters

Attribute	Description	Range of Values	Default Value
Description	Text description of the initialization parameter.		MyError Page
Param Name	Defines the name of this initialization parameter.	String	None

Attribute	Description	Range of Values	Default Value
Param Value	Defines a `String` value for this initialization parameter.	String	None

Security Role Refs

Attribute	Description	Range of Values	Default Value
Description	Text description of the role.	String	MyError Page
Role Name	Defines the name of the security role or principal that is used in the servlet code.	String	None
Role Link	Defines the name of the security role.	String - a security role that is defined in a `<security-role>` element later in the deployment descriptor.	None

Servlet Mappings

Attribute	Description	Range of Values	Default Value
Servlet	The name of the servlet to which you are mapping a URL pattern. This name corresponds to the name you assigned a servlet in a `<servlet>` declaration tag.	A declared servlet.	none

URL Pattern	Describes a pattern used to resolve URLs. The portion of the URL after the `http://host:port` + `WebAppName` is compared to the `<url-pattern>` by WebLogic Server. If the patterns match, the servlet mapped in this element will be called.
	Example patterns:
	`/soda/grape/*`
	`/foo/*`
	`/contents`
	`*.foo`
	The URL must follow the rules specified in Section 10 of the Servlet 2.2 Specification.
		String	MyServlet Mapping

Mime Mappings

Attribute	Description	Range of Values	Default Value
Mime Type	A string describing the defined mime type, for example: `text/plain.`	String	MyMime Mapping
Extension	A string describing an extension, for example: `txt.`	String	None

Session Config

Attribute	Description	Range of Values	Default Value
Session Timeout	The number of minutes after which sessions in this Web Application expire. The value set in this element overrides the value set in the `TimeoutSecs` parameter of the Session Descriptor in the WebLogic-specific deployment descriptor `weblogic.xml`, unless one of the special values listed here is entered.	Maximum value: Integer.MAX_VALUE ÷ 60 Special values: ■ -2 = Use the value set by `TimeoutSecs` in Session Descriptor ■ -1 = Sessions do not timeout. The value set in Session Descriptor is ignored.	-2

Welcome Files

Attribute	Description	Range of Values	Default Value
Welcome File	File name to use as a default welcome file, such as index.html. You may specify one or more welcome files.	String	None

Error Pages

Attribute	Description	Range of Values	Default Value
Error Code	A valid HTTP error code, for example `404.`	String	MyError Page
Exception Type	A fully-qualified class name of a Java exception type, for example `java.lang.`	String	None
Location	The location of the resource to display in response to the error. Must include a leading `/`. For example `/myErrorPg.html.`	String	None

Tag Libs

Attribute	Description	Range of Values	Default Value
URI	Describes a URI, relative to the location of the web.xml document, identifying a Tag Library used in the Web application. If the URI matches the URI string used in the taglib directive on the JSP page, this taglib is used.	String	MyTag Lib
Location	Gives the file name of the tag library descriptor relative to the root of the Web application. It is a good idea to store the tag library descriptor file under the `WEB-INF` directory so it is not publicly available over an HTTP request.	String	None

Resource Env Refs

The resource-env-ref element contains a declaration of an component's reference to an administered object associated with a resource in the component's environment.

Attribute	Description	Range of Values	Default Value
Description	A text description.	String	none
Ref Name	The name of a resource environment reference; its value is the environment entry name used in code.	String	MyResource Env Ref
Ref Type	The type of a resource environment reference.	String	none

Resource Refs

Attribute	Description	Range of Values	Default Value
Description	A text description.	String	none
Ref Name	The name of the resource used in the JNDI tree. Servlets in the Web application use this name to look up a reference to the resource.	String	MyResource Ref
Ref Type	The Java type of the resource that corresponds to the reference name. Use the full package name of the Java type.	Java type	none
Auth	Used to control the resource sign on for security.	`APPLICATION:` The application component code performs resource sign on programmatically. `CONTAINER:` WebLogic Server uses the security context established with login config element.	none
Sharing Scope	Specifies whether connections obtained through the given resource manager connection factory reference can be shared.	`Sharable` `Unsharable`	`Sharable`

Security Constraints

Attribute	Description	Range of Values	Default Value
Display Name	Name of this constraint.	String	MySecurity Constraint

Auth Constraint

Defines which groups or principals have access to the collection of web resources defined in this security constraint.

Attribute	Description	Range of Values	Default Value
Description	A text description of this security constraint.	String	none
Role Name	Defines which security roles can access resources defined in this security-constraint. Security role names are mapped to principals using the `Role Name.`	A defined security role.	none

User Data Constraint

Defines how the client should communicate with the server.

Attribute	Description	Range of Values	Default Value
Description	A text description.	String	none
Transport Guarantee	Specifies that the communication between client and server. WebLogic Server establishes a Secure Sockets Layer (SSL) connection when the user is authenticated using the `INTEGRAL` or `CONFIDENTIAL` constraint.s	■ `NONE`—the application does not require any transport guarantees. ■ `INTEGRAL`—the application requires that the data be sent between the client and server in such a way that it cannot be changed in transit. ■ `CONFIDENTIAL`—the application requires that the data be transmitted in a fashion that prevents other entities from observing the contents of the transmission.	none

Web Resource Collection

Defines the components of the Web application to which this security constraint is applied.

Attribute	Description	Range of Values	Default Value
Web Resource Name	The name of this Web resource collection.	String	none
Description	A text description of this security constraint.	String	none

URL Pattern	Use one or more of the URL Pattern elements to declare to which URL patterns this security constraint applies. If you do not use at least one of these elements, this web resource collection is ignored by WebLogic Server.	String	none
HTTP Method	Use one or more of the HTTP Method elements to declare which HTTP methods (`GET | POST |...`) are subject to the authorization constraint. If you omit the HTTP Method element, the default behavior is to apply the security constraint to all HTTP methods.	`GET` `POST`	none

LoginConfig

Attribute	Description	Range of Values	Default Value
Auth Method	Specifies the method used to authenticate the user. Possible values: `BASIC` - uses browser authentication `FORM` - uses a user-written HTML form `CLIENT-CERT`	`BASIC` - uses browser authentication `FORM` - uses a user-written HTML form `CLIENT-CERT`	BASIC

Realm Name	The name of the realm that is referenced to authenticate the user credentials. If omitted, the WebLogic realm is used by default. For more information, see Specifying a Security Realm at http://e-docs.bea.com/wls/docs61/adminguide/cnfgsec.html#cnfgsec004.	String	None
Login Page	The URI of a Web resource relative to the document root, used to authenticate the user. This can be an HTML page, JSP, or HTTP servlet, and must return an HTML page containing a FORM that conforms to a specific naming convention.	String	None
Error Page	The URI of a Web resource relative to the document root, sent to the user in response to a failed authentication login.	String	None

Security Roles

Attribute	Description	Range of Values	Default Value
Description	A text description of this security role.	String	none
Role Name	The role name. The name you use here must have a corresponding entry in the WebLogic-specific deployment descriptor, `weblogic.xml`, which maps roles to principals in the security realm.	String	system

Env Entries

Attribute	Description	Range of Values	Default Value
Description	A textual description.	String	none
Env Entry Name	The name of the environment entry.	String	MyEnvironment Entry
Env Entry Value	The value of the environment entry.	String	none
Env Entry Type	The name of the environment entry.	String	none

Ejb refs

Attribute	Description	Range of Values	Default Value
Description	A text description of the reference.	String	none
EJBRef Name	The name of the EJB used in the Web application. This name is mapped to the JNDI tree in the WebLogic-specific deployment descriptor `weblogic.xml`.	String	MyEJB Ref
EJBRef Type	The expected Java class type of the referenced EJB.	String	none
Home Interface Name	The fully qualified class name of the EJB home interface.	String	none
Remote Interface Name	The fully qualified class name of the EJB remote interface.	String	none
EJBLink Name	The `<ejb-name>` of an EJB in an encompassing J2EE application package.	String	none
Run As	A security role whose security context is applied to the referenced EJB.	A security role defined in this web application.	none

Security Role Assignment

Attribute	Description	Range of Values	Default Value
Role Name	Specifies the name of a security role.	A valid security role.	none
Principal Name	Specifies the name of a principal that is defined in the security realm. You can use multiple `<principal-name>` elements to map principals to a role. For more information on security realms, see the *Programming WebLogic Security* at http://e-docs.bea.com/wls/docs61/security/index.html.	A principal defined in the security realm.	none

Reference Descriptor

Resource Descriptions

Attribute	Description	Range of Values	Default Value
Res Ref Name	Specifies the name of a resource reference.	String	none
Jndi Name	Specifies a JNDI name for the resource.	Java character set name	none

EJB Reference Description

Attribute	Description	Range of Values	Default Value
Ejb Ref Name	Specifies the name of an EJB reference used in your Web application.	String	none
Jndi Name	Specifies a JNDI name for the reference.	Java character set name	none

Session Descriptor

:

Parameter Name	Default Value	Parameter Value
URLRewritingEnabled	true	Enables URL rewriting, which encodes the session ID into the URL and provides session tracking if cookies are disabled in the browser.
IDLength	52	Sets the size of the session ID. The minimum value is 8 bytes and the maximum value is `Integer.MAX_VALUE`. If you are writing a WAP application, you must use URL rewriting because the WAP protocol does not support cookies. Also, some WAP devices have a 128-character limit on URL length (including parameters), which limits the amount of data that can be transmitted using URL re-writing. To allow more space for parameters, use this parameter to limit the size of the session ID that is randomly generated by WebLogic Server

Parameter Name	Default Value	Parameter Value
CookieComment	Weblogic Server Session Tracking Cookie	Specifies the comment that identifies the session tracking cookie in the cookie file. If unset, this parameter defaults to `WebLogic Session Tracking Cookie`. You may provide a more specific name for your application.
CookieDomain	Null	Identifies the server to which the browser sends cookie information when the browser makes a request. For example, setting the CookieDomain to `.mydomain.com` returns cookies to any server in the `*.mydomain.com` domain. The domain name must have at least two components; setting a name to `*.com` or `*.net` is invalid. If unset, this parameter defaults to the server that issued the cookie.
CookieMaxAgeSecs	-1	Sets the life span of the session cookie, in seconds, after which it expires on the client. If the value is 0, the cookie expires immediately. The maximum value is MAX_VALUE, where the cookie lasts forever. If set to `-1`, the cookie expires when the user exits the browser.
CookieName	`JSESSIONID`	Defines the session cookie name. Defaults to `JSESSIONID` if unset. You may set this to a more specific name for your application.
CookiePath	Null	Specifies the pathname to which the browser sends cookies. If unset, this parameter defaults to `/` (slash), where the browser sends cookies to all URLs served by WebLogic Server. You may set the path to a narrower mapping, to limit the request URLs to which the browser sends cookies.

Parameter Name	Default Value	Parameter Value
InvalidationIntervalSecs	`60`	Sets the time, in seconds, that WebLogic Server waits between doing house-cleaning checks for timed-out and invalid sessions, and deleting the old sessions and freeing up memory. Use this parameter to tune WebLogic Server for best performance on high traffic sites. The minimum value is every second (1). The maximum value is once a week (604,800 seconds). If unset, the parameter defaults to 60 seconds.
JDBConnectionTimeoutSecs	120	Sets the time, in seconds, that WebLogic Server waits before timing out a JDBC connection, where x is the number of seconds between.
PersistenceStoreDir	`session_db`	If you have set `PersistentStoreType` to `file`, this parameter sets the directory path where WebLogic Server will store the sessions. The directory path is either relative to the temp directory or an absolute path. The temp directory is either a generated directory under the `WEB-INF` directory of the Web application, or a directory specified by the context-param `javax.servlet.context.tmpdir`. Ensure that you have enough disk space to store the *number of valid sessions* multiplied by the *size of each session*. You can find the size of a session by looking at the files created in the `PersistenceStoreDir`. You can make file-persistent sessions clusterable by making this directory a shared directory among different servers. You must create this directory manually.
PersistenceStorePool	None	Specifies the name of a JDBC connection pool to be used for persistence storage.

Parameter Name	Default Value	Parameter Value
PersistentStoreType	`memory`	Sets the persistent store method to one of the following options: ■ `memory`—disables persistent session storage ■ `file`—uses file-based persistence (See also `PersistenceStoreDir`, above) ■ `jdbc`—uses a database to store persistent sessions. (see also `PersistenceStorePool`, above) ■ `replicated`—same as `memory`, but session data is replicated across the clustered servers ■ cookie—all session data is stored in a cookie in the user's browser
CookiesEnabled	True	Use of session cookies is enabled by default and is recommended, but you can disable them by setting this property to `false`. You might turn this option off to test using URL rewriting.
TrackingEnabled	True	When set to true, session tracking is enabled.
TimeoutSecs	3600	Sets the time, in seconds, that WebLogic Server waits before timing out a session, where x is the number of seconds between a session's activity. Minimum value is 1, default is 3600, and maximum value is `integer MAX_VALUE`. On busy sites, you can tune your application by adjusting the timeout of sessions. While you want to give a browser client every opportunity to finish a session, you do not want to tie up the server needlessly if the user has left the site or otherwise abandoned the session. This parameter can be overridden by the `session-timeout` element (defined in minutes) in `web.xml`. For more information, see `Session Timeout`.
ConsoleMainAttribute		If you enable Session Monitoring in the WebLogic Server Administration Console, set this parameter to the name of the session parameter you will use to identify each session that is monitored.

Parameter Name	**Default Value**	**Parameter Value**
PersistentStoreCookieName	`WLCOOKIE`	Sets the name of the cookie used for cookie-based persistence.

JSP Descriptor

Parameter Name	**Default Value**	**Parameter Value**
Compile Command	`javac`, or the Java compiler defined for a server under the configuration /tuning tab of the WebLogic Server Administration Console	Specifies the full pathname of the standard Java compiler used to compile the generated JSP servlets. For faster performance, specify a different compiler, such as IBM's Jikes or Symantec's `sj`.
Compiler Class	`None`	Name of a Java compiler that is executed in WebLogic Servers's virtual machine. (Used in place of an executable compiler such as `javac` or `sj`.)

Compile Flags	`None`	Passes one or more command-line flags to the compiler. Enclose multiple flags in quotes, separated by a space. For example:
		`java weblogic.jspc`
		` -compileFlags "-g -v" myFile.jsp`

Working Dir	`internally generated directory`	The name of a directory where WebLogic Server saves the generated Java and compiled class files for a JSP.

Parameter Name	Default Value	Parameter Value
Verbose	`true`	When set to `true`, debugging information is printed out to the browser, the command prompt, and WebLogic Server log file.
keepgenerated	`false`	Saves the Java files that are generated as an intermediary step in the JSP compilation process. Unless this parameter is set to `true`, the intermediate Java files are deleted after they are compiled.
Page Check Seconds	`1`	Sets the interval, in seconds, at which WebLogic Server checks to see if JSP files have changed and need recompiling. Dependencies are also checked and recursively reloaded if changed. If set to 0, pages are checked on every request. If set to `-1`, page checking and recompiling is disabled.
Encoding	Default encoding of your platform	Specifies the default character set used in the JSP page. Use standard Java character set names (see http://java.sun.com/j2se/1.3/docs/guide/intl/encoding.doc.htm). If unset, this parameter defaults to the encoding for your platform. A JSP page directive (included in the JSP code) overrides this setting. For example: `<%@ page contentType="text/html; charset=custom-encoding”%>`
Package Prefix	`jsp_servlet`	Specifies the package into which all JSP pages are compiled.
No Try Blocks	`false`	If a JSP file has numerous or deeply nested custom JSP tags and you receive a `java.lang.VerifyError` exception when compiling, use this flag to allow the JSPs to compile correctly.
Precompile	`false`	When set to true WebLogic Server automatically compiles all JSPs on startup.
Compiler Supports Encoding	`False`	Specifies the encoding used by the WebLogic JSP compiler to create the intermediate.`java` file. When set to true, the JSP compiler uses the encoding specified with the `contentType` attribute contained in the `page` directive on the JSP page, or, if a `contentType` is not specified, the encoding defined with the `encoding` parameter in the jsp-descriptor. When set to false, the JSP compiler uses the default encoding for the JVM when creating the intermediate `.java` file.

Container Descriptor

Attribute	Description	Range of Values	Default Value
Check Auth on Forward Enabled	When enabled, requires authentication of forwarded requests from a servlet or JSP.	Checked, Unchecked	Unchecked
redirect-with-absoute-url	Controls whether the `javax.servlet.http.HttpServletResponse.SendRedirect()` method redirects using a relative or absolute URL. Set this element to `false` if you are using a proxy HTTP server and do not want the URL converted to a non-relative link. The default behavior is to convert the URL to a non-relative link.	boolean	true

Charset Params

Input Charset Descriptors

Attribute	Description	Range of Values	Default Value
Resource Path	A path which, if included in the URL of a request, signals WebLogic Server to use the Java character set specified by the Java Charset Name field.	String	none
Java Charset Name	Specifies the Java characters set to use.	Java character set name	none

Character Set Mapping

Attribute	Description	Range of Values	Default Value
IANA Charset Name	Specifies the IANA character set name that is to be mapped to the Java character set specified by the Java Charset Name field.	IANA character set name	none
Java Charset Name	Specifies the Java characters set to use.	Java character set name	none

Configuring a WebLogic Web Service

The Web Service information in the Deployment Descriptor Editor is provided for monitoring purposes only; changing any of the values might result in the deployed Web Service not functioning correctly.

Web Services

RPC Services

Attribute	Description	Range of Values	Default Value
JNDI Name	JNDI name of the stateless session EJB that comprises the RPC-style Web service.	String.	None.
Home Interface	The Home interface of the stateless session EJB.	String.	None.
Remote Interface	The Remote interface of the stateless session EJB.	String.	None.
URI	URI used by client applications to invoke the Web service.	String.	None.

Message Services

Attribute	Description	Range of Values	Default Value
Service Name	Name of the SOAP Servlet that handles SOAP messages between WebLogic Server and client applications.	String.	Null.
Destination	JNDI name of the JMS topic or queue which receives or sends data between WebLogic Server and the client application.	String.	None.
Destination Type	The type of JMS destination: topic or queue.	`Topic` or `Queue`.	None.
Action	Specifies whether the client application that invokes this message-style Web service sends or receives data to the JMS destination. Specify `send` if the client sends data to the JMS destination and `receive` if the client receives data from the JMS destination.	`Send` or `Receive`.	None.
Connection Factory	JNDI name of the ConnectionFactory used to create a connection to the JMS destination.	String.	None.

Attribute	Description	Range of Values	Default Value
URI	URI used by clients to invoke the Web service.	String.	None.

66 JTA

These procedures describe how to use the Administration Console to set the attributes for configuring JTA. For more information, see Managing Transactions and Managing JDBC Connectivity in the *Administration Guide* and *Programming WebLogic JTA*, the developers' guide.

Configuring JTA

1. Click the Domain node. A dialog displays in the right pane showing the tabs associated with configuration for the domain.
2. Click the JTA tab.
3. Enter values in the Timeout Seconds, Abandon Timeout Seconds, Before Completion Iteration Limit, Max Transactions, and Max Unique Name Statistics attribute fields or accept the default values as assigned.
4. Enable or disable the Forget Heuristics attribute as desired.
5. Click Apply to save any changes you made.

67 JTA Transaction

By default, this pane allows the user to display transactions which have been active for at least 5 seconds sorted by the following criteria:

- Name
- Status
- Servers
- Resources
- Properties

Edit the number of seconds and click View to change the age limit of transactions displayed.

Click the Customize this view text link to change the columns to display or the sort order.

68 LDAP Realm (Deprecated)

The following procedures describe how to use the Administration Console to set the attributes for creating and managing LDAP Realms. To read more about LDAP Realms, please see Managing Security in the *Administration Guide*.

Create an LDAP Realm

1. Click the Realms node in the left pane. The Realms table displays in the right pane showing all the LDAP realms defined in the domain.
2. Click the Create a New LDAP Realm text link. A dialog displays in the right pane showing the tabs associated with configuring a new realm.
3. Enter a value in the Name attribute field.
4. Click the Create button in the lower right corner to create a realm instance with the name you specified in the Name field. The new instance is added under the Realms node in the left pane.
5. Click the LDAP, Users, and Groups tabs individually and change the attribute fields or accept the default values as assigned.
6. Click Apply to save any changes you made.

Clone an LDAP Realm

1. Click the Realms node in the left pane. The Realms table displays in the right pane showing all the LDAP realms defined in the domain.
2. Click the Clone icon in the row of the realm you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a new realm.
3. Enter a value in the Name attribute field.
4. Click Create to create a realm instance with the name you specified in the Name field. The new instance is added under the Realms node in the left pane.
5. Click the LDAP, Users, and Groups tabs individually and change the attribute fields or accept the default values as assigned.
6. Click Apply to save any changes you made.

Delete an LDAP Realm

1. Click the Realms node in the left pane. The Realms table displays in the right pane showing all the LDAP realms defined in the domain.
2. Click the Delete icon in the row of the realm you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click Yes to delete the realm. The realm icon under the Realms node is deleted.

Before you can use the Windows NT Security realm, you need to enable the Caching Realm and enter the class name of the Windows NT Security realm in the Basic Realm field.

Configuration

General

Attribute	**Description**	**Range of Values**	**Default Value**
Name	This attribute specifies the name of the LDAP Security realm. For example, AccountingRealm	The name can be up to 256 alphanumeric characters, but may not contain commas or spaces.	Null
Realm Class Name	This attribute specifies the name of the Java class that contains the LDAP Security realm. The Java class should be included in the CLASSPATH of WebLogic Server	This attribute can not be changed.	

LDAP

Attribute	**Description**	**Range of Values**	**Default Value**
LDAPURL	The location of the LDAP server. Change the URL to the name of the computer on which the LDAP server is running and the number of the port at which it is listening. If you want WebLogic Server to connect to the LDAP server using the SSL protocol, use the LDAP server's SSL port in the URL.		ldap://ldapserver:389
Principal	The distinguished name (DN) of the LDAP User used by WebLogic Server to connect to the LDAP server. This user must be able to list LDAP Users and Groups.		
Credential	The password that authenticates the LDAP User, as defined in the Principal field.	Valid password	null

Attribute	Description	Range of Values	Default Value

Enable SSL	Option for enabling the use of the SSL protocol to protect communications between the LDAP server and WebLogic Server. Keep in mind the following guidelines:
	■ Disable this field if the LDAP server is not configured to use the SSL protocol.
	■ If you set the UserAuthentication field to `external`, this field must be enabled.
		Boolean
		True = selected
		False = not selected
			Null

Auth Protocol	The type of authentication used to authenticate the LDAP server. Netscape Directory Server supports `CRAM-MD5`. Microsoft Site Server and Novell NDS support `Simple`.	■ `None` for no authentication. ■ `Simple` for password authentication. ■ `CRAM-MD5` for certificate authentication.	None

Users

Attribute	Description	Range of Values	Default Value
User Authentication	This attribute determines the method for authenticating Users.	Set the attribute one of the following: ■ `Local` specifies that the LDAP Security realm retrieves user data including the password from the LDAP Directory server and checks the password in WebLogic Server. The `Local` setting is appropriate for Netscape Directory Server and Microsoft Site Server. ■ `External` specifies that the LDAP Security realm authenticates a User by attempting to bind to the LDAP Directory server with the username and password supplied by the WebLogic Server client. If you choose the `External` setting, you must also use the SSL protocol. The `External` setting is appropriate for Novell NDS. ■ Bind specifies that the LDAP Security realm authenticates a User	Null

Attribute	Description	Range of Values	Default Value
User Password Attribute	This attribute sets the password of the LDAP user.	The password of the LDAP User.	Null
User DN	Set this attribute to the list of attributes so that when combined with the attributes in the `UserNameAttribute` the attribute uniquely identifies a User.	String	Null
User Name Attribute	This attribute sets the login name of the LDAP User.	The value of this field can be the common name of an LDAP User but usually it is an abbreviated string, such as the User ID.	null

Groups

Attribute	Description	Range of Values	Default Value
Group DN	Enter the list of attributes that, combined with the `GroupNameAttribute` attribute uniquely identifies a Group in the LDAP server.	String	Null
Group Name Attribute	Enter the name of a Group in the LDAP Server. It is usually a common name.	String	Null
Group IS Context	This boolean attribute specifies how Group membership is recorded in the LDAP server.	Boolean Enable = selected Disabled = not selected Enable this attribute if each Group entry contains one User. By default, the attribute is enabled. Disable this attribute if there is one Group entry containing an attribute for each Group member.	Selected
Group Username Attribute	Set this attribute to the name of the LDAP attribute that contains a Group member in a Group entry.	Valid group member name	Null

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for user supplied information.	The value must be an alphanumeric string.	Null

69 Machine

These procedures describe how to use the Administration Console to set the attributes for configuring and managing machines. For more information, see Starting and Stopping WebLogic Servers and WebLogic Servers and Clusters in the *Administration Guide*.

Configure a Machine

1. Click the Machines node. The Machines table displays in the right pane showing all the machines defined in the domain.
2. Click the Configure a New Machine text link. A dialog displays in the right pane showing the tabs associated with configuring a new machine.
3. Enter the value in the Name attribute field.

Note: Assign a unique name to the machine. Each configurable resource in your WebLogic environment should have a unique name.

4. Click Create to create a machine instance with the name you specified in the Name field. The new instance is added under the Machines node in the left pane.
5. Click the Node Manager tab and modify the attribute values as needed.
6. Click Apply to save the changes.

For more information about Node Manager and instructions for configuring it, see “Node Manager” in *WebLogic Server Administration Guide*.

Configure startup arguments for each Managed Server on the machine that you want to start with Node Manager. See "Step 3: Configure Startup Information for Managed Servers" in *WebLogic Server Administration Guide*.

Clone a Machine

1. Click the Machines node. The Machines table displays in the right pane showing all the machines defined in the domain.
2. Click the Clone icon in the row of the machine you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a machine.
3. Enter the value in the Name attribute field.
4. Click Clone to create a machine instance with the name you specified in the Name field. The new instance is added under the Machines node in the left pane.
5. Click the Node Manager tab and modify the attribute values as needed.
6. Click Apply to save the changes.

Delete a Machine

1. Click the Machines node. The Machines table displays in the right pane showing all the machines defined in the domain.
2. Click the Delete icon in the row of the machine you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click Yes to delete the machine. The machine icon under the Machines node is deleted.

Assign a Machine

1. Click the instance node in the left pane under Machines for the machine you want to assign. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Servers tab.
3. Select one or more targets in the Available column that you want to assign to the machine.
4. Click the mover control to move the targets you selected to the Chosen column.
5. Click Apply to save your assignments.

Binding to Protected Ports on UNIX

On UNIX systems, only processes that run under a privileged user account (in most cases, root) can bind to ports lower than 1024. However, long-running processes like WebLogic Server should not run under these privileged accounts.

If you want a WebLogic Server instance to bind to protected ports (such as port 80), do either of the following:

- Start WebLogic Server instances from a non-privileged account and configure your firewall to use Network Address Translation (NAT) software to map protected ports to unprotected ones. BEA does not provide NAT software.
- For each WebLogic Server instance that needs access to privileged ports, configure the server to start under a privileged user account, bind to privileged ports, and change its user ID to a non-privileged account.

  Note: WebLogic Server uses native code to change user IDs. To verify that this feature is supported for your UNIX platform, refer to "Supported Configurations."

To configure a server instance to bind to protected ports on UNIX:

1. Start the Administration Server for the domain.

   See "Starting and Stopping WebLogic Servers" in the *Administration Guide*.

2. Stop the server instances that you want to configure.

3. Create a UNIX machine and assign the server to the machine:

   a. In the Administration Console, in the left pane, click on the Machines folder.

   b. In the right pane, select the Configure a New Unix Machine link.

   c. On the Create a New UNIX Machine page, in the Name field, enter a name for the new machine.

      A **machine** represents the physical machine that hosts one or more WebLogic Server instances. BEA recommends that you create one UNIX machine for each UNIX host that runs instances of WebLogic Server.

      The machine name is used for identification within the WebLogic Server domain; it does not have to correspond to the machine's network name.

      Note: Assign a unique name to the machine. Each configurable resource in your WebLogic Server environment should have a unique name.

   d. Click Create.

      The General tab displays additional fields.

   e. To specify a non-privileged user account under which the server instance runs, place a check mark in the Enable Post-bind UID field and enter the user ID in the Post-Bind UID field.

      The user ID that you enter must have read, write, and execute privileges within the BEA Home directory, the WebLogic Server product directory tree, and your domain directory.

      The default value of the Post-Bind UID field, `nobody`, is a standard UNIX account that provides the least possible privileges. While the `nobody` account is acceptable for use in a development environment, in a production environment, BEA recommends that you create an operating-system user account specifically for running instances of WebLogic Server. See "Securing the WebLogic Server Host."

   f. To specify a non-privileged group instead of (or in addition to) providing a user ID, place a check mark in the Enable Post-bind GID field and enter the group ID in the Post-Bind GID field.

g. Click Apply.

h. Select the Servers tab. Move each server instance that you want to run on this UNIX machine from the Available list to the Chosen list. Then click Apply.

i. If you want to use the Node Manager to start server instances on this UNIX machine, click the Configuration > Node Manager tab and specify the address and listen port through which the Node Manager can be reached.

   In a production environment, BEA recommends that you **specify a listen port that is secured by SSL**. See “Binding to Protected Ports with Servers That a Node Manager Starts” on page 69-5.

   Click Apply to apply your changes.

4. Log in to the WebLogic Server host computer under an account that has access to protected ports.
5. Do either of the following for the server instances that you assigned to the UNIX machine:
   - Start a WebLogic Server instance by invoking the `weblogic.Server` class or by invoking a script that invokes the class.
   - (For Managed Servers only) Start the Node Manager. Then use the Node Manager to start Managed Servers.

     See "Node Manager" in the *Administration Guide*.

The WebLogic Server instance starts under the privileged user ID. After it binds to ports, it invokes native code to change its user or group ID to the one you specified in the UNIX machine configuration.

Binding to Protected Ports with Servers That a Node Manager Starts

Node Manager is a small Java application that runs on WebLogic Server hosts and can start and stop instances of WebLogic Server. When Node Manager starts a server instance, the server instance starts under the same user account under which the Node Manager is running.

If you Node Manager to start server instances that bind to protected ports, you must run the Node Manager under a privileged account. If you configure a post-bind user ID or group for a server instance, after Node Manager starts a server, the server binds to ports and then changes the user account under which it runs.

In a production environment, BEA recommends that you do the following to secure Node Manager:

1. Configure the Node Manager to listen on a port that is secured by SSL.
2. In the Node Manager's `nodemanager.hosts` file, specify the host name of the Administration Server only.

See "Node Manager" in the *Administration Guide*.

With the above configuration, Node Manager will accept requests only on a secure port and only from a single, known host.

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	This attribute lists the name of this machine	The name can be up to 256 alphanumeric characters, but may not contain commas or spaces.	Null

Servers

Attribute	Description	Range of Values	Default Value
Servers	This attribute lists the servers running on this machine.	List	null

Node Manager

Attribute	Description	Range of Values	Default Value
Listen Address	The host name or IP address where Node Manager listens for requests from the Administration Server. Specify this address when starting the Node Manager process on the machine.		LocalHost
Listen Port	The port where Node Manager listens for requests from the Administration Server. Specify this port when starting the Node Manager process on the machine.		5555
Certificate	The certificate used by the Administration Server to talk to this Node Manager. The default certificate is . It is recommended that you obtain a new certificate for a production environment.		config/demo.crt

Attribute	Description	Range of Values	Default Value
Certificate Password	The certificate password is not displayed because it is encrypted. If you change the certificate used by Node Manager, change the password to match the password used to encrypt the private key in the new digital certificate.	Valid password for the certificate	Null
Trusted Certs File	Contains the list of certificate authorities that are recognized. The certificate authority referred to in the digital certificate you are using must be listed in this file		config/trusted.crt

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for user supplied information.	The value must be an alphanumeric string.	Null

70 MailSession

Create a Mail Session

1. Click the Mail node. The Mail table displays in the right pane showing all the mail sessions defined in your domain.
2. Click the Create a New Mail Session text link. A dialog displays in the right pane showing the tabs associated with configuring a new mail session.
3. Enter values in the Name, JNDI Name, and Properties attribute fields.
4. Click Create to create a mail session instance with the name you specified in the Name field. The new instance is added under the Mail node in the left pane.

Clone a Mail Session

1. Click the Mail node. The Mail table displays in the right pane showing all the mail sessions defined in your domain.
2. Click the Clone icon in the row of the mail session you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a mail session.
3. Enter values in the Name, JNDI Name, and Properties attribute fields.
4. Click Create to create a mail session instance with the name you specified in the Name field. The new instance is added under the Mail node in the left pane.

Delete a Mail Session

1. Click the Mail node. The Mail table displays in the right pane showing all the mail sessions defined in your domain.
2. Click the Delete icon in the row of the mail session you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click Yes to delete the mail session. The mail session icon under the Mail node is deleted.

Monitor All Instances of a Mail Session

1. Click the Mail node. The Mail table displays in the right pane showing all the mail sessions defined in your domain.
2. Click the Monitor All Instances icon in the row of the mail session you want to monitor. A dialog displays in the right pane showing all the instances of the mail session.

Assign a Mail Session

1. Click the instance node in the left pane under Mail for the mail session you want to assign. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Targets tab.
3. Complete the following steps for the Servers, Groups, and Clusters tabs:
   a. Select one or more targets in the Available column that you want to assign to the mail session.

b. Click the mover control to move the targets you selected to the Chosen column.

c. Click Apply to save your assignments.

Configuration

Attribute	Description	Range of Values	Default Value
Name	This attribute returns the name of this mail session.	String	Null
JNDI Name	This attribute sets the JNDI name for this mail session.	String	Null

Properties	These attributes set the properties. They are required for this mail session.	mail.transport.protocol=SMTP
		mail.store.protocol=pop3
			Null

Targets

Servers

Attribute	Description	Range of Values	Default Value
Target Servers	This attribute allows the user to select servers on which this mail session will be deployed.	List	None Selected

Clusters

Attribute	Description	Range of Values	Default Value
Target Clusters	This attribute allows the user to select clusters on which this mail session will be deployed.	List	None Selected

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute allows user supplied comments	String	Null

71 MBean

The Administration Server is populated with JavaBean-like objects called Management Beans (MBeans), which are based on Sun's Java Management Extension (JMX) standard. These objects provide management access to domain resources. The Administration Server contains both configuration MBeans and run-time MBeans. Configuration MBeans provide both SET (write) and GET (read) access to configuration attributes.

For additional information about MBeans, please see the Administration Guide.

72 Message Driven EJB Runtime

You can monitor your Message Driven EJBs by the following statistics:

Statistic	Description
Idle Beans Count	Number of beans currently idle
Beans In Use Count	Number of beans currently in use
Waiter Total Count	Number of transactions currently waiting
Timeout Total Count	Number of transactions that have timed out
JMS Connection Alive	Status of the JMS connection

73 NTRealm

The following procedures describe how to use the Administration Console to set the attributes for creating and managing Realms. To read more about Realms, please see Managing Security in the *Administration Guide*.

Configure a NT Realm

1. Click the Realms node in the left pane. The Realms table displays in the right pane showing all the NT realms defined in the domain.
2. Click the Configure a New NT Realm text link. A dialog displays in the right pane showing the tabs associated with configuring a new realm.
3. Enter values in the Name and Primary Domain attribute fields.
4. Click Create to create a realm instance with the name you specified in the Name field. The new instance is added under the Realms node in the left pane.

Clone a NT Realm

1. Click the Realms node in the left pane. The Realms table displays in the right pane showing all the NT realms defined in the domain.
2. Click the Clone icon in the row of the realm you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a new realm.

3. Enter values in the Name and Primary Domain attribute fields.
4. Click Create to create a realm instance with the name you specified in the Name field. The new instance is added under the Realms node in the left pane.

Delete a NT Realm

1. Click the Realms node in the left pane. The Realms table displays in the right pane showing all the NT realms defined in the domain.
2. Click the Delete icon in the row of the realm you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click Yes to delete the realm. The realm icon under the Realms node is deleted.

Before you can use the Windows NT Security realm, you need to enable the Caching Realm and enter the class name of the Windows NT Security realm in the Basic Realm field.

Configuration

Attribute	Description	Range of Values	Default Value
Name	This attribute specifies the name of the Windows NT Security realm, such as, AccountingRealm.	This attribute must be an alphanumeric string of up to 256 characters.	MyNTRealm
Realm Class Name	This attribute reports the name of the Java class that implements the Windows NT Security realm. The Java class needs to be in the CLASSPATH of WebLogic Server.	This attribute can not be changed	weblogic.security.acl.ntrealm.NTRealm
Primary Domain	This attribute allows you to enter the host and port number of the computer where Users and Groups are defined for the Windows NT domain.		Null

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for optional user-supplied information.	The value must be an alphanumeric string.	Null

74 Preferences

Configuration

General

Attribute	Description	Range of Values	Default Value
Language	This attribute enables the user to select the language for the console and the help system.	English, Japanese	English
Autorefresh Every	This attribute sets the time for autorefresh of the console.	Integer, in seconds	10

Remember Last Tab	This attribute enables or disables the ability to return to the last tab selected.	Boolean
		Enabled = selected
		Disabled = not selected
			Selected

Use Navigation Tree	TThis attribute enables or disables the ability to use the navigation tree on the console.	Boolean
		Enabled = selected
		Disabled = not selected
			Selected

Graphs

Attribute	Description	Range of Values	Default Value
Poll for Graph Data Every	This attribute sets the amount of time between polling for graph data.	Integer, in milliseconds	500

75 RDBMS Realm

The following table describe the attributes for creating and managing RDBMS Realms. To read more about Realms, please see Managing Security in the *Administration Guide*.

Configuration

Attribute	Description	Range of Values	Default Value
Name	This attribute specifies the name of the Security realm, such as, AccountingRealm.	The name can be up to 256 alphanumeric characters, but may not contain commas or spaces.	Null
Realm Classname	This attribute returns the name of the Java class that contains the Security realm. The Java class needs to be in the CLASSPATH of WebLogic Server.	This attribute can not be changed.	

Database

Attribute	Description	Range of Values	Default Value
Database Driver	This attribute sets the full class name of the JDBC driver. This class name must be in the CLASSPATH of WebLogic Server.		Null
Database URL	This attribute sets the URL for the database you are using with the RDBMS realm, as specified by your JDBC driver documentation.		Null
Database Username	This attribute specifies the username used to log in to the database.		Null
Database Password	This attribute sets the password for the default user of the database as determined by the administrator.		Null

Schema Properties

Attribute	Description	Range of Values	Default Value
Schema Properties	This attribute is used to specify properties for obtaining Users, Groups, ACLs, and permissions from the database you are using.	key=value list	Null

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for optional user-supplied information.	The value must be an alphanumeric string.	Null

CHAPTER

76 RDBMS Deployment Descriptors

weblogic-cmp-rdbms-jar.xml deployment descriptors

This section describes the 6.1 WebLogic Server-specific XML elements found in `weblogic600-cmp-rdbms-jar.xml` that are used to define the deployment descriptors in the WebLogic Server EJB container. These elements map to fields, of approximately the same name, in the WebLogic Server Administration Console Use these deployment descriptors for 2.0 EJBs.

automatic-key-generation

Range of values:	n/a
Default value:	n/a
Requirements:	Optional.
Parent elements:	`weblogic-rdbms-bean`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `automatic-key-generation` element specifies the use of the Sequence/Key Generation feature.

cmp-field

Range of values:	Valid name
Default value:	n/a
Requirements:	Field is case sensitive and must match the name of the field in the bean and must also have a `cmp-entry` entry in the `ejb-jar.xml`.

Parent elements:	`weblogic-rdbms-bean`
	`field-map`
	`weblogic-rdbms-relation`
	`field-group`

Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

This name specifies the mapped field in the bean instance which should be populated with information from the database.

cmr-field

Range of values:	Valid name
Default value:	n/a
Requirements:	The field referenced in `cmr-field` must have a matching `cmr-field` entry in the `ejb-jar.xml`.
Parent elements:	`weblogic-rdbms-relation` `field-group`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `cmr-field` element specifies the name of a cmr-field.

column-map

Range of values:	n/a.
Default value:	n/a
Requirements:	The `key-column` element is not specified, if the `foreign-key-column` refers to a remote bean.
Parent elements:	`weblogic-rdbms-bean` `weblogic-relationship-role`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

This element represents the mapping of a foreign key column in one table in the database to a corresponding primary key. The two columns may or may not be in the same table. The tables to which the column belong are implicit from the context in which the `column-map` element appears in the deployment descriptor.

create-default-dbms-table

Range of values:	True | False.
Default value:	False
Requirements:	Use this element only for convenience during the development and prototyping phases. This is because the Table Schema in the DBMS CREATE statement used will be the container's best approximation of the definition. A production environment most likely, will require a more precise schema definition.
Parent elements:	`weblogic-rdbms-jar`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `create-default-dbms-table` element turns on or off a feature that automatically creates a default table based on the descriptions in the deployment files and the bean class. When set to False, this feature is turned off and table will not automatically be generated. When set to `True`, this feature is turned on and the table is automatically created. If `TABLE CREATION` fails, a `Table Not Found` error is thrown and the table must be created by hand.

data-source-name

Range of values:	Valid name of the data source used for all data base connectivity for this bean.
Default value:	n/a
Requirements:	Must be defined as a standard WebLogic Server JDBC data source for database connectivity.
Parent elements:	weblogic-rdbms-bean
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `data-source-name` that specifies the JDBC data source name to be used for all database connectivity for this bean.

db-cascade-delete

Range of values:	
Default value:	n/a
Requirements:	Only supported for Oracle database. Can only be specified for one-to-one or one-to-many relationships.
Parent elements:	`weblogic-rdbms-bean` `weblogic-relationship-role`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `db-cascade-delete` element specifies whether the database cascade feature is turned on. If this element is not specified, WebLogic Server

dbms-column

Range of values:	Valid name
Default value:	n/a
Requirements:	`dbms-column` is case maintaining, although not all database are case sensitive.
Parent elements:	`weblogic-rdbms-bean` `field-map`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The name of the database column to which the field should be mapped.

dbms-column-type

Range of values:	Valid name
Default value:	n/a
Requirements:	`Available for use with Oracle database only.`
Parent elements:	`weblogic-rdbms-bean` `field-map`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `dbms-column-type` element maps the current field to a Blob or Clob in an Oracle database. This element can be either OracleBlob or OracleCLob.

delay-database-insert-until

Range of values:	
Default value:	`ejbPostCreate`
Requirements:	Database insert is delayed until after `ejbPostCreate` when a `cmr-field` is mapped to a `foreign-key column` that does not allow null values. In this case, the `cmr-field` must be set to a non-null value in `ejbPostCreate` before the bean is inserted into the database. The `cmr-fields` may not be set during `ejbCreate`, before the primary key of the bean is known.
Parent elements:	`weblogic-rdbms-bean`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `delay-database-insert-until` element specifies the precise time when a new bean that uses RDBMS CMP is inserted into the database.

It is advisable to delay the database insert until after the `ejbPostCreate` method modifies the persistent fields of the bean. This can yield better performance by avoiding an unnecessary store operation.

For maximum flexibility, you should avoid creating related beans in your `ejbPostCreate` method. This may make delaying the database insert impossible if database constraints prevent related beans from referring to a bean that has not yet been created.

ejb-name

Range of values:	Valid name of an EJB`.`
Default value:	n/a
Requirements:	Must match the `ejb-name` of the cmp entity bean defined in the `ejb-jar.xml`.
Parent elements:	`weblogic-rdbms-bean`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The name that specifies an EJB as defined in the ejb-cmp-rdbms.xml. This name must match the `ejb-name` of a cmp entity bean contained in the `ejb-jar.xml`.

field-group

Range of values:	Valid name
Default value:	A special group named `default` is used for finders and relationships that have no group specified.
Requirements:	The default group contains all of a bean's cmp-fields, but none of its cmr-fields.
Parent elements:	`weblogic-rdbms-relation`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `field-group` element represents a subset of the cmp and cmr-fields of a bean. Related fields in a bean can be put into groups that are faulted into memory together as a unit. A group can be associated with a finder or relationship, so that when a bean is loaded as the result of executing a finder or following a relationship, only the fields specified in the group are loaded.

A field may belong to multiple groups. In this case, the `getXXX` method for the field faults in the first group that contains the field.

field-map

Range of values:	Valid name
Default value:	n/a
Requirements:	Field mapped to the column in the database must correspond to a cmp field in the bean.
Parent elements:	`weblogic-rdbms-bean`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The name of the mapped field for a particular column in a database that corresponds to a cmp field in the bean instance.

foreign-key-column

Range of values:	Valid name
Default value:	n/a
Requirements:	Must correspond to a column of a foreign key.
Parent elements:	`weblogic-rdbms-bean` `column-map`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `foreign-key-column` element represents a column of a foreign key in the database.

generator-name

Range of values:	n/a
Default value:	n/a
Requirements:	Optional.
Parent elements:	`weblogic-rdbms-bean` `automatic-key-generation`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `generator-name` element is used to specify the name of the generator.

For example;

- If the `generator-type` element is `ORACLE`, then the `generator-name` element would be the name of the `ORACLE_SEQUENCE` to be used.
- If the `generator-type` element is `NAMED_SEQUENCE_TABLE`, then the `generator-name` element would be the name of the `SEQUENCE_TABLE` to be used.

generator-type

Range of values:	n/a
Default value:	n/a
Requirements:	Optional
Parent elements:	`weblogic-rdbms-bean` `automatic-key-generation`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The generator-type element specifies the key generation method to use. The options include:

- `ORACLE`
- `SQL_SERVER`
- `NAMED_SEQUENCE_TABLE`

group-name

Range of values:	Valid name
Default value:	n/a
Requirements:	n/a

Parent elements:	`weblogic-rdbms-relation`
	    `field-group`
	`weblogic-rdbms-bean`
	    `finder`
	        `finder-query`

Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `group-name` element specifies the name of a field group.

include-updates

Range of values:	`True	False`
Default value:	`False`
Requirements:	The default value, which is `False`, provides the best performance.
Parent elements:	weblogic-rdbms-bean weblogic-query
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The include-updates element specifies that updates made during the current transaction must be reflected in the result of a query. If this element is set to `True`, the container will flush all changes made by the current transaction to disk before executing the query.

key-cache-size

Range of values:	n/a
Default value:	n/a
Requirements:	Optional
Parent elements:	`weblogic-rdbms-bean` `automatic-key-generation`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The key-cache-size element specifies the optional size of the primary key cache available in the automatic primary key generation feature.

key-column

Range of values:	Valid name
Default value:	n/a
Requirements:	Must correspond to a column of a primary key.
Parent elements:	`weblogic-rdbms-bean` `column-map`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `key-column` element represents a column of a primary key in the database.

max-elements

Range of values:	n/a
Default value:	n/a
Requirements:	n/a
Parent elements:	weblogic-rdbms-bean weblogic-query
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

`max-elements` specifies the maximum number of elements that should be returned by a multi-valued query. This element is similar to the `maxRows` feature in JDBC.

method-name

Range of values:	n/a
Default value:	n/a
Requirements:	The '*' character may not be used as a wildcard.
Parent elements:	`weblogic-rdbms-bean` `query-method`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `method-name` element specifies the name of a finder or `ejbSelect` method.

method-param

Range of values:	Valid name
Default value:	n/a
Requirements:	n/a
Parent elements:	`weblogic-rdbms-bean` `method-params`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `method-param` element contains the fully qualified Java type name of a method parameter.

method-params

Range of values:	list of valid names
Default value:	n/a
Requirements:	n/a
Parent elements:	`weblogic-rdbms-bean` `query-method`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `method-params` element contains an ordered list of the fully-qualified Java type names of the method parameters.

query-method

Range of values:	n/a
Default value:	n/a
Requirements:	n/a
Parent elements:	`weblogic-rdbms-bean`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `query-method` element specifies the method that is associated with a `weblogic-query`. It also uses the same format as the `ejb-jar.xml` descriptor.

relation-name

Range of values:	Valid name
Default value:	n/a
Requirements:	Must match the `ejb-relation-name` of an ejb-relation in the associated `ejb-jar.xml` descriptor file.
Parent elements:	`weblogic-rdbms-relation`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `relation-name` element specifies the name of a relation.

relationship-role-name

Range of values:	Valid name
Default value:	n/a
Requirements:	The name must match the `ejb-relationship-role-name` of an `ejb-relationship-role` in the associated `ejb-jar.xml` descriptor file.
Parent elements:	`weblogic-rdbms-relation` `weblogic-relationship-role`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `relationship-role-name` element specifies the name of a relationship role.

table-name

Range of values:	Valid, fully qualified SQL name of the source table in the database.
Default value:	n/a
Requirements:	`table-name` must be set in all cases.
Parent elements:	`weblogic-rdbms-bean` `weblogic-rdbms-relation`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The fully qualified SQL name of the table. The user defined for the `data-source` for this bean must have read and write privileges for this table, but does not necessarily need schema modification privileges.

weblogic-ql

Range of values:	n/a
Default value:	n/a
Requirements:	n/a
Parent elements:	weblogic-rdbms-bean weblogic-query
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `weblogic-ql` element specifies a query that contains a WebLogic specific extension to the ejb-ql language. You should specify queries that only use standard EJB-QL language features in the `ejb-jar.xml` deployment descriptor.

weblogic-query

Range of values:	n/a
Default value:	n/a
Requirements:	n/a
Parent elements:	`weblogic-rdbms-bean`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `weblogic-query` element allows you to associate WebLogic specific attributes with a query, as necessary. For example, `weblogic-query` can be used to specify a query that contains a WebLogic specific extension to EJB-QL. Queries that do not take advantage of WebLogic extensions to EJB-QL should be specified in the `ejb-jar.xml` deployment descriptor.

Also, the `weblogic-query` element is used to associate a `field-group` with the query if the query retrieves an entity bean that should be pre-loaded into the cache by the query.

weblogic-relationship-role

Range of values:	Valid name
Default value:	n/a
Requirements:	The mapping of a role to a table is specified in the associated `weblogic-rdbms-bean` and `ejb-relation` elements.
Parent elements:	`weblogic-rdbms-relation`
Deployment file:	`weblogic-cmp-rdbms-jar.xml`

Function

The `weblogic-relationship-role` element is used to express a mapping from a foreign key to a primary key. Only one mapping is specified for one-to-one relationships when the relationship is local. However, with a many-to-many relationship, you must specify two mappings

Multiple column mappings are specified for a single role, it the key is complex. No `column-map` is specified if the role is just specifying a group-name.

This section describes the 6.1 WebLogic Server-specific XML elements found in `weblogic510-cmp-rdbms-jar.xml` that are used to define the deployment descriptors in the WebLogic Server EJB container. These elements map to fields, of approximately the same name, in the WebLogic Server Administration Console Use these deployment descriptors for 1.1 EJBs.

RDBMS Definition elements

This section describes the RDBMS definition elements.

pool-name

`pool-name` specifies name of the WebLogic Server connection pool to use for this EJB's database connectivity.

schema-name

`schema-name` specifies the schema where the source table is located in the database. This element is required only if you want to use a schema that is not the default schema for the user defined in the EJB's connection pool.

Note: This field is case sensitive, although many SQL implementations ignore case.

table-name

`table-name` specifies the source table in the database. This element is required in all cases.

Note: The user defined in the EJB's connection pool must have read and write privileges to the specified table, though not necessarily schema modification privileges. This field is case sensitive, although many SQL implementations ignore case.

EJB Field-Mapping elements

This section describes the EJB field-mapping elements.

attribute-map

The `attribute-map` stanza links a single field in the EJB instance to a particular column in the database table. The `attribute-map` must have exactly one entry for each field of an EJB that uses WebLogic Server RDBMS-based persistence.

object-link

Each `attribute-map` entry consists of an `object-link` stanza, which represents a link between a column in the database and a field in the EJB instance.

bean-field

`bean-field` specifies the field in the EJB instance that should be populated from the database. This element is case sensitive and must precisely match the name of the field in the bean instance.

The field referenced in this tag must also have a `cmp-field` element defined in the `ejb-jar.xml` file for the bean.

dbms-column

`dbms-column` specifies the database column to which the EJB field is mapped. This tag is case sensitive, although many databases ignore the case.

Note: WebLogic Server does not support quoted RDBMS keywords as entries to `dbms-column`. For example, you cannot create an attribute map for column names such as “create” or “select” if those names are reserved in the underlying datastore.

Finder elements

This section describes the finder elements.

finder-list

The finder-list stanza defines the set of all finders that are generated to locate sets of beans.

finder-list must contain exactly one entry for each finder method defined in the home interface, except for findByPrimarykey. If an entry is not provided for findByPrimaryKey, one is generated at compilation time.

Note: If you do provide an entry for findByPrimaryKey, WebLogic Server uses that entry without validating it for correctness. In most cases, you should omit an entry for findByPrimaryKey and accept the default, generated method.

finder

The finder stanza describes a finder method defined in the home interface. The elements contained in the finder stanza enable WebLogic Server to identify which method in the home interface is being described, and to perform required database operations.

method-name

method-name defines the name of the finder method in the home interface. This tag must contain the exact name of the method.

method-params

The method-params stanza defines the list of parameters to the finder method being specified in method-name.

Note: WebLogic Server compares this list against the parameter types for the finder method in the EJB’s home interface; the order and type for the parameter list must exactly match the order and type defined in the home interface.

method-param

method-param defines the fully-qualified name for the parameter’s type. The type name is evaluated into a java.lang.Class object, and the resultant object must precisely match the respective parameter in the EJB’s finder method.

You can specify primitive parameters using their primitive names (such as “double” or “int”). If you use a non-primitive data type in a `method-param` element, you must specify a fully qualified name. For example, use `java.sql.Timestamp` rather than `Timestamp`. If you do not use a qualified name, `ejbc` generates an error message when you compile the deployment unit.

finder-query

`finder-query` specifies the WebLogic Query Language (WLQL) string that is used to retrieve values from the database for this finder.

Note: Always define the text of the `finder-query` value using the XML `CDATA` attribute. Using `CDATA` ensures that any special characters in the WLQL string do not cause errors when the finder is compiled.

finder-expression

`finder-expression` specifies a Java language expression to use as a variable in the database query for this finder.

Note: Future versions of the WebLogic Server EJB container will use the EJB QL query language (as required by the EJB 2.0 specification). EJB QL does not provide support for embedded Java expressions. Therefore, to ensure easier upgrades to future EJB containers, create entity EJB finders *without* embedding Java expressions in WLQL.

77 Realm

By default, this pane allows the user to sort the objects by the following criteria:

- Name

78 Server

These procedures describe how to use the Administration Console to set the attributes for configuring and managing WebLogic Servers. For more information, see Configuring WebLogic Servers and Clusters and Starting and Stopping WebLogic Servers in the Administration Guide.

Configure a Server

1. Click the Servers node in the left pane. The Servers table displays in the right pane showing all the servers defined in the domain.
2. Click the Configure a New Server text link. A dialog displays in the right pane showing the tabs associated with configuring a new server.
3. Enter values in the Name, Machine, and Listen Port, Administration Port, and Listen Address attribute fields.

Note: Each server in your WebLogic server environment must have a unique name — even if the servers are in different domains.

4. Click the Create button in the lower right corner to create a server instance with the name you specified in the Name field. The new instance is added under the Servers node in the left pane.
5. Click the additional tabs and change the attribute fields or accept the default values as assigned.
6. Click Apply to save any changes you made.

Clone a Server

1. Click the Servers node in the left pane. The Servers table displays in the right pane showing all the servers defined in the domain.
2. Click the Clone icon in the row of the server you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a new server.
3. Enter values in the Name, Machine, and Listen Port, Administration Port, and List attribute fields.
4. Click the Clone button in the lower right corner to create a server instance with the name you specified in the Name field. The new instance is added under the Servers node in the left pane.
5. Click the additional tabs and change the attribute fields or accept the default values as assigned.
6. Click Apply to save any changes you made.

Delete a Server

1. Click the Servers node in the left pane. The Servers table displays in the right pane showing all the servers defined in the domain.
2. Click the Delete icon in the row of the server you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click Yes to delete the server. The server icon under the Servers node is deleted.

View Server Log

1. Right-click the instance node in the left pane under Servers for the server whose log you want to view. This opens a pop-up menu.
2. Click View Server Log. A new browser window is displayed showing the log.

View Server JNDI Tree

1. Right-click the instance node in the left pane under Servers for the server whose JNDI tree you want to view. This opens a pop-up menu.
2. Click View JNDI Tree. A new browser window is displayed showing the naming context data.

View Server Execute Queues

1. Click the instance node in the left pane under Servers for the server whose execute queues you want to view. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Monitoring tab.
3. Click the View Execute Queues text link. The Execute Queues table displays in the right pane showing all the execute queues defined for this server.

View Server Execute Threads

1. Right-click the instance node in the left pane under Servers for the server whose Execute Threads data you want to view. This opens a pop-up menu.
2. Click View Execute Threads. The information displays in the right pane.

View Server Sockets

1. Click the instance node in the left pane under Servers for the server whose sockets you want to view. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Monitoring tab.
3. Click the Monitor All Active Sockets link.. The Sockets table displays in the right pane showing all the sockets defined for this server.

View Server Connections

1. Click the instance node in the left pane under Servers for the server whose connections you want to view. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Monitoring tab.
3. Click the Monitor All Connections link. The Connections table displays in the right pane showing all the connections defined for this server.

Force Garbage Collection on a Server

1. Click the instance node in the left pane under Servers for the server whose memory usage you want to view. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Monitoring tab.
3. Check the Memory Usage graph for high usage. Note that the Memory Usage graph will only display for servers that are currently running.
4. Click the Force Garbage Collection text link to force garbage collection. A message displays indicating that the collection operation was successful.

Monitor JTA

1. Click the instance node in the left pane under Servers for the server whose JTA usage you want to view. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Monitoring tab.
3. Click the JTA tab. JTA read-only statistics display in the right pane.

Monitor Server Security

1. Click the instance node in the left pane under Servers for the server whose security you want to monitor. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Monitoring tab.
3. Click the Security tab. The security data for this instance is displayed.

View Server Version

1. Click the instance node in the left pane under Servers for the server whose version you want to view. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Monitoring tab.
3. Click the Versions tab. The version data for this instance is displayed.

Monitor Server Clusters

1. Click the instance node in the left pane under Servers for the server whose clusters you want to monitor. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Monitoring tab.
3. Click the Cluster tab. The cluster data for this instance is displayed.

Deploy EJBs on a Server

1. Click the instance node in the left pane under Servers to select a server for EJB deployment. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Deployments tab. This displays the EJB tab.
3. Select one or more EJBs in the Available column that you want to deploy on the server.
4. Click the mover control to move the EJBs you selected to the Chosen column.

5. Click Apply to save your assignments.

Monitor All EJB Deployments on a Server

1. Click the instance node in the left pane under Servers to select a server for EJB monitoring. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Deployments tab. This displays the EJB tab.
3. Click the Monitor All EJBs text link. The Active EJBs table displays in the right pane showing all the EJBs deployed on this server.

Deploy Web Application Components on a Server

1. Click the instance node in the left pane under Servers to select a server for web-application deployment. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Deployments tab.
3. Click the Web Applications tab.
4. Select one or more web applications in the Available column that you want to deploy on the server.
5. Click the mover control to move the web application you selected to the Chosen column.
6. Click Apply to save your assignments.

Monitor All Web Application Components on a Server

1. Click the instance node in the left pane under Servers to select a server for web-application monitoring. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Deployments tab.
3. Click the Web Applications tab.
4. Click the Monitor All Active Web Applications text link. The web applications table displays in the right pane showing all the web applications deployed on this server.

Deploy Startup/Shutdown Classes on a Server

1. Click the instance node in the left pane under Servers to select a server for startup/shutdown class deployment. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Deployments tab.
3. Click the Startup/Shutdown tab.
4. Select one or more startup classes in the Available column that you want to deploy on the server.
5. Click the mover control to move the startup classes you selected to the Chosen column.
6. Click Apply to save your assignments.

7. Repeat steps 4, 5, and 6 using the Shutdown Class control to deploy shutdown classes on the server.

Assign JDBC Connection Pools to a Server

1. Click the instance node in the left pane under Servers to select a server for web-server assignment. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Services tab.
3. Click the JDBC tab.
4. Select one or more JDBC connection pools in the Available column that you want to assign to the server.
5. Click the mover control to move the JDBC connection pools you selected to the Chosen column.
6. Click Apply to save your assignments.

Assign WLEC Connection Pools to a Server

1. Click the instance node in the left pane under Servers to select a server for WLEC connection-pool assignment. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Services tab.
3. Click the WLEC tab.
4. Select one or more WLEC connection pools in the Available column that you want to assign to the server.
5. Click the mover control to move the WLEC connection pools you selected to the Chosen column.

6. Click Apply to save your assignments.

Monitor All WLEC Connection Pools on a Server

1. Click the instance node in the left pane under Servers to select a server for WLEC connection-pool monitoring. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Services tab.
3. Click the WLEC tab.
4. Click the Monitor All WLEC Connection Pools on This Server text link. The WLEC Connection Pools table displays in the right pane showing all the connection pools assigned to this server.

Assign XML Registries to a Server

1. Click the instance node in the left pane under Servers to select a sever for XML registry assignment. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Monitoring tab.
3. Click the XML tab.
4. Select a registry from the XML Registry drop-down list box.
5. Enter new values in the remaing fields or accept the default values.
6. Click Apply to save your assignment.

Assign Mail Sessions to a Server

1. Click the instance node in the left pane under Servers to select a server for mail session assignment. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Monitoring tab.
3. Click the Mail tab.
4. Select one or more mail sessions in the Available column that you want to assign to the server.
5. Click the mover control to move the mail sessions you selected to the Chosen column.
6. Click Apply to save your assignments.

Assign File T3s to a Server

1. Click the instance node in the left pane under Servers to select a server for file T3 assignment. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Services tab.
3. Click the File T3 tab.
4. Select one or more file T3s in the Available column that you want to assign to the server.
5. Click the mover control to move the file T3s you selected to the Chosen column.
6. Click Apply to save your assignments.

Changing Compilers in the WebLogic Server Console

1. Start the WebLogic Server Console.
2. Open the Servers folder in the Navigation Tree.
3. Select your server (myserver in a default installation) in the Servers folder.
4. Click the Configuration tab.
5. Click the Compilers tab and enter the full path of the sj.exe compiler in the Java Compiler text box. For example:`c:\visualcafe31\bin\sj.exe`.
6. Enter the full path to the JRE rt.jar library in the Append to classpath text box. For example: `\%WL_HOME%\jdk130\jre\lib\rt.jar`.
7. Click Apply.
8. Restart your server for the new Java Compiler and Append to classpath values to take effect

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	This attribute returns the name of the server.	The name can be up to 256 alphanumeric characters, but may not contain commas or spaces.	myserver
Machine	This attribute returns the name of the machine.	This attribute is an alpahnumeric string that cannot contain commas or spaces.	Null
Listen Port	This attribute sets the port on which this server listens for ssl messages.	Valid listen port	7001

Attribute	Description	Range of Values	Default Value
Administration Port	Specifies an SSL port that the Administration Server uses to receive all administrative communication. If you specify a value on an Administration Server, all Managed Servers use the port number (and SSL) to send information to the Administration Server. In addition, you must use this port number and SSL for any administrative commands. For example, you must use the port number to access the Administration Console on the Administration Server. If you do not specify a value, an Administration Server uses its default listen port to receive administrative communication.	Any unused port number from 1 to 65534.	Null

Attribute	Description	Range of Values	Default Value
Listen Address	This attribute allows you to set the listen address for the domain. **Note:** To resolve a DNS name to an IP address, Weblogic Server must be able to contact an appropriate DNS server or obtain the IP address mapping locally. Therefore, if you specify a DNS name for the listen address, you must either leave a port open long enough for the WebLogic Server instance to connect to a DNS server and cache its mapping or you must specify the IP address mapping in a local file. If you specify an IP address for ListenAddress and then a client request specifies a DNS name, WebLogic Server will attempt to resolve the DNS name, but if it cannot access DNS name mapping, the request will fail. *MBean:*	Valid listen address	Null

Attribute	Description	Range of Values	Default Value
External DNS Address	If your system includes an address translation firewall that sits between the clustered WebLogic Servers and a plug-in to a web server front-end, such as the Netscape (proxy) plug-in, set this attribute to the address used by the plug-in to talk to this server.		
Interface Address			Null

Attribute	Description	Range of Values	Default Value

WebLogic Plug-in Enabled	Set this attribute to true if the server instance will receive requests from a proxy plug-in or HttpClusterServlet.
	When WeblogicPluginEnabled is true, a call to getRemoteAddr will return the address of the browser client from the proprietary WL-Proxy-Client-IP header, instead of the web server.
	If the server instance is a member of a cluster that will receive proxied requests, set WeblogicPluginEnabled for the cluster, on the Cluster -->Configuration-->General tab.
	WeblogicPluginEnabled can be configured in ClusterMBean or ServerMBean. If specified in both ClusterMBean and ServerMBean, the value in ClusterMBean value takes precedence.
		Boolean
		Selected = True
		Unselected = False
			False

Cluster

Attribute	Description	Range of Values	Default Value
Cluster	This attribute specifies the cluster to which this server belongs.	Valid cluster name	none
Replication Group	This attribute specifies the replication group to which this cluster belongs.	String	none
Preferred Secondary Group	This attribute specifies the preferred secondary group for this cluster.	String	none
Cluster weight	This attribute specifies the percentage of work to be performed by this server.	Integer	100

HTTP

Attribute	Description	Range of Values	Default Value
Default Web Application	This attribute reports the name of the default web application for this server.	String	DefaultWebApp_myserver
FrontendHost	This attribute sets the HTTP frontendHost Provides a method to ensure that the webapp will always have the correct HOST information, even when the request is coming through a firewall or a proxy. If this parameter is configured, the HOST header will be ignored and the information in this parameter will be used in its place.	String	Null
FrontendHTTPPort	This attribute sets the frontendHTTPPort Provides a method to ensure that the webapp will always have the correct PORT information, even when the request is coming through a firewall or a proxy. If this parameter is configured, the HOST header will be ignored and the information in this parameter will be used in its place.	Valid port Number	0

Attribute	Description	Range of Values	Default Value
FrontendHTTPSPort	This attribute sets the frontendHTTPSPort Provides a method to ensure that the webapp will always have the correct PORT information, even when the request is coming through a firewall or a proxy. If this parameter is configured, the HOST header will be ignored and the information in this parameter will be used in its place.	Valid port number	0
Post Timeout Secs	This attribute sets the timeout (in seconds) that WebLogic Server waits between receiving chunks of data in an HTTP POST data. Used to prevent denial-of-service attacks that attempt to overload the server with POST data.	Integer	0
Default Server Name	When WebLogic Server redirects a request, it sets the host name returned in the HTTP response header with the string specified with Default Server Name. Useful when using firewalls or load balancers and you want the redirected request from the browser to reference the same host name that was sent in the original request.	String	Null

Attribute	Description	Range of Values	Default Value

Send Server Header	If set to false, the server name is not sent with the HTTP response. Useful for wireless applications where there is limited space for headers.	Boolean
		True = enabled
		False = not enabled
			True

WAP Enabled	When selected, the session ID no longer includes JVM information. This may be necessary when using URL rewriting with WAP devices that limit the size of the URL to 128 characters. Selecting WAP Enabled may affect the use of replicated sessions in a cluster.	Enabled Disabled	not Enabled
Max Post Time	This attribute sets the time (in seconds) that WebLogic Server waits for chunks of data in an HTTP POST data.	Integer A size of less than 0 means that the size is unlimited.	-1
Max Post Size	This attribute sets the size of the maximum chunks of data in an HTTP POST data.	Integer A size of less than 0 means that the size is unlimited.	-1

Enable Keepalives	This attribute sets whether or not HTTP keep-alive is enabled	Boolean
		True = enabled
		False = not enabled
			Selected

Duration	The number of seconds that WebLogic Server waits before closing an inactive HTTP connection.	Integer	30

Attribute	Description	Range of Values	Default Value
HTTPS Duration	The number of seconds that WebLogic Server waits before closing an inactive HTTPS connection.	Integer	60

SSL

Attribute	Description	Range of Values	Default Value

Enabled	This attribute enables server-to-server SSL connections. It is disabled to override automatic SSL between servers.	Boolean
		True = selected
		False = not selected
			Selected

Listen Port	This attribute specifies the dedicated port on which WebLogic Server listens for SSL connections.	Valid listen port	7002
Server Key File Name	The name of the private key file for WebLogic Server.	String	config/mydomain/demokey.pem
Server Certificate File Name	This attribute designates the name of the private key file for your WebLogic Server.	String	config/mydomain/democert.pem
Server Certificate Chain File Name	This attribute designates the full directory location of the digital certificate for your WebLogic Server.	String	config/mydomain/ca.pem

Client Certificate Enforced	This attribute is set to true to enforce the client certificate.	Boolean
		True = selected
		False = not selected
			Not Selected

Attribute	Description	Range of Values	Default Value
Trusted CAFile Name	This attribute designates the name of the file that contains the digital certificate for the certificate authority trusted by WebLogic Server.	This attribute can be a file that contains a single digital certificate or a chain of digital certificates for certificate authorities.	trusted-ca.pem
Certificate Authenticator	This attribute specifies the certificate authenticator to be used to determine the validity of the certificate.	Valid authenticator	Null
Key Encrypted	Specifies whether or not the private key for the WebLogic Server has been encrypted with a password. ■ If the attribute is set to true, the private key requires a pass phrase be supplied to use the key. ■ If the attribute is set to false, the private key is unencrypted and may be used without providing a pass phrase.	Selected = true Not Selected = false	Not Selected
Use Java	This attribute enables the use of native Java libraries. WebLogic Server provides a pure-Java implementation of the SSL protocol. Native libraries enhance the performance for SSL operations on the Solaris, Windows NT, and IBM AIX platforms.	Selected = true Not Selected = false	Selected

Attribute	Description	Range of Values	Default Value
Handler Enabled	Specifies whether or not WebLogic Server rejects SSL connections that fail client authentication for one of the following reasons: ■ The requested client digital certificate was not furnished. ■ The client did not submit a digital certificate ■ The digital certificate from the client was not issued by a certificate authority specified by the `Trusted CA Filename` attribute. By default, the SSL Handler allows one WebLogic Server to make outgoing SSL connections to another WebLogic Server. For example, an EJB in WebLogic Server may open an HTTPS stream on another web server. With the `HandlerEnabled` attribute enabled, WebLogic Server acts as a client in an SSL connection.	Selected = true Not Selected = false	Selected

Attribute	Description	Range of Values	Default Value
Export Key Lifespan	This attribute specifies the number of times WebLogic Server can use an exportable key between a domestic server and an exportable client before generating a new key. The more secure you want WebLogic Server to be, the fewer times the key should be used before generating a new one.	The maximum value is specified as `java.lang.Integer.` *`MAX_VALUE`*. The minimum value is 1.	500
Login Timeout Millis	This attribute sets the number of milliseconds that WebLogic Server should wait for an SSL connection before timing out. SSL connections take longer to negotiate than regular connections. If clients are connecting over the Internet, raise the default number to accommodate additional network latency. A value of 0 disables the attribute.	The maximum value is specified as `java.lang.Integer.` *`MAX_VALUE`*. The minimum value is 1.	25000
Certificate Cache Size	This attribute sets the number of digital certificates held that have not been redeemed by tokens.	The maximum value is specified as `java.lang.Integer.` *`MAX_VALUE`*. The minimum value is 1.	3
Hostname Verification Ignored	This attribute disables the installed Hostname Verifier when WebLogic Server is acting as a client to another WebLogic Server.		Not Selected

Attribute	Description	Range of Values	Default Value
Hostname Verifier	This attribute sets the name of the Java class that implements the Hostname Verifier interface.	String	Null
Two-way SSL Enabled	This attribute allows the user to set Two-way SSL in order to make mutual authentication optional. If a client certificate is presented then mutual authentication is done, if not, the connection is accepted without requiring a client certificate.	Boolean Selected = Two-way SSL is enabled. Not Selected = Two-way SSL is disabled.	Not Selected

Tuning

Attribute	Description	Range of Values	Default Value
Socket Readers	This attribute sets the percentage of threads to be available as socket readers.	Integer	33%
Login Timeout	This attribute sets the login timeout for the server.	Integer	1000 ms
Accept Backlog	This attribute sets the number of connections available for backlog. To increase the number of connections to be processed, increase this number.	Integer	50

Reverse DNS Allowed	This attribute determines whether reverse DNS lookup is allowed on this server.	Boolean
		True = selected
		False = not selected
			Not Selected

Enable Native IO	This attribute determines whether or not native I/O is enabled for the server.	Boolean Enabled = selected Not Enabled = not selected	Selected

Show only Workspace User Keys	This attribute determines whether or not the User Keys will be shown.	Boolean
		True= selected
		False = not selected
			Not Selected

Attribute	Description	Range of Values	Default Value
Enable Default JMS Connection Factories	This attribute determines whether the default JMS connection factories will be enabled for this server.	Boolean Enabled = selected Not Enabled = not selected	Selected
Enable Tunneling	This attribute determines whether tunneling will be enabled for this server.	Boolean Enabled = selected Not Enabled = not selected	Not Selected
Tunneling Client Ping	This attribute sets the time, in seconds, that the server will wait before pinning the client.	Integer	45
Tunneling Client Timeout	This attribute sets the time, in seconds, that the server will wait before timing out.	Integer	40

Protocols

Attribute	Description	Range of Values	Default Value
Default Protocol	This attribute sets the default protocol to be used by this server.	List	t3
Default Secure Protocol	This attribute sets the default secure protocol to be used by this server.	List	t3s
T3 Max Message Size	This attribute sets the size, in bytes, of the maximum message.	Integer	10000000
T3 Message Timeout	This attribute sets the time, in seconds, after which the message will time out.	Integer	480
HTTP Max Message Size	This attribute sets the size, in bytes, of the maximum message.	Integer	10000000
HTTP Message Timeout	This attribute sets the time, in seconds, after which the message will time out.	Integer	480
Enable IIOP	This attribute determines whether IIOP will be enabled for this server.	Boolean Enabled = selected Not Enabled = not selected	Selected

Attribute	Description	Range of Values	Default Value
IIOP Max Message Size	This attribute sets the size, in bytes, of the maximum message.	Integer	10000000
IIOP Message Timeout	This attribute sets the time, in seconds, after which the message will time out.	Integer	480
Default IIOPPassword	This attribute sets the default IIOP password, if any, for this server.	Valid password	*****
Default IIOPUser	This attribute sets the default IIOP user, if any, for this server	User name as set by the administrator	guest

Compilers

Attribute	Description	Range of Values	Default Value
Java Compiler	This attribute sets the java compiler to be used by the server.	Valid java compiler	javac
Prepend to Classpath	This attribute adds to the beginning of the classpath.	Class path information	Null
Append to Classpath	This attribute adds to the end of the classpath	Class path information	Null
Extra rmic Options	This attribute sets rmic options for this server.	rmic options	Null

Monitoring

General

Attribute	Description	Range of Values	Default Value
State	This attribute describes the state of the server.		Running
Activation Time	This attribute reports the time of the last activation	Date	Date of last activation

Performance

Attribute	Description	Range of Values	Default Value
Request Throughput	This attribute reports the rate at which requests are processed by this server.		N/A
Requests Waiting	This attribute reports the number of requests waiting to be serviced.		N/A
Request Wait Time	This attribute reports the amount of time the oldest request has been waiting to be serviced		
Total Requests Serviced	This attribute reports the total requests processed by this server.		
Memory Usage	This attribute shows the amount of memory being used by this server.		N/A

Memory

Attribute	Description	Range of Values	Default Value
Memory Usage	This attribute returns the amount of memory being used by the server.	Integer	n/a
Memory Allocated	This attribute sets the amount of memory allocated for this instance.	Integer	66846720

Clusters

Attribute	Description	Range of Values	Default Value
Alive Servers	This attribute returns the number of servers currently alive.		n/a
Resend Requests	This attribute returns the number of requests for resends of information.		n/a
Fragments Sent	This attribute returns the number of fragments of information sent.		n/a
Fragments Received	This attribute returns the number of fragments of information received.		n/a
Multicast Messages Lost	This attribute returns the number of multicast messages lost.		n/a
Server Names	This attribute returns the names of the servers currently alive.		n/a
Primary	This attribute returns the primary distribution names of the servers.		n/a
Secondary Distribution Names	This attribute returns the secondary distribution names of the servers.		n/a

Security

Attribute	Description	Range of Values	Default Value
Total Users Locked Out	This attribute returns the number of users who have been locked out of the server.	Integer	0
Invalid Logins Total	This attribute returns the number of invalid login attempts made.	Integer	0
Total Login Attempts While Locked	This attribute returns the number of login attempts (both valid and invalid) made while the server was locked.	Integer	0
Total Users Unlocked	This attribute returns the number of users whose logins have been unlocked.	Integer	0
Invalid Logins High	This attribute returns the number of invalid logins.	Integer	0
Locked Users	This attribute returns the number of users whose logins have been locked.	Integer	0

JMS

Attribute	Description	Range of Values	Default Value
Connections	This attribute reports the number of current connections.	Integer	0
Connections High	This attribute reports the high water number of connections.	Integer	0
Connections Total	This attribute reports the total number of connections.	Integer	0
JMS Servers	This attribute reports the number of JMS servers currently running.	Integer	0
JMS Servers High	This attribute reports the high water number of JMS servers.	Integer	0
JMS Servers Total	This attribute reports the total number of JMS servers.	Integer	0

JTA

Attribute	Description	Range of Values	Default Value
Total Transactions	The total number of transactions processed. This total includes all committed, rolled back and heuristic transaction completions.	Integer	0.
Total Committed	The number of committed transactions.	Integer	0.
Total Rolled Back	The number of transactions that have been rolled back.	Integer	0
Timeout Rollbacks	The number of transactions that were rolled back due to a timeout expiration.	Integer	0
Resource Rollbacks	The number of transactions that were rolled back due to a resource error.	Integer	0
Application Rollbacks	The number of transactions that were rolled back due to an application error.	Integer	0

Attribute	Description	Range of Values	Default Value
System Rollbacks	The number of transactions that were rolled back due to a system error.	Integer	0
Total Heuristics	The number of transactions that completed with a heuristic status.	Integer	0
Total Transactions Abandoned	The number of transactions abandoned prior to being committed.	Integer	0
Average Commit Time	The total number of milliseconds for all committed transactions.	Integer	0

Versions

Attribute	Description	Range of Values	Default Value
WebLogic Version	This attribute returns the version of WebLogic Server currently in use.		Version currently in use.
JDK vendor	This attribute returns the name of the vendor of the JDK in use on the server.		null
JDK version	This attribute returns the version of the JDK in use on the server.		null
Operating System	This attribute returns the operating system for the server.		null
OS version	This attribute returns the operating system version for the server.		null
J2EE Version 1.2 Only Mode Enabled			Not Selected
J2EE Version 1.3 Warning Enabled			Not Selected

Entities

Attribute	Description	Range of Values	Default Value
Total Current Entries	This attribute reports the total number of current entries for this server.		0
Total Current Persistent Entries	This attribute reports the total number of current entries for this server that have been made persistent.		0
Average Percentage Persistent	This attribute reports the average number of entries for this server that have been made persistent.		0
Total Transient Current Entries			0
Average Percentage Transient	This attribute reports the average percentage of transient entries for this server.		0
Minimum Entry Timeout			0
Maximum Entry Timeout			0
Average Timeout			0
Total Current Session Entries			0
Total Persistent Current Session Entries			0

Attribute	Description	Range of Values	Default Value
Average Percentage Session Persistent			0
Total Transient Current Session Entries			0
Average Percentage Session Transient			0
Minimum Entry Session Timeout			0
Maximum Entry Session Timeout			0
Average Session Timeout			0
Maximum Entry Memory Size			0
Minimum Entry Memory Size			0
Average Per Entry Memory size			
Average Per Entry Disk Size			
Total Number of Rejections			
Total Size of Rejections			
Percent Rejected			
Total Number of Renewals			
Total Historical Current Entries			
Total Historical Persistent Current Entries			

Attribute	Description	Range of Values	Default Value
Historical Average Percent Persistent			
Historical Total Transient Current Entries			
Historical Average Percent Transient			
Historical Minimum Entry Timeout			
Historical Maximum Entry Timeout			
Historical Average Entry Timeout			
Historical Maximum Entry Memory Size			
Historical Minimum Entry Memory Size			
Historical Average Per Entry Memory Size			
Historical Average Per Entry Dixk Size			
Historical Total Number of Rejections			
Historical Total Size of Rejections			
Historical Percent Rejected			
Historical Total Number of Renewals			
Current Maximum Entry Memory Size			

Attribute	Description	Range of Values	Default Value
Current Minimum Entry Memory Size			
Current Average Per Entry Memory Size			
Current Average Per Entry Disk Size			
Current Memory Usage			
Current Disk Usage			

Logging

General

Attribute	Description	Range of Values	Default Value

File Name	This attribute describes the name of the file used to write the log messages to disk	The file name may contain up to 256 alphanumeric characters.
		To include a time or date stamp in the file name when the log file is rotated, add `java.text.SimpleDateFormat` variables. Surround each variable with percentage (%) characters.
		For example, if you enter the following value in the File Name field:
		`myserver_%yyyy%_%MM%_%dd%_%hh%_%mm%.log`
		the server's log file will be named `myserver_yyyy_MM_dd_hh_mm.log`
		When the server instance rotates the log file, the rotated file name contains the date stamp. For example, if the server instance rotates its local log file on 2 April, 2003 at 10:05 AM, the log file that contains the old log messages will be named:
		`myserver_2003_04_02_10_05.log`
			`serverName``.log`

Attribute	Description	Range of Values	Default Value

Log to Stdout	Enables the server to send messages to standard out in addition to the log file. Use `Debug to Stdout` and `Stdout severity threshold` to determine the type of messages that the server sends to standard out.	Boolean
		True = selected
		False = not selected
			Selected

Debug to stdout	Determines whether the server sends messages of the `Debug` severity to standard out in addition to the log file. You must enable `Log to Stdout` for this property to be relevant.	Boolean
		True = selected
		False = not selected
			Not Selected

Attribute	Description	Range of Values	Default Value
Stdout severity threshold	The minimum severity of a message that the server sends to standard out. You must enable `Log to Stdout` for this value to be relevant.	■ `Info`. Used for reporting normal operations. ■ `Warning`. A suspicious operation or configuration has occurred but it may not have an impact on normal operation. ■ `Error`. A user error has occurred. The system or application is able to handle the error with no interruption, and limited degradation, of service. ■ `Notice`. An `Info` or `Warning`-level message that is particularly important for monitoring the server. ■ `Critical`. A system or service error has occurred. The system is able to recover but there might be a momentary loss, or permanent degradation, of service. ■ `Alert`. A service is in an unusable state while other parts of the system continue to function. Automatic recovery is not possible; the immediate attention of the administrator is needed to resolve the problem. ■ `Emergency`. The server is in an unusable state. This severity indicates a severe system failure or panic.	`Error`

Rotation

Attribute	Description	Range of Values	Default Value
Rotation Type	Criteria for moving old log messages to a separate file. After the server renames a file, subsequent messages accumulate in a new file with the name that you specified in File Name on the Logging > General tab.	■ `None`. Messages accumulate in a single file. You must erase the contents of the file when the size is unwieldy. ■ `by Size`. When the log file reaches the size that you specify in `File Min Size`, the server renames the file as `FileName.`*`n`*. ■ `by Time`. At each time interval that you specify in `File Time Span`, the server renames the file as `FileName.`*`n`*.	`None`
File Min Size	This attribute sets the threshold at which a new log file is created. Relevant only if `Rotation Type` is `by Size`.	Integer	500

Attribute	Description	Range of Values	Default Value
Rotation Time	Determines the start time for a time-based rotation sequence. At the time that this value specifies, the server renames the current log file as `FileName.`*n*. Thereafter, the server renames the log file at an interval that you specify in `File Time Span`. You can create a recurring start time such as "every Monday at 09:00" or a non-recurring start time such as "9 January, 2002, 09:00." Relevant only if `Rotation Type` is `by Time`.	Use the following `java.text.SimpleDateFormat` format to specify a date and time: `k:mm`. For information about this format, refer to the J2EE Javadoc. If the date that you specify has already past, then the rotation starts immediately.	00:00 By default, the rotation cycle begins on the first minute of each day (12:00 AM), but you can configure the rotation cycle to start on a specific day of the week or on a specific date.
File Time Span	This attribute sets the threshold at which a new log file is created. Relevant only if `Rotation Type` is `by Time`.	Integer	24

Number of Files Limited	Indicates whether a server will limit the number of log files that it creates when it rotates the log. The limit is based on the value in `File Count`.	Boolean
		True = selected
		False = not selected
			Not Selected

File Count	The maximum number of log files that the server creates when it rotates the log. Only valid if `Number of Files Limited` is `true` and `Rotation Type` is either `by Size` or `by Time`.	Integer	7

Domain

Attribute	Description	Range of Values	Default Value
Log to Domain Logfile	Determines whether this server sends messages to the domain log (in addition to keeping its own log).	Boolean Enabled = selected Not Enabled = not selected	Selected
Domain Log Filter	Determines which messages this server sends to the domain log. If you specify `none`, the server sends all messages of severity `Error` and higher. This property is relevant only if `Log To Domain Logfile` is enabled.	Contains all Domain Log Filters that have been defined for the domain. A server can user only one Domain Log Filter.	`none`

HTTP

Attribute	Description	Range of Values	Default Value

Enable Logging	Determines whether the server logs HTTP requests. The requests are saved in a separate HTTP log file.	Boolean
		Enabled = selected
		Disabled = not selected
			Selected

Logfile Name	This attribute sets the complete name of the HTTP logfile.	Fully qualified logfile name	./config/mydomain/logs/access.log
Format	Specifies the format of the HTTP log file. Both formats are defined by the W3C. With the extended log format, you use server directives in the log file to customize the information that the server records.	Common, Extended	Common

Attribute	Description	Range of Values	Default Value

Log Time in GMT	Writes the time stamp of HTTP log messages in Greenwich Mean Time (GMT) regardless of whether the host computer has specified a different local time zone.
	Use this feature to comply with the W3C specification for Extended Format Log Files. This specification states that all time stamps for Extended Format log entries should be in GMT:
	http://www.w3.org/TR/WD-logfile.html.
	This attribute applies only if you have specified the `extended` message format.
		Boolean
		Enabled = selected
		Disabled = not selected
			not selected

Log Buffer Size	The maximum size (in kilobytes) of the buffer that stores HTTP requests. When the buffer reaches this size, the server writes the data to the HTTP log file. Use the `Flush Every` property to determine the frequency with which the server checks the size of the buffer.	Integer, in kilobytes 0-1024	8
Max Log File SizeK Bytes	This attribute sets he maximum size of the log file after which the log file rotates.	Integer, in kilobytes. 0 means that there is no limit to the size the file can reach.	0

Attribute	Description	Range of Values	Default Value
Rotation Type	Criteria for moving old HTTP requests to a separate log file. After the server renames a file, subsequent messages accumulate in a new file with the name that you specified in `Logfile Name`.	■ `size`. When the log file reaches the size that you specify in `Max Log File SizeK Bytes`, the server renames the file as *`LogfileName.n`*. ■ `date`. At each time interval that you specify in `Rotation Period`, the server renames the file as *`LogFileName.n`*.	Size
Rotation Period	The interval (in minutes) at which the server saves old HTTP requests to another log file. This value is relevant only if you use the date-based rotation type.	Integer, in minutes	2147483647
Flush Every	The interval (in seconds) at which the server checks the size of the buffer that stores HTTP requests. When the buffer exceeds the size that is specified in the Logfile BufferK Bytes property, the server writes the data in the buffer to the HTTP request log file.	Integer, in seconds	60

Attribute	Description	Range of Values	Default Value
Rotation Time	Determines the start time for a time-based rotation sequence. At the time that this value specifies, the server renames the current log file as `*LogFileName.n*`. Thereafter, the server renames the log file at an interval that you specify in LogRotationPeriodMins. You can create a recurring start time such as "every Monday at 09:00" or a non-recurring start time such as "9 January, 2002, 09:00."	Use the following `java.text.SimpleDateFormat` format to specify a date and time: `MM-dd-yyyy-k:mm:ss`. For information about this format, refer to the J2EE Javadoc. If the date that you specify has already past, then the day in week (`E`), hour in day (`H`), minute in hour (`m`) and second in minute (`s`) are used with the current date to recalculate a start time.	Null

JDBC

Attribute	Description	Range of Values	Default Value

Enable JDBCLogging	Determines whether the server logs JDBC transactions. The transactions are saved in a separate JDBC log file.	Boolean
		True = selected
		False = not selected
			Not Selected

JDBCLog File Name	This attribute returns the name of the JDBC log file.		Null

JTA

Attribute	Description	Range of Values	Default Value
Transaction Log File Prefix	This attribute sets the prefix for the transaction log files.		./

Debugging

Attribute	Description	Range of Values	Default Value

LogRemoteExceptions	This attribute is set to true if remote exceptions are to be reported to the server log.	Boolean
		True = selected
		False = not selected
			Not Selected

Instrument Stack Traces	IIf set to true, exception messages include the server-side stack trace.	Boolean
		True = selected
		False = not selected
			Selected

Deployments

EJB

Attribute	Description	Range of Values	Default Value
Choose EJBs for this Server	Select deployed EJBs from a drop-down list of available EJBs. To select more than one EJB from the available list, hold down the Control key while selecting the EJBs.	One or more available EJBs.	Null

Web Applications

Attribute	Description	Range of Values	Default Value
Web App Component	This attribute allows the user to select the available Web App components.	List	Null

Connector

Attribute	Description	Range of Values	Default Value
Connector Components	This attribute allows the user to select the available Connector components.	List	Null

Virtual Hosts

Attribute	Description	Range of Values	Default Value
Virtual Hosts	This attribute allows the user to select the available Virtual.Host components.	List	Null

Remote Start

:

Attribute	Description	Range of Values	Default Value
Java Home	Root directory where the JDK is installed.		Null
BEA Home	Root directory where all BEA products and licenses are installed.		Null
Root Directory	Root directory where WebLogic Server is installed.		Null
Class Path	The classpath for starting the Managed Server. At a minimum, specify: `$WL_HOME:$BEA_HOME:$WL_HOME/lib/weblogic.jar:$WL_HOME/lib/weblogic_sp.jar:$JAVA_HOME/lib/rt.jar`		Null
Security Policy File	The JVM's security policy file is used by default. There is also a WebLogic security policy file available, `weblogic/lib/weblogic.policy`.		The JVM's security policy file is used by default.

Attribute	Description	Range of Values	Default Value
Arguments	Arguments to use when starting the Managed Server. For example, you can set the maximum and minimum Java heap memory. Entering `-ms64m` and `-mx64m` options would specify a default allocation of 64 megabytes of Java heap memory to the WebLogic Server. **Note:** Do not specify server name, user name or password. Do not specify the address and port of the Administration Server.		Null
Security Policy File	The JVM's security policy file is used by default. There is also a WebLogic security policy file available, located at weblogic/lib/weblogic.policy.		Null

Services

JDBC

Attribute	Description	Range of Values	Default Value
JDBC Connection Pool	This attribute allows the user to select the connection pool to use in this configuration	List	Null

WLEC

Attribute	Description	Range of Values	Default Value
WLEC Connection Pool	This attribute allows the user to select the WLEC connection pool to use in this configuration	List	Null

Jolt

Attribute	Description	Range of Values	Default Value
Jolt Connection Pool	This attribute allows the user to select the WLEC connection pool to use in this configuration	List	Null

JMS

Attribute	Description	Range of Values	Default Value
JMS Server	This attribute allows the user to select the JMS server to use in this configuration	List	Null
JMSConnection Factory	This attribute allows the user to select the JMSConnection Factory to use in this configuration	List	Null
JMSDestination	This attribute allows the user to select the JMSDestination to use in this configuration	List	Null

Attribute	Description	Range of Values	Default Value
JMS Thread Pool Size	This value specifies the size of the JMS thread pool. **Note:** Incoming RMI calls execute in the JMS execute queue/thread pool, if one exists; otherwise, they execute in the default execute queue. Additional executes (work that cannot be completed in the initial RMI thread) are executed in the default execute queue. The difference in setting up a JMS-specific thread pool is that JMS will not be starved by other execute threads and vice versa.	Integer, Greater than or equal to 0 If it is set to less than 5, it will automatically be set to 5.	The default size on the client is 0 (no JMS thread pool). The default size on the server is 15 (the server always has a JMS thread pool).

XML

Attribute	Description	Range of Values	Default Value
XML Registry	This attribute allows the user to select the xml registry to use in this configuration	List	Null
Cache Memory Size	This attribute allows the user to set the amount of disk space available for the XML Entity Cache.	Integer (KBytes)	0
Cache Timeout Interval	This attribute allows the user to set the time interval after which the cache will time out.	Integer (Seconds)	120

Mail

Attribute	Description	Range of Values	Default Value
Mail Session	This attribute allows the user to select the mail session to use in this configuration	List	Null

File T3

Attribute	Description	Range of Values	Default Value
File T3	This attribute allows the user to select from the available File T3 entities.	List	Null

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for optional user-supplied information.	The value must be an alphanumeric string.	Null

79 Servlet

By default, this pane allows the user to sort the objects by the following criteria:

- Server
- Name
- Reloads
- Invocation Total
- Pool Max Capacity
- Execution Time Total
- Execution Time High
- Execution Time Low
- Execution Time Average

80 Servlet Runtime

Monitor Servlets

Attribute	Description	Range of Values	Default
Server	Specifies the name of the server on which this servlet is deployed		
Name	Specifies the name of the servlet		
Reloads	Specifies the number of times the servlet has been reloaded.		
Invocation Total	Specifies the number of times that the servlet is invoked		
Pool Max Capacity	Specifies the maximum capacity of this servlet for single thread model servlets		
Execution Time Total	Specifies the total execution time, in milliseconds, for the servlet		
Execution Time High	Specifies the highest execution time, in milliseconds, for the servlet		
Execution Time Low	Specifies the lowest execution time, in milliseconds, for the servlet		

Attribute	Description	Range of Values	Default
Execution Time Average	Specifies the average execution time, in milliseconds, for the servlet		

81 Servlet Session

Monitor Servlets

Attribute	**Description**	**Range of Values**	**Default**
Server	Specifies the name of the server on which this servlet is deployed		
Name	Specifies the name of the servlet		
Reloads	Specifies the number of times the servlet has been reloaded.		
Invocation Total	Specifies the number of times that the servlet is invoked		
Pool Max Capacity	Specifies the maximum capacity of this servlet for single thread model servlets		
Execution Time Total	Specifies the total execution time, in milliseconds, for the servlet		
Execution Time High	Specifies the highest execution time, in milliseconds, for the servlet		
Execution Time Low	Specifies the lowest execution time, in milliseconds, for the servlet		

Attribute	Description	Range of Values	Default
Execution Time Average	Specifies the average execution time, in milliseconds, for the servlet		

82 Web Application Deployment Descriptor Editor Help

Configuring the Session Descriptor

The Session Descriptor configures HTTP session-related parameters.

1. If it does not exist in the left pane, create the WebApp Ext node:
    a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

    b. Select Configure a new WebAppExtDescriptor

    c. In the right pane, enter a description of the Web Application in the Description field (optional).

    d. Enter the version of WebLogic Server that your application is running on (optional).

    e. Click Create.

    f. Expand the WebApp Ext node in the left panel.
2. Click on the Session Descriptor node in the left panel.
3. Fill in the fields in the right panel according to the information in Session Descriptor table.

4. Click Apply.

Configuring the JSP Descriptor

The JSP Descriptor configures JSP-related parameters.

1. If it does not exist in the left pane, create the WebApp Ext node:
   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)
   b. Select Configure a new WebAppExtDescriptor
   c. In the right pane, enter a description of the Web Application in the Description field (optional).
   d. Enter the version of WebLogic Server that your application is running on (optional).
   e. Click Create.
   f. Expand the WebApp Ext node in the left panel.
2. Click on the JSP Descriptor node in the left panel.
3. Fill in the fields in the right panel according to the information in JSP Descriptor table.
4. Click Apply.

Configuring Security Role Assignments

Security role assignments map a security role to one or more principals. The principals must be defined in your security realm.

1. If it does not exist in the left pane, create the WebApp Ext node:
   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

   b. Select Configure a new WebAppExtDescriptor

   c. In the right pane, enter a description of the Web Application in the Description field (optional).

   d. Enter the version of WebLogic Server that your application is running on (optional).

   e. Click Create.

   f. Expand the WebApp Ext node in the left panel.

2. Right click on the Security Role Assignment node in the left panel and select Configure a new SecurityRoleAssignment.

3. Select a security role from the Role drop-down list.

4. Add one or more Principal Names to the text box, one per line. The principals must be valid in your security realm.

5. Click Create.

Configuring Character Set Parameters

You can define codeset behavior for non-unicode operations by configuring Character Set Parameters. You can also specify a mapping of Java character sets to IANA character sets. For more information, see Determining the Encoding of an HTTP Request at `http://e-docs.bea.com/wls/docs61/webapp/components.html#encoding-http`, and Mapping IANA Character Sets to Java Character Sets at `http://e-docs.bea.com/wls/docs61/webapp/components.html#map-iana`.

1. If it does not exist in the left pane, create the WebApp Ext node:

   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

   b. Select Configure a new WebAppExtDescriptor

   c. In the right pane, enter a description of the Web Application in the Description field (optional).

   d. Enter the version of WebLogic Server that your application is running on (optional).

   e. Click Create.

   f. Expand the WebApp Ext node in the left panel.

2. Right click on the WebApp Ext node in the left panel and select Configure a new Charset Params.

3. Create Input Charset Descriptors

   a. Expand the Charset Params node.

   b. Right click on the Input Charset Descriptors node and select Configure a new InputCharsetDescriptor.

   c. Enter the path that applies to this Charset Descriptor in the Resource Path field.

   d. Enter a valid Java character set name in the Java Charset Name field.

   e. Click Create

4. Create a Character Set Mapping

   a. Expand the Charset Params node.

   b. Right click on the Charset Mappings node and select Configure a new CharsetMapping.

   c. Enter a valid IANA character set name in the IANA Charset Name field.

   d. Enter a valid Java character set name in the Java Charset Name field.

   e. Click Create

Configuring a Reference Descriptor

The Reference Descriptor maps the JNDI name of a server resource to a name used in the Web application. The Resource Description panel maps a resource, for example, a DataSource, to its JNDI name. The Ejb Reference panel maps an EJB to its JNDI name.

1. If it does not exist in the left pane, create the WebApp Ext node:

   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

   b. Select Configure a new WebAppExtDescriptor

   c. In the right pane, enter a description of the Web Application in the Description field (optional).

   d. Enter the version of WebLogic Server that your application is running on (optional).

   e. Click Create.

   f. Expand the WebApp Ext node in the left panel.

2. Right click on the WebApp Ext node in the left panel and select Configure a new Reference Descriptor.

3. Create Resource Descriptions. You can create one or more Resource Descriptions.

   a. Expand the Resource Descriptor node.

   b. Right click on the Resource Descriptions node and select Configure a new ResourceDescription.

   c. Enter the name of an object from the JNDI tree in the Jndi Name field.

   d. Select a Resource Reference from the Resource Reference drop-down list. The Resource Reference should be already configured under the Resource Refs tab.

   e. Click Create

4. Create an EJB Reference Description

   a. Expand the Resource Descriptor node.

   b. Right click on the Ejb Reference Descriptions node and select Configure a new EjbReferenceDescription.

   c. Enter the name of an object from the JNDI tree in the Jndi Name field.

   d. Select an EJB Reference from the Ejb Reference drop-down list. The EJB Reference should be already configured under the Ejb refs tab.

   e. Click Create

Configuring a Container Descriptor

Enable Check Auth on Forward when you want to require authentication of forwarded requests from a servlet or JSP.

1. If it does not exist in the left pane, create the WebApp Ext node:
   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)
   b. Select Configure a new WebAppExtDescriptor
   c. In the right pane, enter a description of the Web Application in the Description field (optional).
   d. Enter the version of WebLogic Server that your application is running on (optional).
   e. Click Create.
   f. Expand the WebApp Ext node in the left panel.
2. Right click on the WebApp Ext node and select Configure a new Container Descriptor.
3. To enable authentication of forwarded requests, check the Check Auth on Forward Enabled box.
4. To configure redirect behavior, check the redirect-with-absoute-url box.
5. Click Create.

Web App Descriptor

Attribute	Description	Range of Values	Default Value
Description	A text description of the servlet.	String	none
Display Name	This element is not used by WebLogic Server.	String	none
Large Icon File Name	Specifies the location for a large (32x32 pixel) `.gif` or `.jpg` image used to represent the Web application in a GUI tool. Currently, this element is not used by WebLogic Server.	String	none
Small Icon File Name	Specifies the location for a small (16x16 pixel) `.gif` or `.jpg` image used to represent the Web application in a GUI tool. Currently, this is not used by WebLogic Server.	String	none
Distributable	This element is not used by WebLogic Server.	Empty	none

Filters

Attribute	Description	Range of Values	Default Value
Filter Name	The name for this filter	String	MyFilter
Description	A text description of the servlet.	String	none
Display Name	This element is not used by WebLogic Server.	String	none
Small Icon File Name	This element is not used by WebLogic Server.	String	none
Large Icon File Name	This element is not used by WebLogic Server.	String	none
Filter Class	The fully-qualified class name of the filter.	Class name	none

Filter Mappings

Attribute	Description	Range of Values	Default Value
Filter	The name of the filter to which you are mapping a URL pattern. This name corresponds to the name you assigned a filter in a `<filter-name>` element.	String	none

Url Pattern	Describes a pattern used to resolve URLs. The portion of the URL after the `http://host:port` + `WebAppName` is compared to the `<url-pattern>` by WebLogic Server. If the patterns match, the filter mapped in this element will be called.
	Example patterns:
	`/soda/grape/*`
	`/foo/*`
	`/contents`
	`*.foo`
	The URL must follow the rules specified in Section 10 of the Servlet 2.2 Specification.
		String	none

Attribute	Description	Range of Values	Default Value
Servlet	The name of a servlet which, if called, causes this filter to execute.	A valid servlet name defined in this deployment descriptor.	none

Listeners

Attribute	Description	Range of Values	Default Value
Listener Class Name	Name of the class that responds to a Web Application event.	Class name	none

Servlets

Attribute	Description	Range of Values	Default Value
Description	A text description of the servlet.	String	None
Display Name	This element is not used by WebLogic Server.	String	None
Small Icon File Name	This element is not used by WebLogic Server.	String	None
Large Icon File Name	This element is not used by WebLogic Server.	String	None
Servlet Name	Defines the canonical name of the servlet, used to reference the servlet definition elsewhere in the deployment descriptor.	String	MyServlet
Servlet Class	The fully-qualified class name of the servlet. You may use only one of either the `<servlet-class>`tags or `<jsp-file>` tags in your servlet body.	Class name	None

Attribute	Description	Range of Values	Default Value
Jsp File	The full path to a JSP file within the Web application, relative to the Web application root directory. You may use only one of either the `<servlet-class>`tags or `<jsp-file>` tags in your servlet body.	String	None
Load On Startup	WebLogic Server initializes this servlet when WebLogic Server starts up. The optional contents of this element must be a positive integer indicating the order in which the servlet should be loaded. Lower integers are loaded before higher integers.	Positive Integer If no value is specified, or if the value specified is not a positive integer, WebLogic Server can load the servlet in any order in the startup sequence.	0

Parameters

Attribute	Description	Range of Values	Default Value
Description	Text description of the initialization parameter.		MyError Page
Param Name	Defines the name of this initialization parameter.	String	None

Attribute	Description	Range of Values	Default Value
Param Value	Defines a `String` value for this initialization parameter.	String	None

Security Role Refs

Attribute	Description	Range of Values	Default Value
Description	Text description of the role.	String	MyError Page
Role Name	Defines the name of the security role or principal that is used in the servlet code.	String	None
Role Link	Defines the name of the security role.	String - a security role that is defined in a `<security-role>` element later in the deployment descriptor.	None

Servlet Mappings

Attribute	Description	Range of Values	Default Value
Servlet	The name of the servlet to which you are mapping a URL pattern. This name corresponds to the name you assigned a servlet in a `<servlet>` declaration tag.	A declared servlet.	none

URL Pattern	Describes a pattern used to resolve URLs. The portion of the URL after the `http://host:port` + `WebAppName` is compared to the `<url-pattern>` by WebLogic Server. If the patterns match, the servlet mapped in this element will be called.
	Example patterns:
	`/soda/grape/*`
	`/foo/*`
	`/contents`
	`*.foo`
	The URL must follow the rules specified in Section 10 of the Servlet 2.2 Specification.
		String	MyServlet Mapping

Mime Mappings

Attribute	Description	Range of Values	Default Value
Mime Type	A string describing the defined mime type, for example: `text/plain.`	String	MyMime Mapping
Extension	A string describing an extension, for example: `txt.`	String	None

Session Config

Attribute	Description	Range of Values	Default Value
Session Timeout	The number of minutes after which sessions in this Web Application expire. The value set in this element overrides the value set in the `TimeoutSecs` parameter of the Session Descriptor in the WebLogic-specific deployment descriptor `weblogic.xml`, unless one of the special values listed here is entered.	Maximum value: Integer.MAX_VALUE ÷ 60 Special values: ■ -2 = Use the value set by `TimeoutSecs` in Session Descriptor ■ -1 = Sessions do not timeout. The value set in Session Descriptor is ignored.	-2

Welcome Files

Attribute	Description	Range of Values	Default Value
Welcome File	File name to use as a default welcome file, such as index.html. You may specify one or more welcome files.	String	None

Error Pages

Attribute	Description	Range of Values	Default Value
Error Code	A valid HTTP error code, for example `404.`	String	MyError Page
Exception Type	A fully-qualified class name of a Java exception type, for example `java.lang.`	String	None
Location	The location of the resource to display in response to the error. Must include a leading `/`. For example `/myErrorPg.html.`	String	None

Tag Libs

Attribute	Description	Range of Values	Default Value
URI	Describes a URI, relative to the location of the web.xml document, identifying a Tag Library used in the Web application. If the URI matches the URI string used in the taglib directive on the JSP page, this taglib is used.	String	MyTag Lib
Location	Gives the file name of the tag library descriptor relative to the root of the Web application. It is a good idea to store the tag library descriptor file under the `WEB-INF` directory so it is not publicly available over an HTTP request.	String	None

Resource Env Refs

The resource-env-ref element contains a declaration of an component's reference to an administered object associated with a resource in the component's environment.

Attribute	Description	Range of Values	Default Value
Description	A text description.	String	none
Ref Name	The name of a resource environment reference; its value is the environment entry name used in code.	String	MyResource Env Ref
Ref Type	The type of a resource environment reference.	String	none

Resource Refs

Attribute	Description	Range of Values	Default Value
Description	A text description.	String	none
Ref Name	The name of the resource used in the JNDI tree. Servlets in the Web application use this name to look up a reference to the resource.	String	MyResource Ref
Ref Type	The Java type of the resource that corresponds to the reference name. Use the full package name of the Java type.	Java type	none
Auth	Used to control the resource sign on for security.	`APPLICATION:` The application component code performs resource sign on programmatically. `CONTAINER:` WebLogic Server uses the security context established with login config element.	none
Sharing Scope	Specifies whether connections obtained through the given resource manager connection factory reference can be shared.	`Sharable` `Unsharable`	`Sharable`

Security Constraints

Attribute	Description	Range of Values	Default Value
Display Name	Name of this constraint.	String	MySecurity Constraint

Auth Constraint

Defines which groups or principals have access to the collection of web resources defined in this security constraint.

Attribute	Description	Range of Values	Default Value
Description	A text description of this security constraint.	String	none
Role Name	Defines which security roles can access resources defined in this security-constraint. Security role names are mapped to principals using the `Role Name.`	A defined security role.	none

User Data Constraint

Defines how the client should communicate with the server.

Attribute	Description	Range of Values	Default Value
Description	A text description.	String	none
Transport Guarantee	Specifies that the communication between client and server. WebLogic Server establishes a Secure Sockets Layer (SSL) connection when the user is authenticated using the `INTEGRAL` or `CONFIDENTIAL` constraint.s	■ `NONE`—the application does not require any transport guarantees. ■ `INTEGRAL`—the application requires that the data be sent between the client and server in such a way that it cannot be changed in transit. ■ `CONFIDENTIAL`—the application requires that the data be transmitted in a fashion that prevents other entities from observing the contents of the transmission.	none

Web Resource Collection

Defines the components of the Web application to which this security constraint is applied.

Attribute	Description	Range of Values	Default Value
Web Resource Name	The name of this Web resource collection.	String	none
Description	A text description of this security constraint.	String	none

URL Pattern	Use one or more of the URL Pattern elements to declare to which URL patterns this security constraint applies. If you do not use at least one of these elements, this web resource collection is ignored by WebLogic Server.	String	none
HTTP Method	Use one or more of the HTTP Method elements to declare which HTTP methods (`GET | POST |...`) are subject to the authorization constraint. If you omit the HTTP Method element, the default behavior is to apply the security constraint to all HTTP methods.	`GET` `POST`	none

LoginConfig

Attribute	**Description**	**Range of Values**	**Default Value**
Auth Method	Specifies the method used to authenticate the user. Possible values: `BASIC` - uses browser authentication `FORM` - uses a user-written HTML form `CLIENT-CERT`	`BASIC` - uses browser authentication `FORM` - uses a user-written HTML form `CLIENT-CERT`	BASIC

Realm Name	The name of the realm that is referenced to authenticate the user credentials. If omitted, the WebLogic realm is used by default. For more information, see Specifying a Security Realm at http://e-docs.bea.com/wls/docs61/adminguide/cnfgsec.html#cnfgsec004.	String	None
Login Page	The URI of a Web resource relative to the document root, used to authenticate the user. This can be an HTML page, JSP, or HTTP servlet, and must return an HTML page containing a FORM that conforms to a specific naming convention.	String	None
Error Page	The URI of a Web resource relative to the document root, sent to the user in response to a failed authentication login.	String	None

Security Roles

Attribute	Description	Range of Values	Default Value
Description	A text description of this security role.	String	none
Role Name	The role name. The name you use here must have a corresponding entry in the WebLogic-specific deployment descriptor, `weblogic.xml`, which maps roles to principals in the security realm.	String	system

Env Entries

Attribute	Description	Range of Values	Default Value
Description	A textual description.	String	none
Env Entry Name	The name of the environment entry.	String	MyEnvironment Entry
Env Entry Value	The value of the environment entry.	String	none
Env Entry Type	The name of the environment entry.	String	none

Ejb refs

Attribute	Description	Range of Values	Default Value
Description	A text description of the reference.	String	none
EJBRef Name	The name of the EJB used in the Web application. This name is mapped to the JNDI tree in the WebLogic-specific deployment descriptor `weblogic.xml`.	String	MyEJB Ref
EJBRef Type	The expected Java class type of the referenced EJB.	String	none
Home Interface Name	The fully qualified class name of the EJB home interface.	String	none
Remote Interface Name	The fully qualified class name of the EJB remote interface.	String	none
EJBLink Name	The `<ejb-name>` of an EJB in an encompassing J2EE application package.	String	none
Run As	A security role whose security context is applied to the referenced EJB.	A security role defined in this web application.	none

Security Role Assignment

Attribute	Description	Range of Values	Default Value
Role Name	Specifies the name of a security role.	A valid security role.	none
Principal Name	Specifies the name of a principal that is defined in the security realm. You can use multiple `<principal-name>` elements to map principals to a role. For more information on security realms, see the *Programming WebLogic Security* at http://e-docs.bea.com/wls/docs61/security/index.html.	A principal defined in the security realm.	none

Reference Descriptor

Resource Descriptions

Attribute	Description	Range of Values	Default Value
Res Ref Name	Specifies the name of a resource reference.	String	none
Jndi Name	Specifies a JNDI name for the resource.	Java character set name	none

EJB Reference Description

Attribute	Description	Range of Values	Default Value
Ejb Ref Name	Specifies the name of an EJB reference used in your Web application.	String	none
Jndi Name	Specifies a JNDI name for the reference.	Java character set name	none

Session Descriptor

:

Parameter Name	Default Value	Parameter Value
URLRewritingEnabled	true	Enables URL rewriting, which encodes the session ID into the URL and provides session tracking if cookies are disabled in the browser.
IDLength	52	Sets the size of the session ID. The minimum value is 8 bytes and the maximum value is `Integer.MAX_VALUE`. If you are writing a WAP application, you must use URL rewriting because the WAP protocol does not support cookies. Also, some WAP devices have a 128-character limit on URL length (including parameters), which limits the amount of data that can be transmitted using URL re-writing. To allow more space for parameters, use this parameter to limit the size of the session ID that is randomly generated by WebLogic Server

Parameter Name	Default Value	Parameter Value
CookieComment	Weblogic Server Session Tracking Cookie	Specifies the comment that identifies the session tracking cookie in the cookie file. If unset, this parameter defaults to `WebLogic Session Tracking Cookie`. You may provide a more specific name for your application.
CookieDomain	Null	Identifies the server to which the browser sends cookie information when the browser makes a request. For example, setting the CookieDomain to `.mydomain.com` returns cookies to any server in the `*.mydomain.com` domain. The domain name must have at least two components; setting a name to `*.com` or `*.net` is invalid. If unset, this parameter defaults to the server that issued the cookie.
CookieMaxAgeSecs	-1	Sets the life span of the session cookie, in seconds, after which it expires on the client. If the value is 0, the cookie expires immediately. The maximum value is MAX_VALUE, where the cookie lasts forever. If set to `-1`, the cookie expires when the user exits the browser.
CookieName	`JSESSIONID`	Defines the session cookie name. Defaults to `JSESSIONID` if unset. You may set this to a more specific name for your application.
CookiePath	Null	Specifies the pathname to which the browser sends cookies. If unset, this parameter defaults to `/` (slash), where the browser sends cookies to all URLs served by WebLogic Server. You may set the path to a narrower mapping, to limit the request URLs to which the browser sends cookies.

Parameter Name	Default Value	Parameter Value
InvalidationIntervalSecs	`60`	Sets the time, in seconds, that WebLogic Server waits between doing house-cleaning checks for timed-out and invalid sessions, and deleting the old sessions and freeing up memory. Use this parameter to tune WebLogic Server for best performance on high traffic sites. The minimum value is every second (1). The maximum value is once a week (604,800 seconds). If unset, the parameter defaults to 60 seconds.
JDBConnectionTimeoutSecs	120	Sets the time, in seconds, that WebLogic Server waits before timing out a JDBC connection, where x is the number of seconds between.
PersistenceStoreDir	`session_db`	If you have set `PersistentStoreType` to `file`, this parameter sets the directory path where WebLogic Server will store the sessions. The directory path is either relative to the temp directory or an absolute path. The temp directory is either a generated directory under the `WEB-INF` directory of the Web application, or a directory specified by the context-param `javax.servlet.context.tmpdir`. Ensure that you have enough disk space to store the *number of valid sessions* multiplied by the *size of each session*. You can find the size of a session by looking at the files created in the `PersistenceStoreDir`. You can make file-persistent sessions clusterable by making this directory a shared directory among different servers. You must create this directory manually.
PersistenceStorePool	None	Specifies the name of a JDBC connection pool to be used for persistence storage.

Parameter Name	Default Value	Parameter Value
PersistentStoreType	`memory`	Sets the persistent store method to one of the following options: ■ `memory`—disables persistent session storage ■ `file`—uses file-based persistence (See also `PersistenceStoreDir`, above) ■ `jdbc`—uses a database to store persistent sessions. (see also `PersistenceStorePool`, above) ■ `replicated`—same as `memory`, but session data is replicated across the clustered servers ■ cookie—all session data is stored in a cookie in the user's browser
CookiesEnabled	True	Use of session cookies is enabled by default and is recommended, but you can disable them by setting this property to `false`. You might turn this option off to test using URL rewriting.
TrackingEnabled	True	When set to true, session tracking is enabled.
TimeoutSecs	3600	Sets the time, in seconds, that WebLogic Server waits before timing out a session, where x is the number of seconds between a session's activity. Minimum value is 1, default is 3600, and maximum value is `integer MAX_VALUE`. On busy sites, you can tune your application by adjusting the timeout of sessions. While you want to give a browser client every opportunity to finish a session, you do not want to tie up the server needlessly if the user has left the site or otherwise abandoned the session. This parameter can be overridden by the `session-timeout` element (defined in minutes) in `web.xml`. For more information, see `Session Timeout`.
ConsoleMainAttribute		If you enable Session Monitoring in the WebLogic Server Administration Console, set this parameter to the name of the session parameter you will use to identify each session that is monitored.

Parameter Name	Default Value	Parameter Value
PersistentStoreCookieName	`WLCOOKIE`	Sets the name of the cookie used for cookie-based persistence.

JSP Descriptor

Parameter Name	Default Value	Parameter Value
Compile Command	`javac`, or the Java compiler defined for a server under the configuration /tuning tab of the WebLogic Server Administration Console	Specifies the full pathname of the standard Java compiler used to compile the generated JSP servlets. For faster performance, specify a different compiler, such as IBM's Jikes or Symantec's `sj`.
Compiler Class	`None`	Name of a Java compiler that is executed in WebLogic Servers's virtual machine. (Used in place of an executable compiler such as `javac` or `sj`.)

Compile Flags	`None`	Passes one or more command-line flags to the compiler. Enclose multiple flags in quotes, separated by a space. For example:
		`java weblogic.jspc`
		` -compileFlags "-g -v" myFile.jsp`

Working Dir	`internally generated directory`	The name of a directory where WebLogic Server saves the generated Java and compiled class files for a JSP.

Parameter Name	Default Value	Parameter Value
Verbose	`true`	When set to `true`, debugging information is printed out to the browser, the command prompt, and WebLogic Server log file.
keepgenerated	`false`	Saves the Java files that are generated as an intermediary step in the JSP compilation process. Unless this parameter is set to `true`, the intermediate Java files are deleted after they are compiled.
Page Check Seconds	`1`	Sets the interval, in seconds, at which WebLogic Server checks to see if JSP files have changed and need recompiling. Dependencies are also checked and recursively reloaded if changed. If set to 0, pages are checked on every request. If set to `-1`, page checking and recompiling is disabled.
Encoding	Default encoding of your platform	Specifies the default character set used in the JSP page. Use standard Java character set names (see http://java.sun.com/j2se/1.3/docs/guide/intl/encoding.doc.htm). If unset, this parameter defaults to the encoding for your platform. A JSP page directive (included in the JSP code) overrides this setting. For example: `<%@ page contentType="text/html; charset=custom-encoding”%>`
Package Prefix	`jsp_servlet`	Specifies the package into which all JSP pages are compiled.
No Try Blocks	`false`	If a JSP file has numerous or deeply nested custom JSP tags and you receive a `java.lang.VerifyError` exception when compiling, use this flag to allow the JSPs to compile correctly.
Precompile	`false`	When set to true WebLogic Server automatically compiles all JSPs on startup.
Compiler Supports Encoding	`False`	Specifies the encoding used by the WebLogic JSP compiler to create the intermediate`.java` file. When set to true, the JSP compiler uses the encoding specified with the `contentType` attribute contained in the `page` directive on the JSP page, or, if a `contentType` is not specified, the encoding defined with the `encoding` parameter in the jsp-descriptor. When set to false, the JSP compiler uses the default encoding for the JVM when creating the intermediate `.java` file.

Container Descriptor

Attribute	Description	Range of Values	Default Value
Check Auth on Forward Enabled	When enabled, requires authentication of forwarded requests from a servlet or JSP.	Checked, Unchecked	Unchecked
redirect-with-absoute-url	Controls whether the `javax.servlet.http.HttpServletResponse.SendRedirect()` method redirects using a relative or absolute URL. Set this element to `false` if you are using a proxy HTTP server and do not want the URL converted to a non-relative link. The default behavior is to convert the URL to a non-relative link.	boolean	true

Charset Params

Input Charset Descriptors

Attribute	Description	Range of Values	Default Value
Resource Path	A path which, if included in the URL of a request, signals WebLogic Server to use the Java character set specified by the Java Charset Name field.	String	none
Java Charset Name	Specifies the Java characters set to use.	Java character set name	none

Character Set Mapping

Attribute	Description	Range of Values	Default Value
IANA Charset Name	Specifies the IANA character set name that is to be mapped to the Java character set specified by the Java Charset Name field.	IANA character set name	none
Java Charset Name	Specifies the Java characters set to use.	Java character set name	none

Configuring a WebLogic Web Service

The Web Service information in the Deployment Descriptor Editor is provided for monitoring purposes only; changing any of the values might result in the deployed Web Service not functioning correctly.

Web Services

RPC Services

Attribute	Description	Range of Values	Default Value
JNDI Name	JNDI name of the stateless session EJB that comprises the RPC-style Web service.	String.	None.
Home Interface	The Home interface of the stateless session EJB.	String.	None.
Remote Interface	The Remote interface of the stateless session EJB.	String.	None.
URI	URI used by client applications to invoke the Web service.	String.	None.

Message Services

Attribute	Description	Range of Values	Default Value
Service Name	Name of the SOAP Servlet that handles SOAP messages between WebLogic Server and client applications.	String.	Null.
Destination	JNDI name of the JMS topic or queue which receives or sends data between WebLogic Server and the client application.	String.	None.
Destination Type	The type of JMS destination: topic or queue.	`Topic` or `Queue`.	None.
Action	Specifies whether the client application that invokes this message-style Web service sends or receives data to the JMS destination. Specify `send` if the client sends data to the JMS destination and `receive` if the client receives data from the JMS destination.	`Send` or `Receive`.	None.
Connection Factory	JNDI name of the ConnectionFactory used to create a connection to the JMS destination.	String.	None.

Attribute	Description	Range of Values	Default Value
URI	URI used by clients to invoke the Web service.	String.	None.

83 Shutdown Class

These procedures describe how to use the Administration Console to set the attributes for configuring and managing shutdown classes. For more information, see Starting and Stopping WebLogic Servers in the Administration Guide.

Configure a Shutdown Class

1. Click the Startup & Shutdown node. The Startup & Shutdown table displays in the right pane showing all the shutdown classes defined in your domain.
2. Click the Configure a New Shutdown Class text link. A dialog displays in the right pane showing the tabs associated with configuring a new shutdown class.
3. Enter values in the Name, Class Name, Deployment Order, and Arguments attribute fields.

   In the Arguments field, separate multiple arguments with a comma. For example:
   `first=MyFirstName,last=MyLastName`
4. Click Create to create a shutdown-class instance with the name you specified in the Name field. The new instance is added under the Startup & Shutdown node in the left pane.

Clone a Shutdown Class

1. Click the Startup & Shutdown node. The Startup & Shutdown table displays in the right pane showing all the shutdown classes defined in your domain.
2. Click the Clone icon in the row of the shutdown class you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a shutdown class.
3. Enter values in the Name, Class Name, and Arguments attribute fields.
4. Click Create to create a shutdown-class instance with the name you specified in the Name field. The new instance is added under the Startup & Shutdown node in the left pane.

Delete a Shutdown Class

1. Click the Startup & Shutdown node. The Startup & Shutdown table displays in the right pane showing all the shutdown classes defined in your domain.
2. Click the Delete icon in the row of the shutdown class you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click the Yes button to delete the shutdown class. The shutdown-class icon under the Startup & Shutdown node is deleted.

Assign a Shutdown Class

1. Click the instance node in the left pane under Startup & Shutdown for the shutdown class you want to assign. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Targets tab.

3. Complete the following steps for the Servers and Clusters tabs:

    a. Select one or more targets in the Available column that you want to assign to the shutdown class.

    b. Click the mover control to move the targets you selected to the Chosen column.

    c. Click Apply to save your assignments.

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	This attribute returns the name of the shut down class.	Fully qualified class name	Null
Class Name	This attribute returns the class name.		Null
Arguments	This attribute sets the arguments for the shutdown class. Separate multiple arguments with a comma. For example: `first=MyFirstName, last=MyLastName`		Null
Deployment Order	Order in which the class is to be implemented.		

Targets

Servers

Attribute	Description	Range of Values	Default Value
Target Servers	This attribute sets the server targets for this deployment		[Lweblogic.management.configuration.TargetMBean;@6b7920

Clusters

Attribute	Description	Range of Values	Default Value
Target Clusters	This attribute sets the cluster targets for this deployment		[Lweblogic.management.configuration.TargetMBean;@6b7920

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for optional user-supplied information.	The value must be an alphanumeric string.	Null

For more information, see Starting and Stopping WebLogic Servers in the Administration Guide

84 SNMP Trap Runtime

This allows you to sort by the following attributes:

- Name
- Host
- Port
- Community

85 SNMP Runtime

This allows you to sort by the following attributes:

- Enabled
- SNMP Port
- MIB Data Refresh Interval
- Server Status Check Interval Factor
- Community Prefix
- Debug Level

86 SNMP Attribute Change

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	This attribute allows you to set the name for the attribute change filter	String	Null
Attribute MBean Type	This attribute sets the type of the configuration MBean containing the attribute you wish to monitor for changes.	MBean type	Null
Attribute MBean Name	This attribute sets the name of the configuration MBean containing the attribute you wish to monitor for changes.	MBean name	Null

Attribute	Description	Range of Values	Default Value
Attribute Name	This attribute sets the name of the attribute you wish to monitor for changes.		

87 SNMP Attribute Change Runtime

This allows you to sort by the following attributes:

- Name
- Attribute MBean Type
- Attribute MBean Name
- Attribute Name

88 SNMP Counter Monitor

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	This attribute allows the user to set the name for this monitor.	String	Null

Attribute	Description	Range of Values	Default Value
Offset	Specifies an integer value to be added to the threshold if the threshold is crossed. For example, if you set Threshold to 1000 and Offset to 2000, when the observed attribute exceeds 1000, the Counter Monitor sends a notification and increases the threshold to 3000. When the observed attribute exceeds 3000, the Counter Monitor sends a notification and increases the threshold again to 5000. The threshold can increase up to a maximum specified by the Modulus attribute.	Integer	0
Threshold	Specifies a value that triggers the Counter Monitor to generate a trap. The Counter Monitor generates a trap each time it polls the value and determines that the value has transitioned from below the threshold to at or above the threshold. While the value remains at or above the threshold, the Counter Monitor does not generate additional traps.	Integer	0

Attribute	Description	Range of Values	Default Value
Modulus	Specifies the maximum value for the threshold. You can specify an offset that causes the threshold to increase. When the threshold reaches the value specified by the Modulus, the threshold is returned to its original value before any offsets were applied. For example, if the original threshold is set to 1000 and the modulus is set to 5000, when the threshold exceeds 5000, the monitor sends a notification and resets the threshold to 1000. If you specify 0, the Counter Monitor does not use the modulus and the threshold value can grow indefinitely.	Integer	
Monitored MBean Type	This attribute sets the type for the MBean that includes the attribute that you want to monitor.	String	Null
Monitored MBean Name	This attribute sets the name for the MBean that includes the attribute that you want to monitor.	String	Null
Monitored Attribute Name	This attribute sets the name of the attribute that you want to monitor.	String	Null;

Attribute	Description	Range of Values	Default Value
Polling Interval	This attribute sets the frequency that the agent is to check the attribute value, in seconds.	Integer, in seconds	0

89 SNMP Counter Monitor Runtime

This allows you to sort by the following attributes:

- Name
- Offset
- Threshold
- Modulus
- Monitored MBean Type
- Monitored MBean Name
- Monitored Attribute Name
- Polling Interval
- Enabled Servers

90 SNMP Gauge Monitor

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	This attribute allows the user to set the name for this monitor.	String	Null
Monitored MBean Type	This attribute sets the type for the MBean that includes the attribute that you want to monitor.	String	Null
Monitored MBean Name	This attribute sets the name for the MBean that includes the attribute that you want to monitor.	String	Null
Monitored Attribute Name	This attribute sets th ename ofthe attribute that you want to monitor.	String	Null;

Attribute	Description	Range of Values	Default Value
Polling Interval	This attribute sets the frequency that the agent is to check the attribute value, in seconds.	Integer, in seconds	0
Threshold High	This attribute sets the threshold at which a trap will be generated The trap will be generated if the attribute value is equal to or greater than the value entered here..	Integer, in seconds	0
Threshold Low	This attribute sets the threshold at which a trap will be generated The trap will be generated if the attribute value is equal to or less than the value entered here.	Integer, in seconds	0

Servers

Attribute	Description	Range of Values	Default Value
Enabled Servers	This attribute allows the user to select from the list of available servers those on which you wnat this monitor to check the specified attribute.	List	Null

91 SNMP Gauge Monitor Runtime

This allows you to sort by the following Attributes:

- Monitored MBean Type
- Monitored MBean Name
- Monitored Attribute Name
- Polling Interval
- Threshold High
- Threshold Low
- Enabled Servers

92 SNMP JMX Monitor

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	This attribute allows the user to set the name for this monitor.	String	Null
Monitored MBean Type	This attribute sets the type for the MBean that includes the attribute that you want to monitor.	String	Null
Monitored MBean Name	This attribute sets the name for the MBean that includes the attribute that you want to monitor.	String	Null
Monitored Attribute Name	This attribute sets th ename ofthe attribute that you want to monitor.	String	Null;

Attribute	Description	Range of Values	Default Value
Polling Interval	This attribute sets the frequency that the agent is to check the attribute value, in seconds.	Integer, in seconds	0
Threshold High	This attribute sets the threshold at which a trap will be generated The trap will be generated if the attribute value is equal to or greater than the value entered here..	Integer, in seconds	0
Threshold Low	This attribute sets the threshold at which a trap will be generated The trap will be generated if the attribute value is equal to or less than the value entered here.	Integer, in seconds	0

Servers

Attribute	Description	Range of Values	Default Value
Enabled Servers	This attribute allows the user to select from the list of available servers those on which you wnat this monitor to check the specified attribute.	List	Null

93 SNMP Log Filters

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	This attribute allows the user to set the name of the log filter	String	MyLog Filter
Severity Level	This attribute allows the user to set the severity level at which a trap will be generated.		Error
Subsystem Names	This attribute allows the user to set the subsystem name to be monitored.	String	Null
User IDs	This attribute allows the user to set the User ID to be notified when a trap is generated.	String	Null

Attribute	Description	Range of Values	Default Value
Message IDs	This attribute allows the user to seta message ID to be searched for in the message.		Null
Message Substring	This attribute sets the srting to be searched for in the message text.	String	Null

94 SNMP Log Filters Runtime

This allows you to sort by the following attributes:

- Severity Level
- Subsystem Names
- User Ids
- Message Ids
- Message Substring

95 SNMP Proxies

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	This attribute allows the user to set the name of the proxy	String	MyProxy
Port	This attribute allows the user to set the Port number for communication with the other SNMP Agent.	Valid Port	0
Oid Root	This attribute allows the user to set the absolute OID that designates the root, ot top node, of the part of the OID tree being assigned to the agent.	Valid OID Root	Null

Attribute	Description	Range of Values	Default Value
Community	This attribute allows the user to set the community name that the other agent expects in requests from SNMP managers.		na
Timeout	This attribute allows the user to set the interval that the WebLogic SNMP proxy agent waits for a response to requests forwarded to another SNMP agent. If this interval elapses without a response, the WebLogic SNMP agent will send an error to the requesting manager.	Integer	5000

96 SNMP Proxies Runtime

This allows you to sort by the following attributes:

- Name
- Port
- Oid Root
- Community
- Timeout

97 SNMP String Monitor

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	This attribute allows the user to set the name for this monitor.	String	Null
String to Compare	This attribute allows the user to set the string to be compared to the value being monitored in order to determine if a trap should be generated.	String	Null
Notify Differ	This attribute is selected if the user wants to generate a trap if the vlue of the attribute is different from the value entered in the String to Compare field.	Checkbox	Not Selected

Attribute	Description	Range of Values	Default Value
Notify Match	This attribute is selected if the user wants to generate a trap if the vlue of the attribute matches the value entered in the String to Compare field.	Checkbox	Not Selected
Monitored MBean Type	This attribute sets the type for the MBean that includes the attribute that you want to monitor.	String	Null
Monitored MBean Name	This attribute sets the name for the MBean that includes the attribute that you want to monitor.	String	Null
Monitored Attribute Name	This attribute sets th ename ofthe attribute that you want to monitor.	String	Null;
Polling Interval	This attribute sets the frequency that the agent is to check the attribute value, in seconds.	Integer, in seconds	0

Servers

Attribute	Description	Range of Values	Default Value
Enabled Servers	This attribute allows the user to select from the list of available servers those on which you wnat this monitor to check the specified attribute.	List	Null

98 SNMP String Monitor Runtime

This allows you to sort by the following attributes:

- Name
- String to Compare
- Notify Differ
- Notify Match
- Monitored MBean Type
- Monitored MBean Name
- Monitored Attribute Name
- Polling Interval
- Enabled Servers

99 SNMP Trap Destinations

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	This is a user-provided name for this destination	String	Null
Host	This is a string that contains either a hostname or IP address. This is the machine of the SNMP manager that is a target for SNMP trap notifications sent by the WebLogic SNMP agent.	String	Null
Port	This is the port that will be used for sending SNMP trap notifications to the target SNMP manager.		162

Attribute	Description	Range of Values	Default Value
Community	This is the SNMP trap community name. This functions as a password for sending trap notifications to the target SNMP manager.		

100 SNMP

Monitor SNMP Attributes

Configuration

General

Attribute	Description	Range of Values	Default Value

Enabled	The SNMP Service provides the SNMP agent functionality and is a part of the Administration Server. The agent monitors all WebLogic resources for the domain. The Admnistration Server must be restarted for configuration changes to take effect.	True = enabled
		False = disabled
			False

Attribute	Description	Range of Values	Default Value
SNMP Port	This is the port on which the WebLogic SNMP agent listens for incoming requests from SNMP managers.	Valid listen port	161
MIB Data Refresh Interval	The SNMP agent maintains a cache of all the attribute values and responds to manager requests by obtaining the attribute value from this cache. The MIB Data Refresh Interval is the interval, in seconds, at which the SNMP agent does a complete refresh of the cache. When it does a refresh, it does a GET on all the WebLogic attributes represented in the WebLogic SNMP MIB.	Integer, in seconds	120

Attribute	Description	Range of Values	Default Value
Server Status Check Interval Factor	The SNMP agent multiplies this number times the MIB Data Refresh Interval to determine how frequently it should check to determine whether Managed Servers in the domain are up or down. The agent obtains this value from the MIB cache. If the Server Status Check Interval Factor is 1, the WebLogic SNMP agent checks whether Managed Servers are up or down at the interval defined in MIB Data Refresh Interval.	Integer, >0	Obtained from MIB Cache
Community Prefix	This string is used to form the SNMP Community Name which functions as a textual password for communication with SNMP managers. If the community prefix sent by the SNMP manager does not match the value configured in this attribute, the SNMP agent will return an authenticationFailuretrap to the requestor.	String If it is of the form `community_prefix@server_name`, the agent will return data only for the specified Managed Server. If it is of the form `community_prefix@domain_name`, the agent will return data for every server in the domain. If the SNMP manager sends only `community_prefix`, the agent will only retrieve data for the Administration Server.	Null

Attribute	Description	Range of Values	Default Value
Debug Level	This attribute sets the level of debugging messages sent to the administrator.	Integer between 0 and 3 The higher the value, the more detailed the messages: ■ `0` generates no debug messages ■ `1` generates only fatal debug messages ■ `2` generates fatal and critical debug messages ■ `3` generates fatal, critical, and non-critical debug messages	0

101 Socket

By default, this pane allows the user to sort the objects by the following criteria:

- Protocol
- Remote address

102Startup Class

These procedures describe how to use the Administration Console to set the attributes for configuring and managing startup classes. For more information, see Starting and Stopping WebLogic Servers in the Administration Guide.

Configure a Startup Class

1. Click the Startup & Shutdown node. The Startup & Shutdown table displays in the right pane showing all the startup classes defined in your domain.
2. Click the Configure a New Startup Class text link. A dialog displays in the right pane showing the tabs associated with configuring a new startup class.
3. Enter values in the Name, Class Name, Deployment Order, and Arguments attribute fields.

   In the Arguments field, separate multiple arguments with a comma. For example: `first=MyFirstName,last=MyLastName`
4. Select the checkbox to enable Abort Startup on Failure; deselect the checkbox to disable this feature.
5. Click Create to create a startup-class instance with the name you specified in the Name field. The new instance is added under the Startup & Shutdown node in the left pane.

Clone a Startup Class

1. Click the Startup & Shutdown node. The Startup & Shutdown table displays in the right pane showing all the startup classes defined in your domain.
2. Click the Clone icon in the row of the startup class you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a startup class.
3. Enter values in the Name, Class Name, Deployment Order, and Arguments attribute fields.
4. Select the checkbox to enable Abort Startup on Failure; deselect the checkbox to disable this feature.
5. Click Create to create a startup-class instance with the name you specified in the Name field. The new instance is added under the Startup & Shutdown node in the left pane.

Delete a Startup Class

1. Click the Startup & Shutdown node. The Startup & Shutdown table displays in the right pane showing all the startup classes defined in your domain.
2. Click the Delete icon in the row of the startup class you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click Yes to delete the startup class. The startup-class icon under the Startup & Shutdown node is deleted.

Assign a Startup Class

1. Click the instance node in the left pane under Startup & Shutdown for the startup class you want to assign. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Targets tab.
3. Complete the following steps for the Servers, Groups, and Clusters tabs:

    a. Select one or more targets in the Available column that you want to assign to the startup class.

    b. Click the mover control to move the targets you selected to the Chosen column.

    c. Click Apply to save your assignments.

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	This attribute returns the name of the startup class.	Fully qualified class name	Null
Class Name			
Arguments	This attribute sets the arguments for this startup class. Separate multiple arguments with a comma. For example: `first=MyFirstName,last=MyLastName`		

Abort Startup on Failure	This attribute determines the ability of the server to abort the startup in the event of a failure.	Boolean
		True = selected
		False = not selected
			Not Selected

Deployment Order	Order in which the class is to be implemented.		

Target

Servers

Attribute	Description	Range of Values	Default Value
Target Servers	This attribute sets the server targets for this deployment.		[Lweblogic.management.configuration.TargetMBean;@5e3974

Clusters

Attribute	Description	Range of Values	Default Value
Target Clusters	This attribute sets the cluster targets for this deployment.		[Lweblogic.management.configuration.TargetMBean;@5e3974

Notes

Attribut e	Description	Range of Values	Default Value
Notes	This attribute provides a space for user supplied information.	The value must be an alphanumeric string.	Null

For more information, see Starting and Stopping WebLogic Servers in the Administration Guide

103 Stateless EJB Runtime

You can monitor stateless session EJBs using the following statistics:

Statistic	Description
Idle Beans Count	Reports the number of idle beans in the free pool that are available for use.
Beans In Use Count	Reports the number of beans currently in use from the free pool. This information is useful for tracking demand for your EJB. For example, this can be important when investigating an abnormal pool miss ratio. The `beans-in-use` number may slightly exceed the `beans-in-cache` number in certain situations. WebLogic Server uses anonymous instances to execute finder and home methods. These anonymous instances would cause the `beans-in-use` count to go up while not changing the `beans-in-cache` count. However, the difference between these numbers should be relatively small.
Waiter Total Count	Reports the number of times a thread requested and had to wait for a bean from the pool.
Timeout Total Count	Reports the total number of transactions that have timed out. Every EJB request uses valuable server resources such as threads and bean instances. A timed out transaction means that server resources were tied up in vain, suggesting a problem with the application.

Statistic	Description
Transactions Committed Total Count	Reports the number of transactions that committed.
Transactions Rolled Back Total Count	Reports the number of transactions rolled back. Use this to calculate the transaction rollback ratio, which is the ratio of transactions that have rolled back to the number of total transactions involving the EJB: `Transaction Rollback Ratio = (Transaction Total Rollback Count / Transaction Total Count) * 100` This information is useful for several reasons. First, it may be useful for signaling a problem with an application. For example, an unexpectedly high rollback ratio may be caused by a problem with a resource used by the application. You can also use it to gauge the efficiency of an application. A high transaction rollback ratio may mean that a lot of work is being done only to eventually be rolled back, which is inefficient.
Transactions Timed Out Total Count	Reports the number of transactions that timed out. Use this to calculate the transaction timeout ratio, which is the ratio of transactions that have timed out to the total number of transactions involving an EJB: `Transaction Timeout Ratio = (Transaction Total Timeout Count / Transaction Total Count) * 100` Timeouts can signal application inefficiency. Every EJB request uses valuable server resources such as threads and bean instances. A timed out transaction means that server resources were tied up in vain. The transaction timeout ratio is a good indicator of a problem with an application.

104 Stateful EJB Runtime

You can monitor stateful session EJBs using the following statistics:

Statistic	Description
Cached Beans Current Count	Reports the total number of beans from this EJB Home currently in the EJB cache. Use this information to calculate the current percentage of the configured cache capacity being used.
Cache Access Count	Reports the total number of attempts to access a bean from the cache. This information is useful for giving context to other counts such as cache hits.
Cache Hit Count	Reports the total number of times an attempt to access a bean from the cache succeeded. This information is useful for determining the effectiveness of the EJB cache.
Activation Count	Reports the total number of beans from this EJB Home that have been activated.
Passivation Count	Reports the total number of beans from this EJB Home that have been passivated.
Lock Entries Current Count	Reports the current number of lock entries in the lock manager. This information may be helpful in detecting stale lock entries.
Lock Manager Access Count	Reports the total number of attempts to obtain a lock on a bean. This includes attempts to obtain a lock on a bean that is already locked on behalf of the client. This information is useful for giving context to the waiter and timeout total counts.
Waiter Total Count	Reports the number of times a thread requested and had to wait for a bean from the free pool.

Statistic	Description
Timeout Total Count	Reports the total number of transactions that have timed out. Every EJB request uses valuable server resources such as threads and bean instances. A timed out transaction means that server resources were tied up in vain, suggesting a problem with the application.

105 Target Groups

Create a Target Group

1. Click the Target Groups node. The Target Groups table displays in the right pane showing all the target groups defined in your domain.
2. Click the Create a New Target Group text link. A dialog displays in the right-hand pane showing the tabs associated with configuring a new target group.
3. Enter a value in the Name attribute field.
4. Click the Create button in the lower right-hand corner to create a target-group instance with the name you specified in the Name field. The new instance is added under the Target Groups node in the left pane.

Clone a Target Group

1. Click the Target Groups node. The Target Groups table displays in the right pane showing all the target groups defined in your domain.
2. Click the Clone icon in the row of the target group you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a target group.
3. Enter a value in the Name attribute field.
4. Click the Create button in the lower right-hand corner to create a target-group instance with the name you specified in the Name field. The new instance is added under the Target Groups node in the left pane.

Delete a Target Group

1. Click the Target Groups node. The Target Groups table displays in the right pane showing all the target groups defined in your domain.
2. Click the Delete icon in the row of the target group you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click the Yes button to delete the target group. The target-group icon under the Target Groups node is deleted.

Assign a Target Group

1. Click the instance node in the left pane under Target Groups for the target group you want to assign. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Members tab.
3. Complete the following steps for the Servers, Groups, and Clusters tabs:
   a. Select one or more targets in the Available column that you want to assign to the target group.
   b. Click the mover control to move the targets you selected to the Chosen column.
   c. Click Apply to save your assignments.

Configuration

Attribute	Description	Range of Values	Default Value
Name	This attribute returns the name of the target groups.	The name can be up to 256 alphanumeric characters, but may not contain commas or spaces.	Null

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for optional user-supplied information.	The value must be an alphanumeric string.	Null

106 Transaction Name Runtime

By default, this pane allows the user to display transactions sorted by the following criteria:

- Name
- Transactions
- Commits
- Rollbacks
- Timeout Rollbacks
- Resource Rollbacks
- Application Rollbacks
- System Rollbacks
- Heuristics
- Average Commit Time

Click the Customize this view text link to change the columns to display or the sort order.

107 Transaction Resource Runtime

By default, this pane allows the user to sort the objects by the following criteria:

- Name
- Transactions
- Commits
- Rollbacks
- Heuristics
- Heuristic Commits
- Heuristic Rollbacks
- Mixed Heuristics
- Heuristic Hazards

Click the Customize this view text link to change the columns to display or the sort order.

108 UnixMachine

These procedures describe how to use the Administration Console to set the attributes for configuring and managing UNIX machines. For more information, see Starting and Stopping WebLogic Servers and WebLogic Servers and Clusters in the *Administration Guide*

Configure a Unix Machine

1. Click the Machines node. The Machines table displays in the right pane showing all the Unix machines defined in the domain.
2. Click the Configure a New Unix Machine text link. A dialog displays in the right pane showing the tabs associated with configuring a new Unix machine.
3. Enter a value in the Name attribute field.
4. Click Create.
5. Click the Node Manager tab and accept or change the default values.
6. Click Apply to save the changes.

Clone a Unix Machine

1. Click the Machines node. The Machines table displays in the right pane showing all the Unix machines defined in the domain.

2. Click the Clone icon in the row of the Unix machine you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a Unix machine.
3. Enter values in the Name attribute field.
4. Click Clone to create a Unix machine instance with the name you specified in the Name field. The new instance is added under the Machines node in the left pane.
5. Click Apply to apply the changes.
6. Click the Node Manager tab and accept or change the default values.
7. Click Apply to save the changes.

Delete a Unix Machine

1. Click the Machines node. The Machines table displays in the right pane showing all the Unix machines defined in the domain.
2. Click the Delete icon in the row of the Unix machine you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click Yes to delete the Unix machine. The Unix machine icon under the Machines node is deleted.

Assign a Unix Machine

1. Click the instance node in the left pane under Machines for the Unix machine you want to assign. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Servers tab.
3. Select one or more targets in the Available column that you want to assign to the Unix machine.

4. Click the mover control to move the targets you selected to the Chosen column.
5. Click Apply to save your assignments.

Configuration

Attribute	Description	Range of Values	Default Value
Name	This attribute returns the name of the Unix machine.	String	MyUnixMachine

Servers

Attribute	Description	Range of Values	Default Value
Servers	This attribute allows the user to choose servers for this Unix machine.	List	Null

Node Managers

Attribute	**Description**	**Range of Values**	**Default Value**
Listen Address	This attribute assigns the listen address for this machine.		LocalHost
Listen Port	This attribute assigns the listen port for this machine.		7002

Post Bind UID Enabled	This attribute returns the UNIX UID a server running on this machine will run under after it has carried out all privileged startup actions. If this value is set, it is a valid Unix UID. If it is not set it is null.	Boolean
		True = selected
		False = not selected
			Not Selected

Post Bind UID	This attribute sets the UNIX UID a server running on this machine will run under after it has carried out all privileged startup actions.	String	Nobody

Attribute	Description	Range of Values	Default Value

Post Bind GID Enabled	Returns the UNIX GID a server running on this machine will run under after it has carried out all privileged startup actions. If this value is set, it is a valid Unix GID. If it is not set it is null.	Boolean
		True = selected
		False = not selected
			Not Selected

Post Bind GID	This attribute sets the UNIX GID a server running on this machine will run under after it has carried out all privileged startup actions.	String	Nobody
Certificate	This attribute specifies the file for the server's digital certificate.		Null
Certificate Password	This attribute sets the password for the digtial certificate.	Valid password for the certificate	Null
Trusted Certs File	This attribute specifies the trusted CAs for the server.		Null

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attributeprovides space for optional user-supplied information.	The value must be an alphanumeric string.	Null

109 UnixRealm

The following table describe the attributes for creating and managing Unix Realms. To read more about Realms, please see Managing Security in the *Administration Guide*.

Before you can use the UNIX Security realm, you need to enable the Caching Realm and enter the class name of the UNIX Security realm in the Basic Realm field.

Configure a Unix Realm

1. Click the Realms node in the left pane. The Realms table displays in the right pane showing all the Unix realms defined in the domain.
2. Click the Configure a New Unix Realm text link. A dialog displays in the right pane showing the tabs associated with configuring a new realm.
3. Enter values in the Name and Auth Program attribute fields.
4. Click Create to create a realm instance with the name you specified in the Name field. The new instance is added under the Realms node in the left pane.

Clone a Unix Realm

1. Click the Realms node in the left pane. The Realms table displays in the right pane showing all the Unix realms defined in the domain.

2. Click the Clone icon in the row of the realm you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a new realm.
3. Enter values in the Name and Auth Program attribute fields.
4. Click Create to create a realm instance with the name you specified in the Name field. The new instance is added under the Realms node in the left pane.

Delete a Unix Realm

1. Click the Realms node in the left pane. The Realms table displays in the right pane showing all the Unix realms defined in the domain.
2. Click the Delete icon in the row of the realm you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click the Yes button to delete the realm. The realm icon under the Realms node is deleted.

Before you can use the UNIX Security realm, you need to enable the Caching Realm and enter the class name of the UNIX Security realm in the Basic Realm field.

Configuration

Attribute	Description	Range of Values	Default Value
Name	This attribute specifies the name of the UNIX Security realm, such as, AccountingRealm	The name can be up to 256 alphanumeric characters, but may not contain commas or spaces.	Null
Realm Classname	This attribute returns the name of the Java class that contains the UNIX Security realm. The Java class needs to be in the CLASSPATH of WebLogic Server.	This attribute can not be changed.	weblogic.security.acl.unixrealm.UnixRealm
Auth Program	This attribute specifies the name of the program used to authenticate users in the UNIX security realm.	String	In most cases, the name of the program is `wlauth.`

Notes

Attribut e	Description	Range of Values	Default Value
Notes	This attribute provides a space for optional user-supplied information.	The value must be an alphanumeric string.	Null

110 User

Add a User

1. Click the Users node in the left pane. A dialog displays in the right pane showing the controls for working with users.
2. Enter the name of the new user in the Name text-entry field.
3. Enter the password of the new user in the Password text-entry field.
4. Click Create to add the user you specified. The user is added to the Users list.
5. Click the The Changes You Have Made Must Be Saved to the Realm Implementation text link. Your changes are now persistent.

Remove a User

1. Click the Users node in the left pane. A dialog displays in the right pane showing the controls for working with users.
2. Enter the name of the user you want to remove in the Remove These Users list. You can remove multiple users by separating each name in the list with a blank.
3. Click Remove to delete the user you specified. The user is removed from the Users list.

Configuration

Attribute	Description	Range of Values	Default Value
Name	The name of a User, that is, an entity that will access WebLogic Server resources.	String Names are case-sensitive.	
Password	The password for the User.	The password must contain a minimum of 8 characters in length. Passwords are case-sensitive.	

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for optional user-supplied information.	The value must be an alphanumeric string.	Null

111 VirtualHost

The following procedures describe how to use the Administration Console to configure and manage VirtualHosts. For additional information about VirtualHosts, please see the *Administration Guide*, Overview of WebLogic Server HTTP Services.

Configure a VirtualHost

1. Click the VirtualHost node. The VirtualHosts table displays in the right pane showing all the VirtualHosts defined in your domain.
2. Click the Configure a New VirtualHost text link. A dialog displays in the right pane showing the tabs associated with configuring a new VirtualHost.
3. Enter values in the Name and Virtual Host Names attribute fields, and select a Default Web App from the drop-down list.
4. Click the Logging and HTTP tabs individually and change the attribute fields or accept the default values as assigned.
5. Click Create to create a web-server instance with the name you specified in the Name field. The new instance is added under the VirtualHost node in the left pane.
6. Restart the server for the Virtual Host to take effect.

For additional information about VirtualHosts, please see the *Administration Guide*, Configuring VirtualHosts.

Clone a VirtualHost

1. Click the VirtualHost node. The VirtualHosts table displays in the right pane showing all the VirtualHosts defined in your domain.
2. Click the Clone icon in the row of the VirtualHost you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a VirtualHost.
3. Enter values in the Name and Virtual Host Names attribute fields, and select a Default Web App from the drop-down list.
4. Click the Logging and HTTP tabs individually and change the attribute fields or accept the default values as assigned.
5. Click Create to create a web-server instance with the name you specified in the Name field. The new instance is added under the VirtualHost node in the left pane.

For additional information about VirtualHosts, please see the *Administration Guide*, Configuring VirtualHosts.

Delete a VirtualHost

1. Click the VirtualHost node. The VirtualHosts table displays in the right pane showing all the VirtualHosts defined in your domain.
2. Click the Delete icon in the row of the VirtualHost you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click Yes to delete the VirtualHost. The web-server icon under the VirtualHost node is deleted.

For additional information about VirtualHosts, please see the *Administration Guide*, Overview of WebLogic Server HTTP Services.

Assign a VirtualHost

1. Click the instance node in the left pane for the VirtualHost you want to assign. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Targets tab.
3. Complete the following steps for the Servers and Clusters tabs:
   a. Select one or more targets in the Available column that you want to assign to the VirtualHost.
   b. Click the mover control to move the targets you selected to the Chosen column.
   c. Click Apply to save your assignments.

For additional information about VirtualHosts, please see the *Administration Guide*, Overview of WebLogic Server HTTP Services.

Target Web Applications to the Virtual Host.

1. Click the Web Applications node in the left panel.
2. Select the Web Application you want to target.
3. Click the Targets tab in the right panel.
4. Click the Virtual Hosts tab.
5. Click a Virtual Host in the available column and use the right arrow button to move the Virtual Host to the chosen column.

For additional information about VirtualHosts, please see the *Administration Guide*, Overview of WebLogic Server HTTP Services.

Monitor All Instances of a VirtualHost

1. Click the VirtualHost node. The VirtualHosts table displays in the right pane showing all the VirtualHosts defined in your domain.
2. Click the Monitor All Instances icon in the row of the VirtualHost you want to monitor. A dialog displays in the right pane showing all instances of the VirtualHost deployed across the server domain.

For additional information about VirtualHosts, please see the *Administration Guide*, Overview of WebLogic Server HTTP Services.

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	This attribute returns the name of the VirtualHost.	String	Null
Virtual Host Name	This attribute returns the host name for which this VirtualHost will serve requests.	String	Null
Default Web Application	This attribute sets the name of the default Web application.	String	Null

Logging

Attribute	Description	Range of Values	Default Value
Enable Logging	This attribute sets whether or not HTTP access logs are generated.	Boolean True = access logs are generated False = access logs are not generated	true
LogFile Name	If HTTP access logging is enabled, this attribute sets the name of the log file.	String To include a time or date stamp in the file name when the log file is rotated, add `java.text.SimpleDateFormat` variables. Surround each variable with percentage (%) characters. For example, `access_%yyyy%_%MM%_%dd%_%hh%_%mm%.log` If you do not include a time and date stamp, the rotated log files are numbered in order of creation *`filename`*`nnnnn`, where *`filename`* is the name configured for the log file.	Null

Attribute	Description	Range of Values	Default Value
Format	If HTTP access logging is enabled, this attribute sets the type of log file.	`common` = uses standard HTTP access log format `extended` = uses extended HTTP access log format	common
Log Buffer Size	If the `Rotation Type` is set to `size`, this attribute sets the maximum size of the log file, in kilobytes. When the log file reaches this size it rotates.	Integer	8 Kilobytes
Rotation Type	This attribute defines whether the HTTP log file is rotated by date or rotated based on the size of the log file.	`size`= log rotates based on its size. You can also override the default value of the Log Buffer Size attribute `date` = log rotates based on date. You can also override the default value of the Rotation Period attribute and/or set the Rotation Time attribute.	size
Rotation Period	If the `Rotation Type` is set to `date`, this attribute defines the number of minutes after which the access log rotates.	Integer	2147483647 minutes
Flush Every	If the `Rotation Type` is set to `date`, this attribute sets the number of seconds after which any new data is written to the log file.	Integer	60 Seconds

Attribute	Description	Range of Values	Default Value
Rotation Time	If the `Rotation Type` is set to `date`, this attribute defines the date and time of the first rotation of the access log.	The format follows `java.text.SimpleDateFormat`, MM-dd-yyyy-k:mm:ss	
Maximum Log File Size	This attribute allows the user to set the maximum allowed size for the log file.	Integer in KBytes	

HTTP

Attribute	Description	Range of Values	Default Value
Default Server Name	When WebLogic Server redirects a request, it sets the host name returned in the HTTP response header with the string specified with Default Server Name. Useful when using firewalls or load balancers and you want the redirected request from the browser to reference the same host name that was sent in the original request.	String	Null

Enable Keepalives	This attribute sets whether or not HTTP keep-alive is enabled	Boolean
		True = enabled
		False = not enabled
			Selected

Send Server Header	If set to false, the server name is not sent with the HTTP response. Useful for wireless applications where there is limited space for headers.	Boolean
		True = enabled
		False = not enabled
			True

Duration	The number of seconds that WebLogic Server waits before closing an inactive HTTP connection.	Integer	30

Attribute	Description	Range of Values	Default Value
HTTPS Duration	The number of seconds that WebLogic Server waits before closing an inactive HTTPS connection.	Integer	60
WAP Enabled	When selected, the session ID no longer includes JVM information. This may be necessary when using URL rewriting with WAP devices that limit the size of the URL to 128 characters. Selecting WAP Enabled may affect the use of replicated sessions in a cluster.	Enabled Disabled	not Enabled
Post Timeout Secs	This attribute sets the timeout (in seconds) that WebLogic Server waits between receiving chunks of data in an HTTP POST data. Used to prevent denial-of-service attacks that attempt to overload the server with POST data.	Integer	0
Max Post Time	This attribute sets the time (in seconds) that WebLogic Server waits for chunks of data in an HTTP POST data.	Integer	0
Max Post Size	This attribute sets the size of the maximum chunks of data in an HTTP POST data.	Integer	0

For more information on setting up HTTP access logs, see the Setting Up HTTP Access Logs section of the *WebLogic Server Administration Guide*

Targets

Servers

Attribute	Description	Range of Values	Default Value
Target Servers	This attribute selects which servers this VirtualHost is deployed on.	List	Null

Clusters

Attribute	Description	Range of Values	Default Value
Target Clusters	This attribute is a list which allows you to select one or more targets for use with this webserver.	List	Null

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for optional user-supplied information.	The value must be an alphanumeric string containing up to 256 characters.	Null

For additional information about VirtualHosts, please see the *Administration Guide,* Configuring VirtualHosts.

112 Web Applications

These procedures describe how to use the Administration Console to set the attributes for installing Web Applications. For more information about configuring Web Applications, please see *Assembling and Configuring Web Applications*.

Install a New Web Application

1. Click the Web Applications node in the left pane. The Web Applications table displays in the right pane showing all the deployed Web Applications.
2. Enter the path of the `.war` file or the folder containing a Web Application in exploded format in the text field, or click Browse to select it in your file system.
3. Click Upload to install the application. The new Web Application is added under the Web Applications node in the left pane. When you choose "Install a new Web Application", the application is copied to applications directory, and the Web Application name defaults to the name of the `.war` file.

Configure a New Web Application

1. Click the Web Applications node in the left pane.

   The Web Applications table displays in the right pane showing all the deployed Web Applications.

2. Enter the path of the `.war` file or folder containing a Web Application in exploded format in the text field, or click the Browse button to browse your file system and choose the application you want to install.
3. Click Upload to install the application.

If you select the "Configure a new Web App" option , you can specify a Web Application name, URI, path and other options. The advantages to choosing “Configure a new Web App” are

1. The `.war` file is not copied in to the applications directory
2. You can specify the targets for this webapp.

Configuration

General

Attribute	Description	Range of Values	Default Value
Name	This attribute sets the name of the Web Application. It is the context path used in URLs to access components in the Web Application.	String	Null
Path URI	The full path to the Web Application in the file system. For a Web Application in exploded directory format, use the path to the directory. For a Web Application in archived `.war` format, use the path to the `.war` file. Use an absolute path or a path relative to the *path attribute* of a J2EE application that contains this Web Application.	String	Null

Attribute	Description	Range of Values	Default Value
Deployment Order	The order to be used for deployment. Ordering is with respect to other deployable units in the same class, for example, EJBs and Web Applications. Deployments with the lowest DeploymentOrder are deployed first. There is no guarantee of the order of deployments with equal DeploymentOrder values. There is no guarantee of ordering across clusters.	Range of values: Minimum 0 Maximum: Integer.MAX_VALUE	1000
Deployed	If true, the Web Application is deployed.	Boolean Deployed = selected Not Deployed = not selected	Not Selected

Files

Attribute	Description	Range of Values	Default Value
Index Directories	This attribute sets whether or not to automatically generate an HTML directory listing if no suitable index file is found	Boolean True = A directory listing will be returned if a suitable index file cannot be served. False = A directory listing will never be returned.	Selected
Reload Period	This attribute sets the amount of time WebLogic Server waits before checking to see if a servlet has been modified.	-1=WebLogic Server never checks for modified servlets 0=WebLogic Server always checks to see if a servlet has been modified and reloads the modified servlet. 1-maximum integer = number of seconds WebLogic Server waits before checking for a modified servlet.	1

Case Sensitive Extensions	This attribute sets whether the case of the file extension is considered when resolving HTTP requests.	Boolean
		True = case is considered
		False = case is ignored
			Not Selected

Other

Attribut e	Description	Range of Values	Default Value
Single Threaded Servlet Pool Size	This attribute defines the size of the pool used for SingleThreadMode instance pools.	Integer	5
Auth Realm Name	This attribute sets the Realm in the Basic Authentication HTTP dialog box which appears on the browsers.	string	weblogic

Targets

Servers

Attribut e	Description	Range of Values	Default Value
Target Servers	This attribute is a list which allows you to select one or more targets for use with this webserver.	List	Null

Clusters

Attribut e	Description	Range of Values	Default Value
Target Clusters	This attribute is a list which allows you to select one or more targets for use with this webserver.	List	Null

VirtualHosts

Attribute	Description	Range of Values	Default Value
Target Virtual Hosts	This attribute is a list which allows you to select one or more targets for use with this webserver.	List	Null

Monitoring

Attribute	Description	Range of Values	Default Value

Enable Session Monitoring	If true, runtime Mbeans will be created for sessions, otherwise, they will not.	Boolean
		Enabled = selected
		Disabled = not selected
			Not Selected

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for optional user-supplied information.	The value must be an alphanumeric string.	Null

For more information about configuring Web Applications, please see Assembling and Configuring Web Applications.

For information about writing deployment descriptors, please see Writing Web Application Deployment Descriptors.

113 WebApp Component Runtime

By default, this pane allows the user to sort the objects by the following criteria:

- Server
- Machine
- Source info
- Servlets
- Session
- Sessions high
- Total sessions

114 Web Application Deployment Descriptor Editor Help

The following sections contains information about using the Web Application Deployment Descriptor Editor:

- Overview of the Web Application Deployment Descriptor Editor
- Configuring a Web Application using the Web Application Deployment Descriptor Editor
- Creating a New Web Application
- Using the Web Application Deployment Descriptor Editor
- Editing an Existing Deployment Descriptor
- Instructions for configuring Web Application components (The component name is followed by the top-level XML element used to represent it in the deployment descriptor):
  - Configuring Basic Web Application Attributes (`<icon>, <display-name>, <description>, <distributable>`)
  - Configuring Context Parameters (`<context-param>)`)
  - Configuring Filters (`<filter>, <filter-mapping>`)
  - Configuring Listeners (`<listener>`)
  - Configuring Servlets (`<servlet>`, `<servlet-mapping>`)

    - Configuring Welcome Pages (`<welcome-file-list>`)
    - Configuring the Session Timeout (`<session-config>`)
    - Configuring MIME Mappings (`<mime-mapping>`)
    - Configuring Error Pages (`<error-page>`)
    - Configuring JSP Tag Libraries (`<taglib>`)
    - Configuring Resource Environment References (`<resource-env-ref>`)
    - Configuring Resource References (`<resource-ref>`)
    - Configuring Security Constraints (`<security-constraint>`)
    - Configuring Logins (`<login-config>`)
    - Configuring Security Roles (`<security-role>`)
    - Configuring Environment Entries (`<env-entry>`)
    - Configuring EJB References (`<ejb-ref>`)
- Configuring a New Webapp Ext (`weblogic.xml`)
    - Configuring the Session Descriptor (`<session-descriptor>`)
    - Configuring the JSP Descriptor (`<jsp-descriptor>`)
    - Configuring Security Role Assignments (`<security-role-assignment>`)
    - Configuring a Reference Descriptor (`<reference-descriptor>`)
    - Configuring Character Set Parameters (`<charset-params>`)
    - Configuring a Container Descriptor (`<container-descriptor>`)
- Configuring a WebLogic Web Service

Overview of the Web Application Deployment Descriptor Editor

The Web Application Deployment Descriptor Editor allows you to edit the deployment descriptors that define a Web Application. A Web Application is a J2EE deployment unit that defines a collection of Web resources such as JSPs, servlets, and HTML pages. Web Applications can also define references to external resources such as EJBs.

Each Web Application has two deployment descriptors, `web.xml` and `weblogic.xml`, both of which can be edited using the Web Application Deployment Descriptor Editor. The `web.xml` deployment descriptor is defined by the Servlet specification from Sun Microsystems and the `weblogic.xml` deployment descriptor is specific to WebLogic Server. For more information about Web Applications, see *`Assembling and Configuring Web Applications`* in the WebLogic Server Documentation at `http://e-docs.bea.com/wls/docs61/webapp/index.html`.

The Web Application Deployment Descriptor Editor can only edit an existing deployment descriptor, it cannot create a new deployment descriptor.

Creating a New Web Application

To create a new Web Application:

1. Create the components of the Web Application: the servlets, JSPs and HTML pages.

2. Arrange the components on your file system into the directory structure defined for Web Applications. For more information, see Directory Structure `http://e-docs.bea.com/wls/docs61/webapp/basics.html#dir`.

3. Create a skeleton deployment descriptor. You can do this one of two ways:

   - Follow the procedures described under Packaging Web Applications at `http://e-docs.bea.com/wls/docs61/programming/packaging.html#pack005`.

- Create a text file called `web.xml` and save it in the `WEB-INF` directory of your Web Application. This file must contain the following text:

  ```
  <!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//
  DTD Web Application 2.3//EN"
  "http://java.sun.com/dtd/web-app_2_3.dtd">

  <web-app>
  </web-app>
  ```

4. Configure the functionality of your Web Application by using the deployment descriptor editor as described in the next section.

Using the Web Application Deployment Descriptor Editor

1. Start the WebLogic Administration Server if it is not already running.
2. Start the WebLogic Server Administration Console in a browser.
3. If the Web Application is already defined in the current domain, expand the Web Applications node in the left panel and skip to Step 5.
4. If the Web Application is not yet configured in the current domain, configure a new Web Application:

   a. Click on “Configure a new Web Application” in the right panel.

   b. Enter the name of this Web Application in the Name field.

   c. Enter the name of the directory containing your Web Application in the URI field.

   d. Enter the path to the directory containing your Web Application in the Path field.

   e. Click Create. The new Web Application appears under Web Applications in the left panel

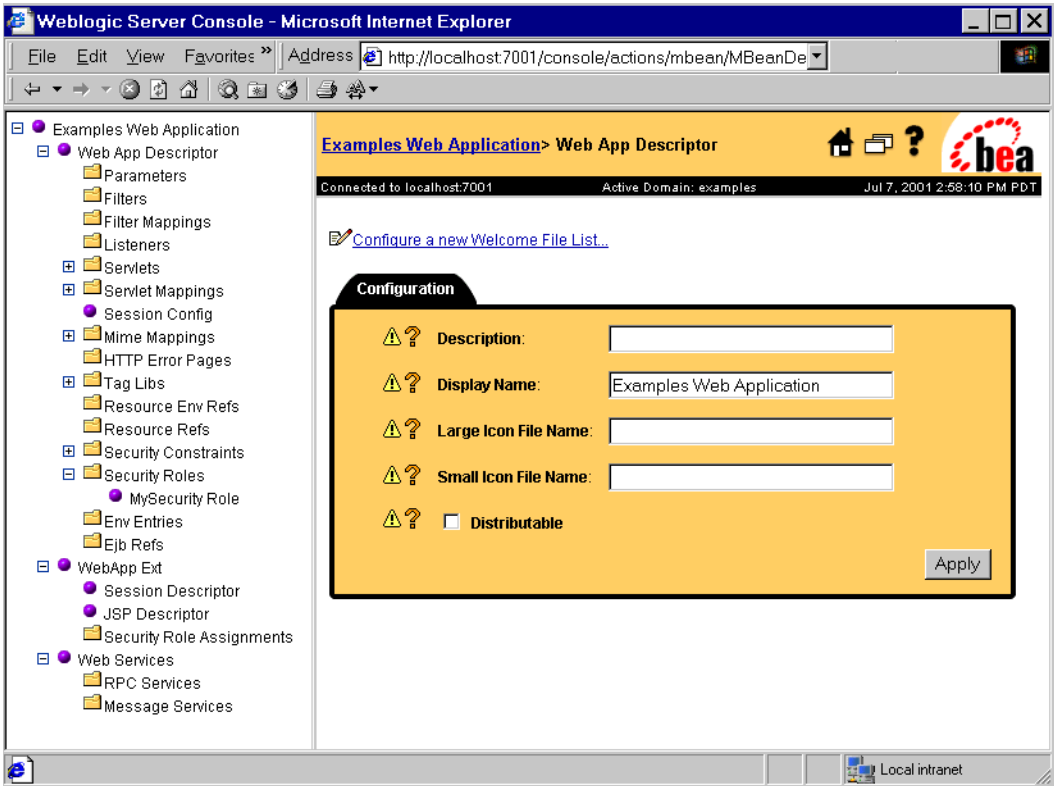
5. Right click on the Web Application and select Edit Web Application Descriptor. A new browser window opens in which you can edit the deployment descriptors.
6. Configure the functionality of your Web Application. See Configuring a Web Application using the Web Application Deployment Descriptor Editor for instructions on configuring Web Application functionality.

7. When you have finished editing the deployment descriptor, select the Web App Descriptor node in the left navigation tree.

8. Click Validate in the right panel. The validation only checks to see that the deployment descriptor conforms to its document type definition (DTD). It does not verify the existence of any of the components you have configured nor perform any other type of validation.

9. After you have validated the deployment descriptors, click Persist in the right panel. Your changes to the deployment descriptor are not written to the deployment descriptor files on disk until you click the Persist button.

Figure 114-1 Web Application Deployment Descriptor Editor


Weblogic Server Console - Microsoft Internet Explorer
File
Edit
View
Favorites
Address
http://localhost:7001/console/actions/mbean/MBeanDe
Local intranet

- Examples Web Application
  - Web App Descriptor
    - Parameters
    - Filters
    - Filter Mappings
    - Listeners
    - Servlets
    - Servlet Mappings
    - Session Config
    - Mime Mappings
    - HTTP Error Pages
    - Tag Libs
    - Resource Env Refs
    - Resource Refs
    - Security Constraints
    - Security Roles
      - MySecurity Role
    - Env Entries
    - Ejb Refs
  - WebApp Ext
    - Session Descriptor
    - JSP Descriptor
    - Security Role Assignments
  - Web Services
    - RPC Services
    - Message Services

Examples Web Application> Web App Descriptor

Connected to localhost:7001 Active Domain: examples Jul 7, 2001 2:58:10 PM PDT

Configure a new Welcome File List...

Configuration

Description:	
Display Name:	Examples Web Application
Large Icon File Name:	
Small Icon File Name:	

☐ Distributable

Apply

Editing an Existing Deployment Descriptor

1. Follow the procedure Using the Web Application Deployment Descriptor Editor
2. Select the deployment descriptor elements you wish to edit from the navigation tree in the left panel. Some elements require you to right click on their parent nodes in the left navigation tree and select the option to create the element. For more information, see Configuring a Web Application using the Web Application Deployment Descriptor Editor
3. Fill in or change the fields in the right panel as required.
4. Click the Create or Apply button as appropriate.
5. When you have finished editing the deployment descriptor, select the top node in the left navigation tree.
6. Click Validate in the right panel. The validation only checks to see that the deployment descriptor conforms to its document type definition (DTD). It does not verify the existence of any of the components you have configured nor perform any other type of validation.
7. Click Persist in the right panel. Your changes to the deployment descriptor are not written to the deployment descriptor files on disk until you click the Persist button.

Configuring a Web Application using the Web Application Deployment Descriptor Editor

The procedures in this section describe how to create the deployment information required for a Web Application. You should have created a basic deployment descriptor and have that Web Application open in the Web Application Deployment Descriptor Editor as described in Creating a New Web Application and Using the Web Application Deployment Descriptor Editor.

The sections that follow provide procedures for configuring the components of a Web Application. Tables of configurable attributes follow the procedures and are also linked from the appropriate procedures.

Configuring Basic Web Application Attributes

1. Click on the Web App Descriptor node in the left panel
2. Configure the following elements on the Web App Descriptor panel as required:
   - Description
   - Display Name
   - Large Icon File Name
   - Small Icon File Name
   - Distributable

Configuring Context Parameters

Context parameters are used to pass name/value pairs to the servlet container. You can enter any number of context parameter name/value pairs.

1. Right click on the Parameters node in the left panel and select Configure a new Parameter.
2. Enter a description for the Parameter in the Description field. (optional)
3. Enter the name of the parameter in the Param Name field.
4. Enter the value of the parameter in the Param Value field.
5. Click Create

You can access these parameters in your code using the `javax.servlet.ServletContext.getInitParameter()` and `javax.servlet.ServletContext.getInitParameterNames()` methods.

Configuring Filters

Filters intercept requests for Web Application resources to perform additional functions. To configure a filter, you create a filter and then map it to a servlet or URL pattern. You can configure any number of filters or filter mappings. For more information, see Filters at `http://e-docs.bea.com/wls/docs61/webapp/filters.html`.

Filters are a part of the proposed final draft of the Servlet 2.3 specification from Sun Microsystems. If you are planning to use filters in your application, note that the specification has not been finalized and could change in the future.

1. Right click on the Filters node in the left panel and select Configure a new Filter.
2. Fill in the following fields:
   - Filter Name (required) The name of this filter.
   - Description (optional)
   - Display Name (optional)
   - Small Icon File Name (optional)
   - Large Icon File Name (optional)
   - Filter Class (required) The full class name of the Java class that executes the filter. For example, `myApp.filters.myFilter`.
3. Click Create
4. Right click on the Filter Mappings node in the left panel and select Configure a new FilterMapping. You can map a filter to a URL pattern or a servlet that is already configured in this Web Application. You may create more than one mapping for each filter.
5. Select the filter you are mapping from the Filter drop-down list.
6. If you are mapping this filter to a URL pattern, enter it in the Url Pattern field.
7. If you are mapping this filter to a servlet, select a servlet from the Servlet drop-down list. You must configure a servlet before you can map it to a filter.
8. Click Create

9. Create Filter initialization parameters. Filter initialization parameters can be read by your filter class using the `FilterConfig.getInitParameter()` or `FilterConfig.getInitParameters()` methods.

   a. Expand the Filters node in the left panel.

   b. Expand the node of the filter for which you want to create initialization parameters.

   c. Right click on the Parameters node and select Configure a new Parameter.

   d. Enter a description for the Parameter in the Description field (optional).

   e. Enter the name of the parameter in the Param Name field.

   f. Enter the value of the parameter in the Param Value field.

   g. Click Create.

10. Repeat these steps to create additional filter mappings.

Configuring Listeners

Listeners are Java classes that respond to HTTP session or servlet context events. You can create any number of listeners. For more information, see Application Events and Listeners at `http://e-docs.bea.com/wls/docs61/webapp/app_events.html`.

Application events and listeners are a part of the proposed final draft of the Servlet 2.3 specification from Sun Microsystems. If you are planning to use application events and listeners in your application, note that the specification has not been finalized and could change in the future.

1. Right click on the Listeners node in the left panel and select Configure a new Listener

2. Enter the full name of the listener class in the Listener Class Name field.

3. Click Create.

Configuring Servlets

To configure a servlet, define the servlet and then map the servlet to a URL pattern. You can create any number of servlets. For more information, see Configuring Servlets `http://e-docs.bea.com/wls/docs61/webapp/components.html#configuring-servlets`.

1. Right click on the Servlet node in the left panel and select Configure a new Servlet.
2. Fill in the following fields:
   - Description (optional)
   - Display Name (optional)
   - Small Icon File Name (optional)
   - Large Icon File Name (optional)
   - Servlet Name (required)
   - Servlet Class. The Java class that executes this servlet.
     or
     Jsp File. A JSP file that is executed this servlet.
   - Load On Startup. (optional)
3. Click Create.
4. Right Click on Servlet Mappings and select Configure a new ServletMapping.
5. Select the servlet you want to map from the Servlet drop-down list.
6. Enter a URL pattern for this servlet. For more information about URL patterns, See Servlet Mapping at http://e-docs.bea.com/wls/docs61/webapp/components.html#servlet-mapping.
7. Click Create
8. Create Servlet initialization parameters. (optional).
   a. Expand the Servlet node in the left panel.
   b. Expand the node of the servlet for which you want to create initialization parameters.

   c. Right click on the Parameters node in the left panel and select Configure a new Parameter.

   d. Enter a description for the Parameter in the Description field (optional).

   e. Enter the name of the parameter in the Param Name field.

   f. Enter the value of the parameter in the Param Value field.

   g. Click Create.

   To retrieve initialization parameters, call the `getInitParameter(String name)` method from the parent `javax.servlet.GenericServlet` class. When passed the name of the parameter, this method returns the parameter's value as a `String`.

9. Create Servlet Security Role Refs.

   a. Expand the Servlet node in the left panel.

   b. Expand the node of the servlet for which you want to create Security Role Refs.

   c. Right click on the Security Role Refs node and select Configure a new SecurityRoleRef.

   d. Enter a description for the Security Role Ref in the Description field (optional).

   e. Select a security role from the Role Link drop-down list. The security role must be previously defined on the Security Roles panel.

   f. Enter the role name used in your servlet code in the Role Name field.

Configuring Welcome Pages

When the HTTP request is for a directory name, WebLogic Server serves the first file specified in this list. If that file is not found, the server then tries the next file in the list. For more information, see Configuring Welcome Pages at `http://e-docs.bea.com/wls/docs61/webapp/components.html#welcome_pages`.

1. Right click on the Web App Descriptor node in the left panel and select Configure a new WelcomeFileList.

2. Enter Welcome Files names, one per line.

3. Click Create.

Configuring the Session Timeout

Session Timeout set the time after which HTTP sessions in this Web Application expire.

1. Select the Session Config node in the left panel.
2. Edit the Session Timeout value.
3. Click Apply.

Configuring MIME Mappings

MIME mappings defines a mapping between a file extension and a MIME type. You can create any number of MIME mappings.

1. Right click on MIME Mappings in the left panel and select Configure a new MIME Mapping.
2. Enter a valid MIME type in the Mime Type field.
3. Enter the file extension you want to map in the Extension field.
4. Click Create

Configuring Error Pages

An error page is a JSP or HTML page that you configure to be displayed in response to an HTTP error code or Java exception. You can configure different error pages to respond to various HTTP error codes or Java exceptions. For more information, see Customizing HTTP Error Responses at `http://e-docs.bea.com/wls/docs61/webapp/components.html#error-page`.

1. Right Click on Error Pages in the left panel and select Configure a new ErrorPage.

2. Enter an HTTP Error Code in the Error Code field
   or
   Enter a Java Exception class in the Exception Type field.
3. Enter the name of the resource to display in response to the error. The entry must include a leading `/`. For example `/myErrorPg.html`.
4. Click Create.

Configuring JSP Tag Libraries

JSP tag libraries contain the Java classes and descriptors that define a user-written JSP tag. You can create any number of tag libraries. You can configure one or more JSP tag libraries. For more information, see *Programming WebLogic JSP Tag Extensions* at `http://e-docs.bea.com/wls/docs61/taglib/index.html`.

1. Right click on the Tag Libs node in the left panel and select Configure a new TagLib.
2. Enter the name (relative to the `WEB-INF` directory of the Web Application) you will use to refer to this tag library in the JSP `taglib` directive in the URI field.
3. Enter the location (relative to the root directory of the Web Application) of the tag library or tag library jar file in the Location field.
4. Click Create.

Configuring Resource Environment References

Resource environment references are a part of the proposed final draft of the Servlet 2.3 specification from Sun Microsystems and are not currently implemented in WebLogic Server.

1. Right click on the Resource Env Refs node in the left panel and select Configure a new ResourceEnvRef.
2. Enter a description for this Resource Env-Ref in the Description (optional).
3. Enter the name of this Resource Env-Ref in the Ref Name field.

4. Enter the Java type for this Resource Env-Ref in the Ref Type field.
5. Click Create

Configuring Resource References

A resource reference defines a lookup name for an external resource. Your servlet code can look up a resource by this “virtual” name that is mapped to the actual location at deployment time. You can create any number of resource references.

1. Right click on the Resource Refs node in the left panel and select Configure a new ResourceRef.
2. Enter a description for this Resource Ref in the Description field (optional).
3. Enter the name of this Resource Ref in the Ref Name field.
4. Enter the Java type of the Resource Ref in the Ref Type field. Enter the full package name of the Java type.
5. Enter the authorization type in the Auth field. Valid values are `APPLICATION` or `CONTAINER.`
6. Enter the sharing scope in the Sharing Scope field.Valid values are `Sharable` and `Unsharable`.
7. Click Create.

Configuring Security Constraints

Security constraints apply security to specified resources. You can create any number of security constraints. For more information, see Restricting Access to Resources in a Web Application at `http://e-docs.bea.com/wls/docs61/webapp/security.html#resources-webapp`.

1. Right click on the Security Constraints node in the left panel and select Configure a new SecurityConstraint.
2. Enter a name for this Security Constraint in the Display Name field.

3. Click Create.

4. Expand the Security Constraint node in the left panel

5. Expand the node for the Security Constraint you are adding.

6. Create a Web Resource Collection. A Web Resource Collection defines the URL pattern and HTTP method that is constrained by this security constraint.

    a. Right click on Web Resource Collection and select Configure a new WebResourceCollection.

    b. Enter a name for this Web Resource Collection in the Web Resource Name field.

    c. Enter a description for this Web Resource Collection in the Description field. (optional)

    d. Enter one or more URL patterns, one per line, in the URL Pattern field. The URL pattern defines which resources are subject to this security constraint.

    e. Enter a HTTP method in the HTTP Method box. This entry is usually `GET` or `POST`.

    f. Click Create.

7. Create an Auth Constraint

    a. Right click on the name of the Security Constraint you are editing and select Configure a new AuthConstraint.

    b. Enter a description of this auth constraint in the Description field. (optional)

    c. Click Create. A list of previously defined security roles appears in the Available panel.

    d. Use the arrow buttons to move security roles from the available panel to the chosen panel. Note: You must first Create one or more security roles before you can assign them here. For more information, see Configuring Security Roles.

    e. Click Apply.

8. Create a User Data Constraint

    a. Right click on the name of the Security Constraint you are editing and select Configure a new UserDataConstraint.

b. Enter a description of this User Data Constraint in the Description field. (optional)

c. Select the transport guarantee from the Transport Guarantee drop-down list. Valid values are `NONE`, `INTEGRAL`, or `CONFIDENTIAL`.

d. Click Create.

Configuring Logins

Login configuration describes how the user is authenticated, the realm name that should be used for this application, and the attributes that are needed by the form login mechanism.

If this element is present, the user must be authenticated in order to access any resource that is constrained by Security Constraints defined in the Web application. Once authenticated, the user can be authorized to access other resources with access privileges.

1. Right click on the Web App Descriptor node and select Configure a new LoginConfig.
2. Select the Authorization method from the Auth Method drop-down list. Valid values are `BASIC`, `FORM`, or `CLIENT-CERT`.
3. Fill in the name of the authentication realm in the Realm Name field.
4. If you set the Auth Method to `FORM`, enter the URI of the file containing the authentication form in the Login Page field.
5. If you set the Auth Method to `FORM`, enter the URI of the file containing the page a user is redirected to when authentication fails in the Error Page field.

Configuring Security Roles

A Security Role defines a security context for principals (usually a user name) assigned to the role. You can create any number of security roles. For more information, see Restricting Access to Resources in a Web Application at `http://e-docs.bea.com/wls/docs61/webapp/security.html#resources-webapp`.

1. Right click on the Security Roles node in the left panel and select Configure a new SecurityRole.
2. Enter a description for this Security Role in the Description field.(optional).
3. Enter the name of this security role in the Role Name field.
4. Click Create.

Configuring Environment Entries

Declares an environment entry for an application. You can create any number of environment entries.

1. Right click on the Env Entries node in the left panel and select Configure a new EnvEntry.
2. Enter a description for this Environment entry in the Description field.(optional).
3. Enter the name of this Env entry in the Env Entry Name field.
4. Enter the value of this Env Entry in the Env Entry Value field
5. Enter the type of this Env Entry in the Env Entry Type field.

Configuring EJB References

The ejb-ref defines a reference to an EJB resource. This reference is mapped to the actual location of the EJB at deployment time by defining the mapping in the WebLogic-specific deployment descriptor file, `weblogic.xml`.You can create any number of EJB references.

1. Right click on the Ejb refs node in the left panel and select Configure a new EjbRef.
2. Enter a description for this EJB Reference in the Description field.(optional)
3. Enter the name of this EJB Reference in the EJBRef Name field.
4. Enter the Java class of the referenced EJB in the EJBRef Type field.
5. Enter the name of the home interface of the EJB in the Home Interface Name field.
6. Enter the name of the remote interface of the EJB in the Remote Interface Name field.
7. If the EJB is packaged in an Enterprise Application Archive (`EAR`), enter the name of the EJB as it is referenced in the `<ejb-name>` element of the `EAR's` deployment descriptor in the EJBLink Name field.
8. Enter the name of a security role defined in this web application in the Run As field. (optional)

Configuring a New Webapp Ext

The WebApp Ext node in the left pane represents the WebLogic-specific deployment descriptor, `weblogic.xml`. The `weblogic.xml` descriptor is used to configure attributes unique to WebLogic Server. To configure attributes in `weblogic.xml`, you must first create its node in the deployment descriptor editor.

To create the `weblogic.xml` descriptor node:

1. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)
2. Select Configure a new WebAppExtDescriptor
3. In the right pane, enter a description of the Web Application in the Description field (optional).
4. Enter the version of WebLogic Server that your application is running on (optional).

5. Click Create.

Configuring the Session Descriptor

The Session Descriptor configures HTTP session-related parameters.

1. If it does not exist in the left pane, create the WebApp Ext node:
   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)
   b. Select Configure a new WebAppExtDescriptor
   c. In the right pane, enter a description of the Web Application in the Description field (optional).
   d. Enter the version of WebLogic Server that your application is running on (optional).
   e. Click Create.
   f. Expand the WebApp Ext node in the left panel.
2. Click on the Session Descriptor node in the left panel.
3. Fill in the fields in the right panel according to the information in Session Descriptor table.
4. Click Apply.

Configuring the JSP Descriptor

The JSP Descriptor configures JSP-related parameters.

1. If it does not exist in the left pane, create the WebApp Ext node:
   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

   b. Select Configure a new WebAppExtDescriptor

   c. In the right pane, enter a description of the Web Application in the Description field (optional).

   d. Enter the version of WebLogic Server that your application is running on (optional).

   e. Click Create.

   f. Expand the WebApp Ext node in the left panel.

2. Click on the JSP Descriptor node in the left panel.

3. Fill in the fields in the right panel according to the information in JSP Descriptor table.

4. Click Apply.

Configuring Security Role Assignments

Security role assignments map a security role to one or more principals. The principals must be defined in your security realm.

1. If it does not exist in the left pane, create the WebApp Ext node:

   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

   b. Select Configure a new WebAppExtDescriptor

   c. In the right pane, enter a description of the Web Application in the Description field (optional).

   d. Enter the version of WebLogic Server that your application is running on (optional).

   e. Click Create.

   f. Expand the WebApp Ext node in the left panel.

2. Right click on the Security Role Assignment node in the left panel and select Configure a new SecurityRoleAssignment.

3. Select a security role from the Role drop-down list.
4. Add one or more Principal Names to the text box, one per line. The principals must be valid in your security realm.
5. Click Create.

Configuring Character Set Parameters

You can define codeset behavior for non-unicode operations by configuring Character Set Parameters. You can also specify a mapping of Java character sets to IANA character sets. For more information, see Determining the Encoding of an HTTP Request at `http://e-docs.bea.com/wls/docs61/webapp/components.html#encoding-http`, and Mapping IANA Character Sets to Java Character Sets at `http://e-docs.bea.com/wls/docs61/webapp/components.html#map-iana`.

1. If it does not exist in the left pane, create the WebApp Ext node:
   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)
   b. Select Configure a new WebAppExtDescriptor
   c. In the right pane, enter a description of the Web Application in the Description field (optional).
   d. Enter the version of WebLogic Server that your application is running on (optional).
   e. Click Create.
   f. Expand the WebApp Ext node in the left panel.
2. Right click on the WebApp Ext node in the left panel and select Configure a new Charset Params.
3. Create Input Charset Descriptors
   a. Expand the Charset Params node.

   b. Right click on the Input Charset Descriptors node and select Configure a new InputCharsetDescriptor.

   c. Enter the path that applies to this Charset Descriptor in the Resource Path field.

   d. Enter a valid Java character set name in the Java Charset Name field.

   e. Click Create

4. Create a Character Set Mapping

   a. Expand the Charset Params node.

   b. Right click on the Charset Mappings node and select Configure a new CharsetMapping.

   c. Enter a valid IANA character set name in the IANA Charset Name field.

   d. Enter a valid Java character set name in the Java Charset Name field.

   e. Click Create

Configuring a Reference Descriptor

The Reference Descriptor maps the JNDI name of a server resource to a name used in the Web application. The Resource Description panel maps a resource, for example, a DataSource, to its JNDI name. The Ejb Reference panel maps an EJB to its JNDI name.

1. If it does not exist in the left pane, create the WebApp Ext node:

   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

   b. Select Configure a new WebAppExtDescriptor

   c. In the right pane, enter a description of the Web Application in the Description field (optional).

   d. Enter the version of WebLogic Server that your application is running on (optional).

   e. Click Create.

   f. Expand the WebApp Ext node in the left panel.

2. Right click on the WebApp Ext node in the left panel and select Configure a new Reference Descriptor.

3. Create Resource Descriptions. You can create one or more Resource Descriptions.

   a. Expand the Resource Descriptor node.

   b. Right click on the Resource Descriptions node and select Configure a new ResourceDescription.

   c. Enter the name of an object from the JNDI tree in the Jndi Name field.

   d. Select a Resource Reference from the Resource Reference drop-down list. The Resource Reference should be already configured under the Resource Refs tab.

   e. Click Create

4. Create an EJB Reference Description

   a. Expand the Resource Descriptor node.

   b. Right click on the Ejb Reference Descriptions node and select Configure a new EjbReferenceDescription.

   c. Enter the name of an object from the JNDI tree in the Jndi Name field.

   d. Select an EJB Reference from the Ejb Reference drop-down list. The EJB Reference should be already configured under the Ejb refs tab.

   e. Click Create

Configuring a Container Descriptor

Enable Check Auth on Forward when you want to require authentication of forwarded requests from a servlet or JSP.

1. If it does not exist in the left pane, create the WebApp Ext node:

   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

   b. Select Configure a new WebAppExtDescriptor

   c. In the right pane, enter a description of the Web Application in the Description field (optional).

   d. Enter the version of WebLogic Server that your application is running on (optional).

   e. Click Create.

   f. Expand the WebApp Ext node in the left panel.

2. Right click on the WebApp Ext node and select Configure a new Container Descriptor.

3. To enable authentication of forwarded requests, check the Check Auth on Forward Enabled box.

4. To configure redirect behavior, check the redirect-with-absoute-url box.

5. Click Create.

Web App Descriptor

Attribute	Description	Range of Values	Default Value
Description	A text description of the servlet.	String	none
Display Name	This element is not used by WebLogic Server.	String	none
Large Icon File Name	Specifies the location for a large (32x32 pixel) `.gif` or `.jpg` image used to represent the Web application in a GUI tool. Currently, this element is not used by WebLogic Server.	String	none
Small Icon File Name	Specifies the location for a small (16x16 pixel) `.gif` or `.jpg` image used to represent the Web application in a GUI tool. Currently, this is not used by WebLogic Server.	String	none
Distributable	This element is not used by WebLogic Server.	Empty	none

Filters

Attribute	Description	Range of Values	Default Value
Filter Name	The name for this filter	String	MyFilter
Description	A text description of the servlet.	String	none
Display Name	This element is not used by WebLogic Server.	String	none
Small Icon File Name	This element is not used by WebLogic Server.	String	none
Large Icon File Name	This element is not used by WebLogic Server.	String	none
Filter Class	The fully-qualified class name of the filter.	Class name	none

Filter Mappings

Attribute	Description	Range of Values	Default Value
Filter	The name of the filter to which you are mapping a URL pattern. This name corresponds to the name you assigned a filter in a `<filter-name>` element.	String	none

Url Pattern	Describes a pattern used to resolve URLs. The portion of the URL after the `http://host:port` + `WebAppName` is compared to the `<url-pattern>` by WebLogic Server. If the patterns match, the filter mapped in this element will be called.
	Example patterns:
	`/soda/grape/*`
	`/foo/*`
	`/contents`
	`*.foo`
	The URL must follow the rules specified in Section 10 of the Servlet 2.2 Specification.
		String	none

Attribute	Description	Range of Values	Default Value
Servlet	The name of a servlet which, if called, causes this filter to execute.	A valid servlet name defined in this deployment descriptor.	none

Listeners

Attribute	Description	Range of Values	Default Value
Listener Class Name	Name of the class that responds to a Web Application event.	Class name	none

Servlets

Attribute	Description	Range of Values	Default Value
Description	A text description of the servlet.	String	None
Display Name	This element is not used by WebLogic Server.	String	None
Small Icon File Name	This element is not used by WebLogic Server.	String	None
Large Icon File Name	This element is not used by WebLogic Server.	String	None
Servlet Name	Defines the canonical name of the servlet, used to reference the servlet definition elsewhere in the deployment descriptor.	String	MyServlet
Servlet Class	The fully-qualified class name of the servlet. You may use only one of either the `<servlet-class>`tags or `<jsp-file>` tags in your servlet body.	Class name	None

Attribute	Description	Range of Values	Default Value
Jsp File	The full path to a JSP file within the Web application, relative to the Web application root directory. You may use only one of either the `<servlet-class>`tags or `<jsp-file>` tags in your servlet body.	String	None
Load On Startup	WebLogic Server initializes this servlet when WebLogic Server starts up. The optional contents of this element must be a positive integer indicating the order in which the servlet should be loaded. Lower integers are loaded before higher integers.	Positive Integer If no value is specified, or if the value specified is not a positive integer, WebLogic Server can load the servlet in any order in the startup sequence.	0

Parameters

Attribute	Description	Range of Values	Default Value
Description	Text description of the initialization parameter.		MyError Page
Param Name	Defines the name of this initialization parameter.	String	None

Attribute	Description	Range of Values	Default Value
Param Value	Defines a `String` value for this initialization parameter.	String	None

Security Role Refs

Attribute	Description	Range of Values	Default Value
Description	Text description of the role.	String	MyError Page
Role Name	Defines the name of the security role or principal that is used in the servlet code.	String	None
Role Link	Defines the name of the security role.	String - a security role that is defined in a `<security-role>` element later in the deployment descriptor.	None

Servlet Mappings

Attribute	Description	Range of Values	Default Value
Servlet	The name of the servlet to which you are mapping a URL pattern. This name corresponds to the name you assigned a servlet in a `<servlet>` declaration tag.	A declared servlet.	none

URL Pattern	Describes a pattern used to resolve URLs. The portion of the URL after the `http://host:port` + `WebAppName` is compared to the `<url-pattern>` by WebLogic Server. If the patterns match, the servlet mapped in this element will be called.
	Example patterns:
	`/soda/grape/*`
	`/foo/*`
	`/contents`
	`*.foo`
	The URL must follow the rules specified in Section 10 of the Servlet 2.2 Specification.
		String	MyServlet Mapping

Mime Mappings

Attribute	Description	Range of Values	Default Value
Mime Type	A string describing the defined mime type, for example: `text/plain.`	String	MyMime Mapping
Extension	A string describing an extension, for example: `txt.`	String	None

Session Config

Attribute	Description	Range of Values	Default Value
Session Timeout	The number of minutes after which sessions in this Web Application expire. The value set in this element overrides the value set in the `TimeoutSecs` parameter of the Session Descriptor in the WebLogic-specific deployment descriptor `weblogic.xml`, unless one of the special values listed here is entered.	Maximum value: Integer.MAX_VALUE ÷ 60 Special values: ■ -2 = Use the value set by `TimeoutSecs` in Session Descriptor ■ -1 = Sessions do not timeout. The value set in Session Descriptor is ignored.	-2

Welcome Files

Attribute	Description	Range of Values	Default Value
Welcome File	File name to use as a default welcome file, such as index.html. You may specify one or more welcome files.	String	None

Error Pages

Attribute	Description	Range of Values	Default Value
Error Code	A valid HTTP error code, for example `404.`	String	MyError Page
Exception Type	A fully-qualified class name of a Java exception type, for example `java.lang.`	String	None
Location	The location of the resource to display in response to the error. Must include a leading `/`. For example `/myErrorPg.html.`	String	None

Tag Libs

Attribute	Description	Range of Values	Default Value
URI	Describes a URI, relative to the location of the web.xml document, identifying a Tag Library used in the Web application. If the URI matches the URI string used in the taglib directive on the JSP page, this taglib is used.	String	MyTag Lib
Location	Gives the file name of the tag library descriptor relative to the root of the Web application. It is a good idea to store the tag library descriptor file under the `WEB-INF` directory so it is not publicly available over an HTTP request.	String	None

Resource Env Refs

The resource-env-ref element contains a declaration of an component's reference to an administered object associated with a resource in the component's environment.

Attribute	Description	Range of Values	Default Value
Description	A text description.	String	none
Ref Name	The name of a resource environment reference; its value is the environment entry name used in code.	String	MyResource Env Ref
Ref Type	The type of a resource environment reference.	String	none

Resource Refs

Attribute	Description	Range of Values	Default Value
Description	A text description.	String	none
Ref Name	The name of the resource used in the JNDI tree. Servlets in the Web application use this name to look up a reference to the resource.	String	MyResource Ref
Ref Type	The Java type of the resource that corresponds to the reference name. Use the full package name of the Java type.	Java type	none
Auth	Used to control the resource sign on for security.	`APPLICATION:` The application component code performs resource sign on programmatically. `CONTAINER:` WebLogic Server uses the security context established with login config element.	none
Sharing Scope	Specifies whether connections obtained through the given resource manager connection factory reference can be shared.	`Sharable` `Unsharable`	`Sharable`

Security Constraints

Attribute	Description	Range of Values	Default Value
Display Name	Name of this constraint.	String	MySecurity Constraint

Auth Constraint

Defines which groups or principals have access to the collection of web resources defined in this security constraint.

Attribute	Description	Range of Values	Default Value
Description	A text description of this security constraint.	String	none
Role Name	Defines which security roles can access resources defined in this security-constraint. Security role names are mapped to principals using the `Role Name.`	A defined security role.	none

User Data Constraint

Defines how the client should communicate with the server.

Attribute	Description	Range of Values	Default Value
Description	A text description.	String	none
Transport Guarantee	Specifies that the communication between client and server. WebLogic Server establishes a Secure Sockets Layer (SSL) connection when the user is authenticated using the `INTEGRAL` or `CONFIDENTIAL` constraint.s	■ `NONE`—the application does not require any transport guarantees. ■ `INTEGRAL`—the application requires that the data be sent between the client and server in such a way that it cannot be changed in transit. ■ `CONFIDENTIAL`—the application requires that the data be transmitted in a fashion that prevents other entities from observing the contents of the transmission.	none

Web Resource Collection

Defines the components of the Web application to which this security constraint is applied.

Attribute	Description	Range of Values	Default Value
Web Resource Name	The name of this Web resource collection.	String	none
Description	A text description of this security constraint.	String	none

URL Pattern	Use one or more of the URL Pattern elements to declare to which URL patterns this security constraint applies. If you do not use at least one of these elements, this web resource collection is ignored by WebLogic Server.	String	none
HTTP Method	Use one or more of the HTTP Method elements to declare which HTTP methods (`GET | POST |...`) are subject to the authorization constraint. If you omit the HTTP Method element, the default behavior is to apply the security constraint to all HTTP methods.	`GET` `POST`	none

LoginConfig

Attribute	**Description**	**Range of Values**	**Default Value**
Auth Method	Specifies the method used to authenticate the user. Possible values: `BASIC` - uses browser authentication `FORM` - uses a user-written HTML form `CLIENT-CERT`	`BASIC` - uses browser authentication `FORM` - uses a user-written HTML form `CLIENT-CERT`	BASIC

Realm Name	The name of the realm that is referenced to authenticate the user credentials. If omitted, the WebLogic realm is used by default. For more information, see Specifying a Security Realm at http://e-docs.bea.com/wls/docs61/adminguide/cnfgsec.html#cnfgsec004.	String	None
Login Page	The URI of a Web resource relative to the document root, used to authenticate the user. This can be an HTML page, JSP, or HTTP servlet, and must return an HTML page containing a FORM that conforms to a specific naming convention.	String	None
Error Page	The URI of a Web resource relative to the document root, sent to the user in response to a failed authentication login.	String	None

Security Roles

Attribute	Description	Range of Values	Default Value
Description	A text description of this security role.	String	none
Role Name	The role name. The name you use here must have a corresponding entry in the WebLogic-specific deployment descriptor, `weblogic.xml`, which maps roles to principals in the security realm.	String	system

Env Entries

Attribute	Description	Range of Values	Default Value
Description	A textual description.	String	none
Env Entry Name	The name of the environment entry.	String	MyEnvironment Entry
Env Entry Value	The value of the environment entry.	String	none
Env Entry Type	The name of the environment entry.	String	none

Ejb refs

Attribute	Description	Range of Values	Default Value
Description	A text description of the reference.	String	none
EJBRef Name	The name of the EJB used in the Web application. This name is mapped to the JNDI tree in the WebLogic-specific deployment descriptor `weblogic.xml`.	String	MyEJB Ref
EJBRef Type	The expected Java class type of the referenced EJB.	String	none
Home Interface Name	The fully qualified class name of the EJB home interface.	String	none
Remote Interface Name	The fully qualified class name of the EJB remote interface.	String	none
EJBLink Name	The `<ejb-name>` of an EJB in an encompassing J2EE application package.	String	none
Run As	A security role whose security context is applied to the referenced EJB.	A security role defined in this web application.	none

Security Role Assignment

Attribute	Description	Range of Values	Default Value
Role Name	Specifies the name of a security role.	A valid security role.	none
Principal Name	Specifies the name of a principal that is defined in the security realm. You can use multiple `<principal-name>` elements to map principals to a role. For more information on security realms, see the *Programming WebLogic Security* at http://e-docs.bea.com/wls/docs61/security/index.html.	A principal defined in the security realm.	none

Reference Descriptor

Resource Descriptions

Attribute	Description	Range of Values	Default Value
Res Ref Name	Specifies the name of a resource reference.	String	none
Jndi Name	Specifies a JNDI name for the resource.	Java character set name	none

EJB Reference Description

Attribute	Description	Range of Values	Default Value
Ejb Ref Name	Specifies the name of an EJB reference used in your Web application.	String	none
Jndi Name	Specifies a JNDI name for the reference.	Java character set name	none

Session Descriptor

:

Parameter Name	Default Value	Parameter Value
URLRewritingEnabled	true	Enables URL rewriting, which encodes the session ID into the URL and provides session tracking if cookies are disabled in the browser.
IDLength	52	Sets the size of the session ID. The minimum value is 8 bytes and the maximum value is `Integer.MAX_VALUE`. If you are writing a WAP application, you must use URL rewriting because the WAP protocol does not support cookies. Also, some WAP devices have a 128-character limit on URL length (including parameters), which limits the amount of data that can be transmitted using URL re-writing. To allow more space for parameters, use this parameter to limit the size of the session ID that is randomly generated by WebLogic Server

Parameter Name	Default Value	Parameter Value
CookieComment	Weblogic Server Session Tracking Cookie	Specifies the comment that identifies the session tracking cookie in the cookie file. If unset, this parameter defaults to `WebLogic Session Tracking Cookie`. You may provide a more specific name for your application.
CookieDomain	Null	Identifies the server to which the browser sends cookie information when the browser makes a request. For example, setting the CookieDomain to `.mydomain.com` returns cookies to any server in the `*.mydomain.com` domain. The domain name must have at least two components; setting a name to `*.com` or `*.net` is invalid. If unset, this parameter defaults to the server that issued the cookie.
CookieMaxAgeSecs	-1	Sets the life span of the session cookie, in seconds, after which it expires on the client. If the value is 0, the cookie expires immediately. The maximum value is MAX_VALUE, where the cookie lasts forever. If set to `-1`, the cookie expires when the user exits the browser.
CookieName	`JSESSIONID`	Defines the session cookie name. Defaults to `JSESSIONID` if unset. You may set this to a more specific name for your application.
CookiePath	Null	Specifies the pathname to which the browser sends cookies. If unset, this parameter defaults to `/` (slash), where the browser sends cookies to all URLs served by WebLogic Server. You may set the path to a narrower mapping, to limit the request URLs to which the browser sends cookies.

Parameter Name	Default Value	Parameter Value
InvalidationIntervalSecs	`60`	Sets the time, in seconds, that WebLogic Server waits between doing house-cleaning checks for timed-out and invalid sessions, and deleting the old sessions and freeing up memory. Use this parameter to tune WebLogic Server for best performance on high traffic sites. The minimum value is every second (1). The maximum value is once a week (604,800 seconds). If unset, the parameter defaults to 60 seconds.
JDBConnectionTimeoutSecs	120	Sets the time, in seconds, that WebLogic Server waits before timing out a JDBC connection, where x is the number of seconds between.
PersistenceStoreDir	`session_db`	If you have set `PersistentStoreType` to `file`, this parameter sets the directory path where WebLogic Server will store the sessions. The directory path is either relative to the temp directory or an absolute path. The temp directory is either a generated directory under the `WEB-INF` directory of the Web application, or a directory specified by the context-param `javax.servlet.context.tmpdir`. Ensure that you have enough disk space to store the *number of valid sessions* multiplied by the *size of each session*. You can find the size of a session by looking at the files created in the `PersistenceStoreDir`. You can make file-persistent sessions clusterable by making this directory a shared directory among different servers. You must create this directory manually.
PersistenceStorePool	None	Specifies the name of a JDBC connection pool to be used for persistence storage.

Parameter Name	Default Value	Parameter Value
PersistentStoreType	`memory`	Sets the persistent store method to one of the following options: ■ `memory`—disables persistent session storage ■ `file`—uses file-based persistence (See also `PersistenceStoreDir`, above) ■ `jdbc`—uses a database to store persistent sessions. (see also `PersistenceStorePool`, above) ■ `replicated`—same as `memory`, but session data is replicated across the clustered servers ■ cookie—all session data is stored in a cookie in the user's browser
CookiesEnabled	True	Use of session cookies is enabled by default and is recommended, but you can disable them by setting this property to `false`. You might turn this option off to test using URL rewriting.
TrackingEnabled	True	When set to true, session tracking is enabled.
TimeoutSecs	3600	Sets the time, in seconds, that WebLogic Server waits before timing out a session, where x is the number of seconds between a session's activity. Minimum value is 1, default is 3600, and maximum value is `integer MAX_VALUE`. On busy sites, you can tune your application by adjusting the timeout of sessions. While you want to give a browser client every opportunity to finish a session, you do not want to tie up the server needlessly if the user has left the site or otherwise abandoned the session. This parameter can be overridden by the `session-timeout` element (defined in minutes) in `web.xml`. For more information, see `Session Timeout`.
ConsoleMainAttribute		If you enable Session Monitoring in the WebLogic Server Administration Console, set this parameter to the name of the session parameter you will use to identify each session that is monitored.

Parameter Name	Default Value	Parameter Value
PersistentStoreCookieName	`WLCOOKIE`	Sets the name of the cookie used for cookie-based persistence.

JSP Descriptor

Parameter Name	Default Value	Parameter Value
Compile Command	`javac`, or the Java compiler defined for a server under the configuration /tuning tab of the WebLogic Server Administration Console	Specifies the full pathname of the standard Java compiler used to compile the generated JSP servlets. For faster performance, specify a different compiler, such as IBM's Jikes or Symantec's `sj`.
Compiler Class	`None`	Name of a Java compiler that is executed in WebLogic Servers's virtual machine. (Used in place of an executable compiler such as `javac` or `sj`.)
Compile Flags	`None`	Passes one or more command-line flags to the compiler. Enclose multiple flags in quotes, separated by a space. For example: `java weblogic.jspc -compileFlags "-g -v" myFile.jsp`
Working Dir	`internally generated directory`	The name of a directory where WebLogic Server saves the generated Java and compiled class files for a JSP.

Parameter Name	Default Value	Parameter Value
Verbose	`true`	When set to `true`, debugging information is printed out to the browser, the command prompt, and WebLogic Server log file.
keepgenerated	`false`	Saves the Java files that are generated as an intermediary step in the JSP compilation process. Unless this parameter is set to `true`, the intermediate Java files are deleted after they are compiled.
Page Check Seconds	`1`	Sets the interval, in seconds, at which WebLogic Server checks to see if JSP files have changed and need recompiling. Dependencies are also checked and recursively reloaded if changed. If set to 0, pages are checked on every request. If set to `-1`, page checking and recompiling is disabled.
Encoding	Default encoding of your platform	Specifies the default character set used in the JSP page. Use standard Java character set names (see http://java.sun.com/j2se/1.3/docs/guide/intl/encoding.doc.htm). If unset, this parameter defaults to the encoding for your platform. A JSP page directive (included in the JSP code) overrides this setting. For example: `<%@ page contentType="text/html; charset=custom-encoding”%>`
Package Prefix	`jsp_servlet`	Specifies the package into which all JSP pages are compiled.
No Try Blocks	`false`	If a JSP file has numerous or deeply nested custom JSP tags and you receive a `java.lang.VerifyError` exception when compiling, use this flag to allow the JSPs to compile correctly.
Precompile	`false`	When set to true WebLogic Server automatically compiles all JSPs on startup.
Compiler Supports Encoding	`False`	Specifies the encoding used by the WebLogic JSP compiler to create the intermediate.`java` file. When set to true, the JSP compiler uses the encoding specified with the `contentType` attribute contained in the `page` directive on the JSP page, or, if a `contentType` is not specified, the encoding defined with the `encoding` parameter in the jsp-descriptor. When set to false, the JSP compiler uses the default encoding for the JVM when creating the intermediate `.java` file.

Container Descriptor

Attribute	Description	Range of Values	Default Value
Check Auth on Forward Enabled	When enabled, requires authentication of forwarded requests from a servlet or JSP.	Checked, Unchecked	Unchecked
redirect-with-absoute-url	Controls whether the `javax.servlet.http.HttpServletResponse.SendRedirect()` method redirects using a relative or absolute URL. Set this element to `false` if you are using a proxy HTTP server and do not want the URL converted to a non-relative link. The default behavior is to convert the URL to a non-relative link.	boolean	true

Charset Params

Input Charset Descriptors

Attribute	Description	Range of Values	Default Value
Resource Path	A path which, if included in the URL of a request, signals WebLogic Server to use the Java character set specified by the Java Charset Name field.	String	none
Java Charset Name	Specifies the Java characters set to use.	Java character set name	none

Character Set Mapping

Attribute	Description	Range of Values	Default Value
IANA Charset Name	Specifies the IANA character set name that is to be mapped to the Java character set specified by the Java Charset Name field.	IANA character set name	none
Java Charset Name	Specifies the Java characters set to use.	Java character set name	none

Configuring a WebLogic Web Service

The Web Service information in the Deployment Descriptor Editor is provided for monitoring purposes only; changing any of the values might result in the deployed Web Service not functioning correctly.

Web Services

RPC Services

Attribute	Description	Range of Values	Default Value
JNDI Name	JNDI name of the stateless session EJB that comprises the RPC-style Web service.	String.	None.
Home Interface	The Home interface of the stateless session EJB.	String.	None.
Remote Interface	The Remote interface of the stateless session EJB.	String.	None.
URI	URI used by client applications to invoke the Web service.	String.	None.

Message Services

Attribute	Description	Range of Values	Default Value
Service Name	Name of the SOAP Servlet that handles SOAP messages between WebLogic Server and client applications.	String.	Null.
Destination	JNDI name of the JMS topic or queue which receives or sends data between WebLogic Server and the client application.	String.	None.
Destination Type	The type of JMS destination: topic or queue.	`Topic` or `Queue`.	None.
Action	Specifies whether the client application that invokes this message-style Web service sends or receives data to the JMS destination. Specify `send` if the client sends data to the JMS destination and `receive` if the client receives data from the JMS destination.	`Send` or `Receive`.	None.
Connection Factory	JNDI name of the ConnectionFactory used to create a connection to the JMS destination.	String.	None.

Attribute	Description	Range of Values	Default Value
URI	URI used by clients to invoke the Web service.	String.	None.

115 Web Application Deployment Descriptor Editor Help

The following sections contains information about using the Web Application Deployment Descriptor Editor:

- Overview of the Web Application Deployment Descriptor Editor
- Configuring a Web Application using the Web Application Deployment Descriptor Editor
- Creating a New Web Application
- Using the Web Application Deployment Descriptor Editor
- Editing an Existing Deployment Descriptor
- Instructions for configuring Web Application components (The component name is followed by the top-level XML element used to represent it in the deployment descriptor):
  - Configuring Basic Web Application Attributes (`<icon>, <display-name>, <description>, <distributable>`)
  - Configuring Context Parameters (`<context-param>)`)
  - Configuring Filters (`<filter>, <filter-mapping>`)
  - Configuring Listeners (`<listener>`)
  - Configuring Servlets (`<servlet>`, `<servlet-mapping>`)

  - Configuring Welcome Pages (`<welcome-file-list>`)
  - Configuring the Session Timeout (`<session-config>`)
  - Configuring MIME Mappings (`<mime-mapping>`)
  - Configuring Error Pages (`<error-page>`)
  - Configuring JSP Tag Libraries (`<taglib>`)
  - Configuring Resource Environment References (`<resource-env-ref>`)
  - Configuring Resource References (`<resource-ref>`)
  - Configuring Security Constraints (`<security-constraint>`)
  - Configuring Logins (`<login-config>`)
  - Configuring Security Roles (`<security-role>`)
  - Configuring Environment Entries (`<env-entry>`)
  - Configuring EJB References (`<ejb-ref>`)
- Configuring a New Webapp Ext (`weblogic.xml`)
  - Configuring the Session Descriptor (`<session-descriptor>`)
  - Configuring the JSP Descriptor (`<jsp-descriptor>`)
  - Configuring Security Role Assignments (`<security-role-assignment>`)
  - Configuring a Reference Descriptor (`<reference-descriptor>`)
  - Configuring Character Set Parameters (`<charset-params>`)
  - Configuring a Container Descriptor (`<container-descriptor>`)
- Configuring a WebLogic Web Service

Overview of the Web Application Deployment Descriptor Editor

The Web Application Deployment Descriptor Editor allows you to edit the deployment descriptors that define a Web Application. A Web Application is a J2EE deployment unit that defines a collection of Web resources such as JSPs, servlets, and HTML pages. Web Applications can also define references to external resources such as EJBs.

Each Web Application has two deployment descriptors, `web.xml` and `weblogic.xml`, both of which can be edited using the Web Application Deployment Descriptor Editor. The `web.xml` deployment descriptor is defined by the Servlet specification from Sun Microsystems and the `weblogic.xml` deployment descriptor is specific to WebLogic Server. For more information about Web Applications, see *Assembling and Configuring Web Applications* in the WebLogic Server Documentation at `http://e-docs.bea.com/wls/docs61/webapp/index.html`.

The Web Application Deployment Descriptor Editor can only edit an existing deployment descriptor, it cannot create a new deployment descriptor.

Creating a New Web Application

To create a new Web Application:

1. Create the components of the Web Application: the servlets, JSPs and HTML pages.

2. Arrange the components on your file system into the directory structure defined for Web Applications. For more information, see Directory Structure `http://e-docs.bea.com/wls/docs61/webapp/basics.html#dir`.

3. Create a skeleton deployment descriptor. You can do this one of two ways:

   - Follow the procedures described under Packaging Web Applications at `http://e-docs.bea.com/wls/docs61/programming/packaging.html#pack005`.

- Create a text file called `web.xml` and save it in the `WEB-INF` directory of your Web Application. This file must contain the following text:

  ```
  <!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//
  DTD Web Application 2.3//EN"
  "http://java.sun.com/dtd/web-app_2_3.dtd">

  <web-app>
  </web-app>
  ```

4. Configure the functionality of your Web Application by using the deployment descriptor editor as described in the next section.

Using the Web Application Deployment Descriptor Editor

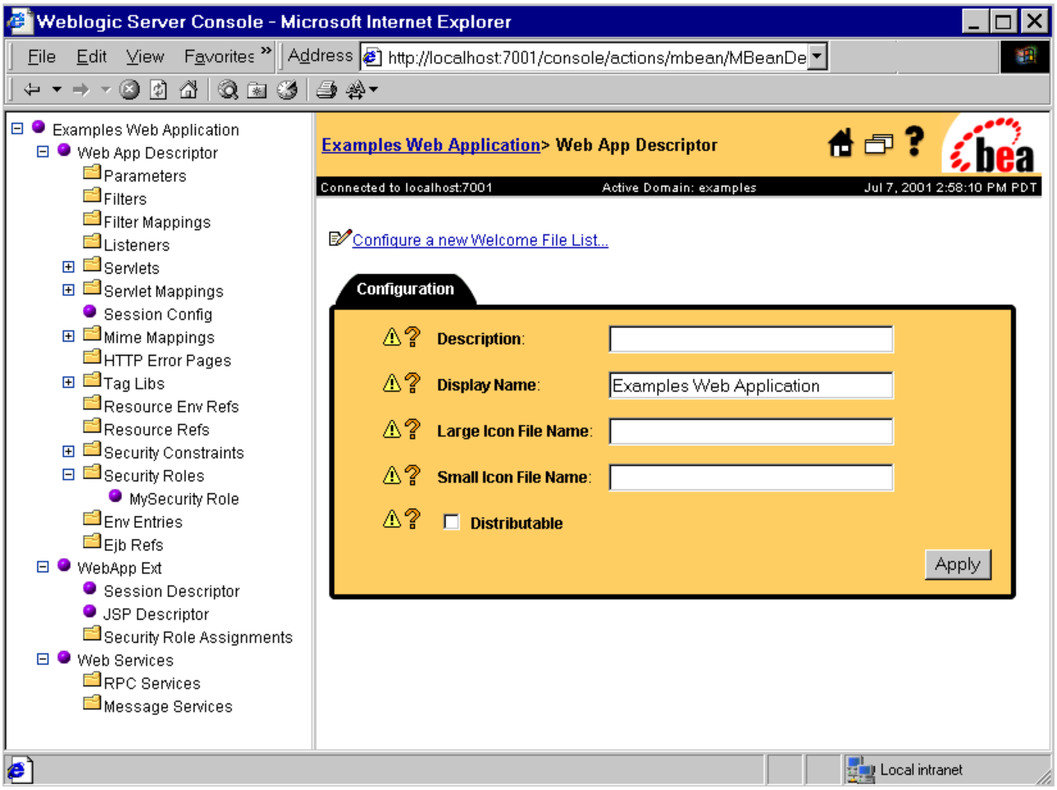
1. Start the WebLogic Administration Server if it is not already running.
2. Start the WebLogic Server Administration Console in a browser.
3. If the Web Application is already defined in the current domain, expand the Web Applications node in the left panel and skip to Step 5.
4. If the Web Application is not yet configured in the current domain, configure a new Web Application:
   a. Click on “Configure a new Web Application” in the right panel.
   b. Enter the name of this Web Application in the Name field.
   c. Enter the name of the directory containing your Web Application in the URI field.
   d. Enter the path to the directory containing your Web Application in the Path field.
   e. Click Create. The new Web Application appears under Web Applications in the left panel
5. Right click on the Web Application and select Edit Web Application Descriptor. A new browser window opens in which you can edit the deployment descriptors.
6. Configure the functionality of your Web Application. See Configuring a Web Application using the Web Application Deployment Descriptor Editor for instructions on configuring Web Application functionality.

7. When you have finished editing the deployment descriptor, select the Web App Descriptor node in the left navigation tree.

8. Click Validate in the right panel. The validation only checks to see that the deployment descriptor conforms to its document type definition (DTD). It does not verify the existence of any of the components you have configured nor perform any other type of validation.

9. After you have validated the deployment descriptors, click Persist in the right panel. Your changes to the deployment descriptor are not written to the deployment descriptor files on disk until you click the Persist button.

Figure 115-1 Web Application Deployment Descriptor Editor



Description:

Display Name: Examples Web Application

Large Icon File Name:

Small Icon File Name:

□ Distributable

Editing an Existing Deployment Descriptor

1. Follow the procedure Using the Web Application Deployment Descriptor Editor
2. Select the deployment descriptor elements you wish to edit from the navigation tree in the left panel. Some elements require you to right click on their parent nodes in the left navigation tree and select the option to create the element. For more information, see Configuring a Web Application using the Web Application Deployment Descriptor Editor
3. Fill in or change the fields in the right panel as required.
4. Click the Create or Apply button as appropriate.
5. When you have finished editing the deployment descriptor, select the top node in the left navigation tree.
6. Click Validate in the right panel. The validation only checks to see that the deployment descriptor conforms to its document type definition (DTD). It does not verify the existence of any of the components you have configured nor perform any other type of validation.
7. Click Persist in the right panel. Your changes to the deployment descriptor are not written to the deployment descriptor files on disk until you click the Persist button.

Configuring a Web Application using the Web Application Deployment Descriptor Editor

Configuring a New Webapp Ext

The WebApp Ext node in the left pane represents the WebLogic-specific deployment descriptor, `weblogic.xml`. The `weblogic.xml` descriptor is used to configure attributes unique to WebLogic Server. To configure attributes in `weblogic.xml`, you must first create its node in the deployment descriptor editor.

To create the `weblogic.xml` descriptor node:

1. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)
2. Select Configure a new WebAppExtDescriptor
3. In the right pane, enter a description of the Web Application in the Description field (optional).
4. Enter the version of WebLogic Server that your application is running on (optional).
5. Click Create.

Configuring the Session Descriptor

The Session Descriptor configures HTTP session-related parameters.

1. If it does not exist in the left pane, create the WebApp Ext node:

   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

   b. Select Configure a new WebAppExtDescriptor

   c. In the right pane, enter a description of the Web Application in the Description field (optional).

   d. Enter the version of WebLogic Server that your application is running on (optional).

   e. Click Create.

   f. Expand the WebApp Ext node in the left panel.

2. Click on the Session Descriptor node in the left panel.
3. Fill in the fields in the right panel according to the information in Session Descriptor table.
4. Click Apply.

Configuring the JSP Descriptor

The JSP Descriptor configures JSP-related parameters.

1. If it does not exist in the left pane, create the WebApp Ext node:

   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

   b. Select Configure a new WebAppExtDescriptor

   c. In the right pane, enter a description of the Web Application in the Description field (optional).

   d. Enter the version of WebLogic Server that your application is running on (optional).

   e. Click Create.

   f. Expand the WebApp Ext node in the left panel.

2. Click on the JSP Descriptor node in the left panel.
3. Fill in the fields in the right panel according to the information in JSP Descriptor table.
4. Click Apply.

Configuring Security Role Assignments

Security role assignments map a security role to one or more principals. The principals must be defined in your security realm.

1. If it does not exist in the left pane, create the WebApp Ext node:
   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)
   b. Select Configure a new WebAppExtDescriptor
   c. In the right pane, enter a description of the Web Application in the Description field (optional).
   d. Enter the version of WebLogic Server that your application is running on (optional).
   e. Click Create.
   f. Expand the WebApp Ext node in the left panel.
2. Right click on the Security Role Assignment node in the left panel and select Configure a new SecurityRoleAssignment.
3. Select a security role from the Role drop-down list.
4. Add one or more Principal Names to the text box, one per line. The principals must be valid in your security realm.
5. Click Create.

Configuring Character Set Parameters

You can define codeset behavior for non-unicode operations by configuring Character Set Parameters. You can also specify a mapping of Java character sets to IANA character sets. For more information, see Determining the Encoding of an HTTP Request at `http://e-docs.bea.com/wls/docs61/webapp/components.html#encoding-http`, and Mapping IANA Character Sets to Java Character Sets at `http://e-docs.bea.com/wls/docs61/webapp/components.html#map-iana`.

1. If it does not exist in the left pane, create the WebApp Ext node:

   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

   b. Select Configure a new WebAppExtDescriptor

   c. In the right pane, enter a description of the Web Application in the Description field (optional).

   d. Enter the version of WebLogic Server that your application is running on (optional).

   e. Click Create.

   f. Expand the WebApp Ext node in the left panel.

2. Right click on the WebApp Ext node in the left panel and select Configure a new Charset Params.

3. Create Input Charset Descriptors

   a. Expand the Charset Params node.

   b. Right click on the Input Charset Descriptors node and select Configure a new InputCharsetDescriptor.

   c. Enter the path that applies to this Charset Descriptor in the Resource Path field.

   d. Enter a valid Java character set name in the Java Charset Name field.

   e. Click Create

4. Create a Character Set Mapping

a. Expand the Charset Params node.

b. Right click on the Charset Mappings node and select Configure a new CharsetMapping.

c. Enter a valid IANA character set name in the IANA Charset Name field.

d. Enter a valid Java character set name in the Java Charset Name field.

e. Click Create

Configuring a Reference Descriptor

The Reference Descriptor maps the JNDI name of a server resource to a name used in the Web application. The Resource Description panel maps a resource, for example, a DataSource, to its JNDI name. The Ejb Reference panel maps an EJB to its JNDI name.

1. If it does not exist in the left pane, create the WebApp Ext node:

   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

   b. Select Configure a new WebAppExtDescriptor

   c. In the right pane, enter a description of the Web Application in the Description field (optional).

   d. Enter the version of WebLogic Server that your application is running on (optional).

   e. Click Create.

   f. Expand the WebApp Ext node in the left panel.

2. Right click on the WebApp Ext node in the left panel and select Configure a new Reference Descriptor.

3. Create Resource Descriptions. You can create one or more Resource Descriptions.

   a. Expand the Resource Descriptor node.

b. Right click on the Resource Descriptions node and select Configure a new ResourceDescription.

c. Enter the name of an object from the JNDI tree in the Jndi Name field.

d. Select a Resource Reference from the Resource Reference drop-down list. The Resource Reference should be already configured under the Resource Refs tab.

e. Click Create

4. Create an EJB Reference Description

a. Expand the Resource Descriptor node.

b. Right click on the Ejb Reference Descriptions node and select Configure a new EjbReferenceDescription.

c. Enter the name of an object from the JNDI tree in the Jndi Name field.

d. Select an EJB Reference from the Ejb Reference drop-down list. The EJB Reference should be already configured under the Ejb refs tab.

e. Click Create

Configuring a Container Descriptor

Enable Check Auth on Forward when you want to require authentication of forwarded requests from a servlet or JSP.

1. If it does not exist in the left pane, create the WebApp Ext node:

a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

b. Select Configure a new WebAppExtDescriptor

c. In the right pane, enter a description of the Web Application in the Description field (optional).

d. Enter the version of WebLogic Server that your application is running on (optional).

e. Click Create.

f. Expand the WebApp Ext node in the left panel.

2. Right click on the WebApp Ext node and select Configure a new Container Descriptor.
3. To enable authentication of forwarded requests, check the Check Auth on Forward Enabled box.
4. To configure redirect behavior, check the redirect-with-absoute-url box.
5. Click Create.

Web App Descriptor

Attribute	Description	Range of Values	Default Value
Description	A text description of the servlet.	String	none
Display Name	This element is not used by WebLogic Server.	String	none
Large Icon File Name	Specifies the location for a large (32x32 pixel) `.gif` or `.jpg` image used to represent the Web application in a GUI tool. Currently, this element is not used by WebLogic Server.	String	none
Small Icon File Name	Specifies the location for a small (16x16 pixel) `.gif` or `.jpg` image used to represent the Web application in a GUI tool. Currently, this is not used by WebLogic Server.	String	none
Distributable	This element is not used by WebLogic Server.	Empty	none

Filters

Attribute	Description	Range of Values	Default Value
Filter Name	The name for this filter	String	MyFilter
Description	A text description of the servlet.	String	none
Display Name	This element is not used by WebLogic Server.	String	none
Small Icon File Name	This element is not used by WebLogic Server.	String	none
Large Icon File Name	This element is not used by WebLogic Server.	String	none
Filter Class	The fully-qualified class name of the filter.	Class name	none

Filter Mappings

Attribute	Description	Range of Values	Default Value
Filter	The name of the filter to which you are mapping a URL pattern. This name corresponds to the name you assigned a filter in a `<filter-name>` element.	String	none

Url Pattern	Describes a pattern used to resolve URLs. The portion of the URL after the `http://host:port` + `WebAppName` is compared to the `<url-pattern>` by WebLogic Server. If the patterns match, the filter mapped in this element will be called.
	Example patterns:
	`/soda/grape/*`
	`/foo/*`
	`/contents`
	`*.foo`
	The URL must follow the rules specified in Section 10 of the Servlet 2.2 Specification.
		String	none

Attribute	Description	Range of Values	Default Value
Servlet	The name of a servlet which, if called, causes this filter to execute.	A valid servlet name defined in this deployment descriptor.	none

Listeners

Attribute	Description	Range of Values	Default Value
Listener Class Name	Name of the class that responds to a Web Application event.	Class name	none

Servlets

Attribute	Description	Range of Values	Default Value
Description	A text description of the servlet.	String	None
Display Name	This element is not used by WebLogic Server.	String	None
Small Icon File Name	This element is not used by WebLogic Server.	String	None
Large Icon File Name	This element is not used by WebLogic Server.	String	None
Servlet Name	Defines the canonical name of the servlet, used to reference the servlet definition elsewhere in the deployment descriptor.	String	MyServlet
Servlet Class	The fully-qualified class name of the servlet. You may use only one of either the `<servlet-class>`tags or `<jsp-file>` tags in your servlet body.	Class name	None

Attribute	Description	Range of Values	Default Value
Jsp File	The full path to a JSP file within the Web application, relative to the Web application root directory. You may use only one of either the `<servlet-class>` tags or `<jsp-file>` tags in your servlet body.	String	None
Load On Startup	WebLogic Server initializes this servlet when WebLogic Server starts up. The optional contents of this element must be a positive integer indicating the order in which the servlet should be loaded. Lower integers are loaded before higher integers.	Positive Integer If no value is specified, or if the value specified is not a positive integer, WebLogic Server can load the servlet in any order in the startup sequence.	0

Parameters

Attribute	Description	Range of Values	Default Value
Description	Text description of the initialization parameter.		MyError Page
Param Name	Defines the name of this initialization parameter.	String	None

Attribute	Description	Range of Values	Default Value
Param Value	Defines a `String` value for this initialization parameter.	String	None

Security Role Refs

Attribute	Description	Range of Values	Default Value
Description	Text description of the role.	String	MyError Page
Role Name	Defines the name of the security role or principal that is used in the servlet code.	String	None
Role Link	Defines the name of the security role.	String - a security role that is defined in a `<security-role>` element later in the deployment descriptor.	None

Servlet Mappings

Attribute	Description	Range of Values	Default Value
Servlet	The name of the servlet to which you are mapping a URL pattern. This name corresponds to the name you assigned a servlet in a `<servlet>` declaration tag.	A declared servlet.	none

URL Pattern	Describes a pattern used to resolve URLs. The portion of the URL after the `http://host:port` + `WebAppName` is compared to the `<url-pattern>` by WebLogic Server. If the patterns match, the servlet mapped in this element will be called.
	Example patterns:
	`/soda/grape/*`
	`/foo/*`
	`/contents`
	`*.foo`
	The URL must follow the rules specified in Section 10 of the Servlet 2.2 Specification.
		String	MyServlet Mapping

Mime Mappings

Attribute	Description	Range of Values	Default Value
Mime Type	A string describing the defined mime type, for example: `text/plain.`	String	MyMime Mapping
Extension	A string describing an extension, for example: `txt.`	String	None

Session Config

Attribute	Description	Range of Values	Default Value
Session Timeout	The number of minutes after which sessions in this Web Application expire. The value set in this element overrides the value set in the `TimeoutSecs` parameter of the Session Descriptor in the WebLogic-specific deployment descriptor `weblogic.xml`, unless one of the special values listed here is entered.	Maximum value: Integer.MAX_VALUE ÷ 60 Special values: ■ -2 = Use the value set by `TimeoutSecs` in Session Descriptor ■ -1 = Sessions do not timeout. The value set in Session Descriptor is ignored.	-2

Welcome Files

Attribute	Description	Range of Values	Default Value
Welcome File	File name to use as a default welcome file, such as index.html. You may specify one or more welcome files.	String	None

Error Pages

Attribute	Description	Range of Values	Default Value
Error Code	A valid HTTP error code, for example `404.`	String	MyError Page
Exception Type	A fully-qualified class name of a Java exception type, for example `java.lang.`	String	None
Location	The location of the resource to display in response to the error. Must include a leading `/`. For example `/myErrorPg.html.`	String	None

Tag Libs

Attribute	Description	Range of Values	Default Value
URI	Describes a URI, relative to the location of the web.xml document, identifying a Tag Library used in the Web application. If the URI matches the URI string used in the taglib directive on the JSP page, this taglib is used.	String	MyTag Lib
Location	Gives the file name of the tag library descriptor relative to the root of the Web application. It is a good idea to store the tag library descriptor file under the `WEB-INF` directory so it is not publicly available over an HTTP request.	String	None

Resource Env Refs

The resource-env-ref element contains a declaration of an component's reference to an administered object associated with a resource in the component's environment.

Attribute	Description	Range of Values	Default Value
Description	A text description.	String	none
Ref Name	The name of a resource environment reference; its value is the environment entry name used in code.	String	MyResource Env Ref
Ref Type	The type of a resource environment reference.	String	none

Resource Refs

Attribute	Description	Range of Values	Default Value
Description	A text description.	String	none
Ref Name	The name of the resource used in the JNDI tree. Servlets in the Web application use this name to look up a reference to the resource.	String	MyResource Ref
Ref Type	The Java type of the resource that corresponds to the reference name. Use the full package name of the Java type.	Java type	none
Auth	Used to control the resource sign on for security.	`APPLICATION:` The application component code performs resource sign on programmatically. `CONTAINER:` WebLogic Server uses the security context established with login config element.	none
Sharing Scope	Specifies whether connections obtained through the given resource manager connection factory reference can be shared.	`Sharable` `Unsharable`	`Sharable`

Security Constraints

Attribute	Description	Range of Values	Default Value
Display Name	Name of this constraint.	String	MySecurity Constraint

Auth Constraint

Defines which groups or principals have access to the collection of web resources defined in this security constraint.

Attribute	Description	Range of Values	Default Value
Description	A text description of this security constraint.	String	none
Role Name	Defines which security roles can access resources defined in this security-constraint. Security role names are mapped to principals using the `Role Name.`	A defined security role.	none

User Data Constraint

Defines how the client should communicate with the server.

Attribute	**Description**	**Range of Values**	**Default Value**
Description	A text description.	String	none
Transport Guarantee	Specifies that the communication between client and server. WebLogic Server establishes a Secure Sockets Layer (SSL) connection when the user is authenticated using the `INTEGRAL` or `CONFIDENTIAL` constraint.s	■ `NONE`—the application does not require any transport guarantees. ■ `INTEGRAL`—the application requires that the data be sent between the client and server in such a way that it cannot be changed in transit. ■ `CONFIDENTIAL`—the application requires that the data be transmitted in a fashion that prevents other entities from observing the contents of the transmission.	none

Web Resource Collection

Defines the components of the Web application to which this security constraint is applied.

Attribute	**Description**	**Range of Values**	**Default Value**
Web Resource Name	The name of this Web resource collection.	String	none
Description	A text description of this security constraint.	String	none

URL Pattern	Use one or more of the URL Pattern elements to declare to which URL patterns this security constraint applies. If you do not use at least one of these elements, this web resource collection is ignored by WebLogic Server.	String	none
HTTP Method	Use one or more of the HTTP Method elements to declare which HTTP methods (`GET | POST |...`) are subject to the authorization constraint. If you omit the HTTP Method element, the default behavior is to apply the security constraint to all HTTP methods.	`GET` `POST`	none

LoginConfig

Attribute	**Description**	**Range of Values**	**Default Value**
Auth Method	Specifies the method used to authenticate the user. Possible values: `BASIC` - uses browser authentication `FORM` - uses a user-written HTML form `CLIENT-CERT`	`BASIC` - uses browser authentication `FORM` - uses a user-written HTML form `CLIENT-CERT`	BASIC

Realm Name	The name of the realm that is referenced to authenticate the user credentials. If omitted, the WebLogic realm is used by default. For more information, see Specifying a Security Realm at http://e-docs.bea.com/wls/docs61/adminguide/cnfgsec.html#cnfgsec004.	String	None
Login Page	The URI of a Web resource relative to the document root, used to authenticate the user. This can be an HTML page, JSP, or HTTP servlet, and must return an HTML page containing a FORM that conforms to a specific naming convention.	String	None
Error Page	The URI of a Web resource relative to the document root, sent to the user in response to a failed authentication login.	String	None

Security Roles

Attribute	Description	Range of Values	Default Value
Description	A text description of this security role.	String	none
Role Name	The role name. The name you use here must have a corresponding entry in the WebLogic-specific deployment descriptor, `weblogic.xml`, which maps roles to principals in the security realm.	String	system

Env Entries

Attribute	Description	Range of Values	Default Value
Description	A textual description.	String	none
Env Entry Name	The name of the environment entry.	String	MyEnvironment Entry
Env Entry Value	The value of the environment entry.	String	none
Env Entry Type	The name of the environment entry.	String	none

Ejb refs

Attribute	Description	Range of Values	Default Value
Description	A text description of the reference.	String	none
EJBRef Name	The name of the EJB used in the Web application. This name is mapped to the JNDI tree in the WebLogic-specific deployment descriptor `weblogic.xml`.	String	MyEJB Ref
EJBRef Type	The expected Java class type of the referenced EJB.	String	none
Home Interface Name	The fully qualified class name of the EJB home interface.	String	none
Remote Interface Name	The fully qualified class name of the EJB remote interface.	String	none
EJBLink Name	The `<ejb-name>` of an EJB in an encompassing J2EE application package.	String	none
Run As	A security role whose security context is applied to the referenced EJB.	A security role defined in this web application.	none

Security Role Assignment

Attribute	Description	Range of Values	Default Value
Role Name	Specifies the name of a security role.	A valid security role.	none
Principal Name	Specifies the name of a principal that is defined in the security realm. You can use multiple `<principal-name>` elements to map principals to a role. For more information on security realms, see the *Programming WebLogic Security* at http://e-docs.bea.com/wls/docs61/security/index.html.	A principal defined in the security realm.	none

Reference Descriptor

Resource Descriptions

Attribute	Description	Range of Values	Default Value
Res Ref Name	Specifies the name of a resource reference.	String	none
Jndi Name	Specifies a JNDI name for the resource.	Java character set name	none

EJB Reference Description

Attribute	Description	Range of Values	Default Value
Ejb Ref Name	Specifies the name of an EJB reference used in your Web application.	String	none
Jndi Name	Specifies a JNDI name for the reference.	Java character set name	none

Session Descriptor

:

Parameter Name	Default Value	Parameter Value
URLRewritingEnabled	true	Enables URL rewriting, which encodes the session ID into the URL and provides session tracking if cookies are disabled in the browser.
IDLength	52	Sets the size of the session ID. The minimum value is 8 bytes and the maximum value is `Integer.MAX_VALUE`. If you are writing a WAP application, you must use URL rewriting because the WAP protocol does not support cookies. Also, some WAP devices have a 128-character limit on URL length (including parameters), which limits the amount of data that can be transmitted using URL re-writing. To allow more space for parameters, use this parameter to limit the size of the session ID that is randomly generated by WebLogic Server

Parameter Name	Default Value	Parameter Value
CookieComment	Weblogic Server Session Tracking Cookie	Specifies the comment that identifies the session tracking cookie in the cookie file. If unset, this parameter defaults to `WebLogic Session Tracking Cookie`. You may provide a more specific name for your application.
CookieDomain	Null	Identifies the server to which the browser sends cookie information when the browser makes a request. For example, setting the CookieDomain to `.mydomain.com` returns cookies to any server in the `*.mydomain.com` domain. The domain name must have at least two components; setting a name to `*.com` or `*.net` is invalid. If unset, this parameter defaults to the server that issued the cookie.
CookieMaxAgeSecs	-1	Sets the life span of the session cookie, in seconds, after which it expires on the client. If the value is 0, the cookie expires immediately. The maximum value is MAX_VALUE, where the cookie lasts forever. If set to `-1`, the cookie expires when the user exits the browser.
CookieName	`JSESSIONID`	Defines the session cookie name. Defaults to `JSESSIONID` if unset. You may set this to a more specific name for your application.
CookiePath	Null	Specifies the pathname to which the browser sends cookies. If unset, this parameter defaults to `/` (slash), where the browser sends cookies to all URLs served by WebLogic Server. You may set the path to a narrower mapping, to limit the request URLs to which the browser sends cookies.

Parameter Name	Default Value	Parameter Value
InvalidationIntervalSecs	`60`	Sets the time, in seconds, that WebLogic Server waits between doing house-cleaning checks for timed-out and invalid sessions, and deleting the old sessions and freeing up memory. Use this parameter to tune WebLogic Server for best performance on high traffic sites. The minimum value is every second (1). The maximum value is once a week (604,800 seconds). If unset, the parameter defaults to 60 seconds.
JDBConnectionTimeoutSecs	120	Sets the time, in seconds, that WebLogic Server waits before timing out a JDBC connection, where x is the number of seconds between.
PersistenceStoreDir	`session_db`	If you have set `PersistentStoreType` to `file`, this parameter sets the directory path where WebLogic Server will store the sessions. The directory path is either relative to the temp directory or an absolute path. The temp directory is either a generated directory under the `WEB-INF` directory of the Web application, or a directory specified by the context-param `javax.servlet.context.tmpdir`. Ensure that you have enough disk space to store the *number of valid sessions* multiplied by the *size of each session*. You can find the size of a session by looking at the files created in the `PersistenceStoreDir`. You can make file-persistent sessions clusterable by making this directory a shared directory among different servers. You must create this directory manually.
PersistenceStorePool	None	Specifies the name of a JDBC connection pool to be used for persistence storage.

Parameter Name	Default Value	Parameter Value
PersistentStoreType	memory	Sets the persistent store method to one of the following options: ■ memory—disables persistent session storage ■ file—uses file-based persistence (See also PersistenceStoreDir, above) ■ jdbc—uses a database to store persistent sessions. (see also PersistenceStorePool, above) ■ replicated—same as memory, but session data is replicated across the clustered servers ■ cookie—all session data is stored in a cookie in the user's browser
CookiesEnabled	True	Use of session cookies is enabled by default and is recommended, but you can disable them by setting this property to false. You might turn this option off to test using URL rewriting.
TrackingEnabled	True	When set to true, session tracking is enabled.
TimeoutSecs	3600	Sets the time, in seconds, that WebLogic Server waits before timing out a session, where x is the number of seconds between a session's activity. Minimum value is 1, default is 3600, and maximum value is integer MAX_VALUE. On busy sites, you can tune your application by adjusting the timeout of sessions. While you want to give a browser client every opportunity to finish a session, you do not want to tie up the server needlessly if the user has left the site or otherwise abandoned the session. This parameter can be overridden by the session-timeout element (defined in minutes) in web.xml. For more information, see Session Timeout.
ConsoleMainAttribute		If you enable Session Monitoring in the WebLogic Server Administration Console, set this parameter to the name of the session parameter you will use to identify each session that is monitored.

Parameter Name	Default Value	Parameter Value
PersistentStoreCookieName	`WLCOOKIE`	Sets the name of the cookie used for cookie-based persistence.

JSP Descriptor

Parameter Name	Default Value	Parameter Value
Compile Command	`javac`, or the Java compiler defined for a server under the configuration /tuning tab of the WebLogic Server Administration Console	Specifies the full pathname of the standard Java compiler used to compile the generated JSP servlets. For faster performance, specify a different compiler, such as IBM's Jikes or Symantec's `sj`.
Compiler Class	`None`	Name of a Java compiler that is executed in WebLogic Servers's virtual machine. (Used in place of an executable compiler such as `javac` or `sj`.)
Compile Flags	`None`	Passes one or more command-line flags to the compiler. Enclose multiple flags in quotes, separated by a space. For example: `java weblogic.jspc -compileFlags "-g -v" myFile.jsp`
Working Dir	`internally generated directory`	The name of a directory where WebLogic Server saves the generated Java and compiled class files for a JSP.

Parameter Name	Default Value	Parameter Value
Verbose	`true`	When set to `true`, debugging information is printed out to the browser, the command prompt, and WebLogic Server log file.
keepgenerated	`false`	Saves the Java files that are generated as an intermediary step in the JSP compilation process. Unless this parameter is set to `true`, the intermediate Java files are deleted after they are compiled.
Page Check Seconds	`1`	Sets the interval, in seconds, at which WebLogic Server checks to see if JSP files have changed and need recompiling. Dependencies are also checked and recursively reloaded if changed. If set to 0, pages are checked on every request. If set to `-1`, page checking and recompiling is disabled.
Encoding	Default encoding of your platform	Specifies the default character set used in the JSP page. Use standard Java character set names (see http://java.sun.com/j2se/1.3/docs/guide/intl/encoding.doc.htm). If unset, this parameter defaults to the encoding for your platform. A JSP page directive (included in the JSP code) overrides this setting. For example: `<%@ page contentType="text/html; charset=custom-encoding”%>`
Package Prefix	`jsp_servlet`	Specifies the package into which all JSP pages are compiled.
No Try Blocks	`false`	If a JSP file has numerous or deeply nested custom JSP tags and you receive a `java.lang.VerifyError` exception when compiling, use this flag to allow the JSPs to compile correctly.
Precompile	`false`	When set to true WebLogic Server automatically compiles all JSPs on startup.
Compiler Supports Encoding	`False`	Specifies the encoding used by the WebLogic JSP compiler to create the intermediate.`java` file. When set to true, the JSP compiler uses the encoding specified with the `contentType` attribute contained in the `page` directive on the JSP page, or, if a `contentType` is not specified, the encoding defined with the `encoding` parameter in the jsp-descriptor. When set to false, the JSP compiler uses the default encoding for the JVM when creating the intermediate `.java` file.

Container Descriptor

Attribute	Description	Range of Values	Default Value
Check Auth on Forward Enabled	When enabled, requires authentication of forwarded requests from a servlet or JSP.	Checked, Unchecked	Unchecked
redirect-with-absoute-url	Controls whether the `javax.servlet.http.HttpServletResponse.SendRedirect()` method redirects using a relative or absolute URL. Set this element to `false` if you are using a proxy HTTP server and do not want the URL converted to a non-relative link. The default behavior is to convert the URL to a non-relative link.	boolean	true

Charset Params

Input Charset Descriptors

Attribute	Description	Range of Values	Default Value
Resource Path	A path which, if included in the URL of a request, signals WebLogic Server to use the Java character set specified by the Java Charset Name field.	String	none
Java Charset Name	Specifies the Java characters set to use.	Java character set name	none

Character Set Mapping

Attribute	Description	Range of Values	Default Value
IANA Charset Name	Specifies the IANA character set name that is to be mapped to the Java character set specified by the Java Charset Name field.	IANA character set name	none
Java Charset Name	Specifies the Java characters set to use.	Java character set name	none

Configuring a WebLogic Web Service

The Web Service information in the Deployment Descriptor Editor is provided for monitoring purposes only; changing any of the values might result in the deployed Web Service not functioning correctly.

Web Services

RPC Services

Attribute	Description	Range of Values	Default Value
JNDI Name	JNDI name of the stateless session EJB that comprises the RPC-style Web service.	String.	None.
Home Interface	The Home interface of the stateless session EJB.	String.	None.
Remote Interface	The Remote interface of the stateless session EJB.	String.	None.
URI	URI used by client applications to invoke the Web service.	String.	None.

Message Services

Attribute	Description	Range of Values	Default Value
Service Name	Name of the SOAP Servlet that handles SOAP messages between WebLogic Server and client applications.	String.	Null.
Destination	JNDI name of the JMS topic or queue which receives or sends data between WebLogic Server and the client application.	String.	None.
Destination Type	The type of JMS destination: topic or queue.	`Topic` or `Queue`.	None.
Action	Specifies whether the client application that invokes this message-style Web service sends or receives data to the JMS destination. Specify `send` if the client sends data to the JMS destination and `receive` if the client receives data from the JMS destination.	`Send` or `Receive`.	None.
Connection Factory	JNDI name of the ConnectionFactory used to create a connection to the JMS destination.	String.	None.

Attribute	Description	Range of Values	Default Value
URI	URI used by clients to invoke the Web service.	String.	None.

116 Web Application Deployment Descriptor Editor Help

The following sections contains information about using the Web Application Deployment Descriptor Editor:

- Overview of the Web Application Deployment Descriptor Editor
- Configuring a Web Application using the Web Application Deployment Descriptor Editor
- Creating a New Web Application
- Using the Web Application Deployment Descriptor Editor
- Editing an Existing Deployment Descriptor
- Instructions for configuring Web Application components (The component name is followed by the top-level XML element used to represent it in the deployment descriptor):
  - Configuring Basic Web Application Attributes (`<icon>, <display-name>, <description>, <distributable>`)
  - Configuring Context Parameters (`<context-param>)`)
  - Configuring Filters (`<filter>, <filter-mapping>`)
  - Configuring Listeners (`<listener>`)
  - Configuring Servlets (`<servlet>`, `<servlet-mapping>`)

    - Configuring Welcome Pages (`<welcome-file-list>`)
    - Configuring the Session Timeout (`<session-config>`)
    - Configuring MIME Mappings (`<mime-mapping>`)
    - Configuring Error Pages (`<error-page>`)
    - Configuring JSP Tag Libraries (`<taglib>`)
    - Configuring Resource Environment References (`<resource-env-ref>`)
    - Configuring Resource References (`<resource-ref>`)
    - Configuring Security Constraints (`<security-constraint>`)
    - Configuring Logins (`<login-config>`)
    - Configuring Security Roles (`<security-role>`)
    - Configuring Environment Entries (`<env-entry>`)
    - Configuring EJB References (`<ejb-ref>`)
- Configuring a New Webapp Ext (`weblogic.xml`)
    - Configuring the Session Descriptor (`<session-descriptor>`)
    - Configuring the JSP Descriptor (`<jsp-descriptor>`)
    - Configuring Security Role Assignments (`<security-role-assignment>`)
    - Configuring a Reference Descriptor (`<reference-descriptor>`)
    - Configuring Character Set Parameters (`<charset-params>`)
    - Configuring a Container Descriptor (`<container-descriptor>`)
- Configuring a WebLogic Web Service

Overview of the Web Application Deployment Descriptor Editor

The Web Application Deployment Descriptor Editor allows you to edit the deployment descriptors that define a Web Application. A Web Application is a J2EE deployment unit that defines a collection of Web resources such as JSPs, servlets, and HTML pages. Web Applications can also define references to external resources such as EJBs.

Each Web Application has two deployment descriptors, `web.xml` and `weblogic.xml`, both of which can be edited using the Web Application Deployment Descriptor Editor. The `web.xml` deployment descriptor is defined by the Servlet specification from Sun Microsystems and the `weblogic.xml` deployment descriptor is specific to WebLogic Server. For more information about Web Applications, see *`Assembling and Configuring Web Applications`* in the WebLogic Server Documentation at `http://e-docs.bea.com/wls/docs61/webapp/index.html`.

The Web Application Deployment Descriptor Editor can only edit an existing deployment descriptor, it cannot create a new deployment descriptor.

Creating a New Web Application

To create a new Web Application:

1. Create the components of the Web Application: the servlets, JSPs and HTML pages.

2. Arrange the components on your file system into the directory structure defined for Web Applications. For more information, see Directory Structure `http://e-docs.bea.com/wls/docs61/webapp/basics.html#dir`.

3. Create a skeleton deployment descriptor. You can do this one of two ways:

   - Follow the procedures described under Packaging Web Applications at `http://e-docs.bea.com/wls/docs61/programming/packaging.html#pack005`.

- Create a text file called `web.xml` and save it in the `WEB-INF` directory of your Web Application. This file must contain the following text:

  ```
  <!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//
  DTD Web Application 2.3//EN"
  "http://java.sun.com/dtd/web-app_2_3.dtd">

  <web-app>
  </web-app>
  ```

4. Configure the functionality of your Web Application by using the deployment descriptor editor as described in the next section.

Using the Web Application Deployment Descriptor Editor

1. Start the WebLogic Administration Server if it is not already running.
2. Start the WebLogic Server Administration Console in a browser.
3. If the Web Application is already defined in the current domain, expand the Web Applications node in the left panel and skip to Step 5.
4. If the Web Application is not yet configured in the current domain, configure a new Web Application:

   a. Click on “Configure a new Web Application” in the right panel.

   b. Enter the name of this Web Application in the Name field.

   c. Enter the name of the directory containing your Web Application in the URI field.

   d. Enter the path to the directory containing your Web Application in the Path field.

   e. Click Create. The new Web Application appears under Web Applications in the left panel

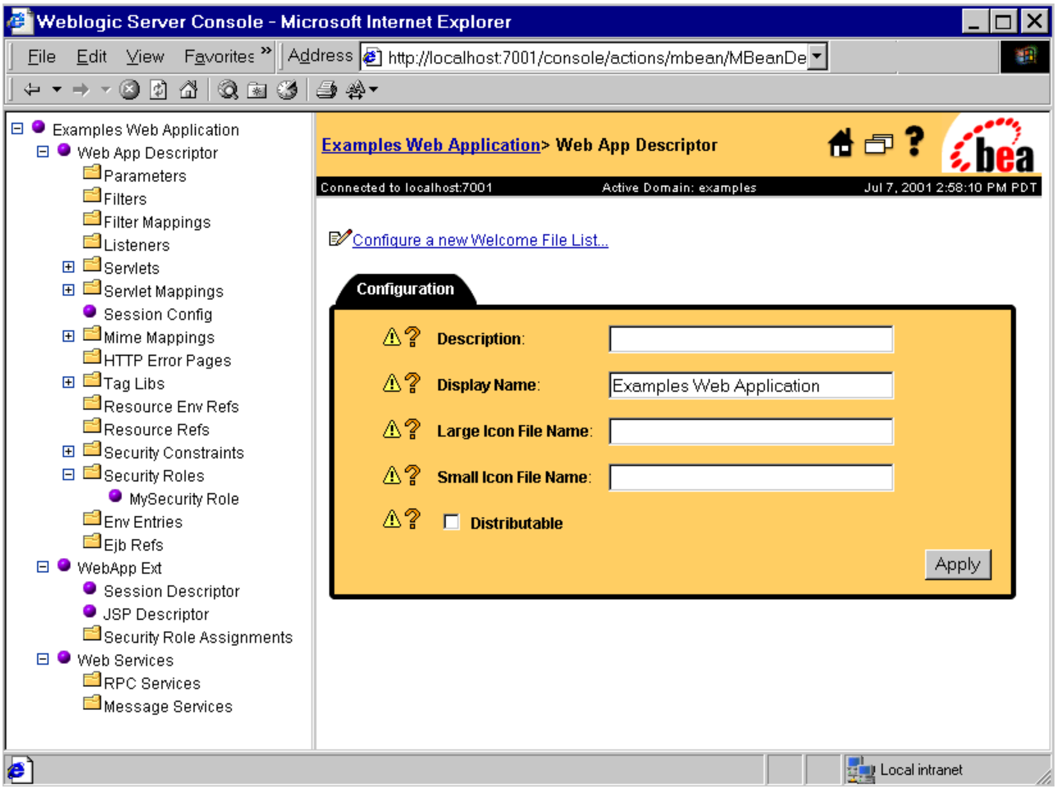
5. Right click on the Web Application and select Edit Web Application Descriptor. A new browser window opens in which you can edit the deployment descriptors.
6. Configure the functionality of your Web Application. See Configuring a Web Application using the Web Application Deployment Descriptor Editor for instructions on configuring Web Application functionality.

7. When you have finished editing the deployment descriptor, select the Web App Descriptor node in the left navigation tree.

8. Click Validate in the right panel. The validation only checks to see that the deployment descriptor conforms to its document type definition (DTD). It does not verify the existence of any of the components you have configured nor perform any other type of validation.

9. After you have validated the deployment descriptors, click Persist in the right panel. Your changes to the deployment descriptor are not written to the deployment descriptor files on disk until you click the Persist button.

Figure 116-1 Web Application Deployment Descriptor Editor



Description:

Display Name: Examples Web Application

Large Icon File Name:

Small Icon File Name:

☐ Distributable

Editing an Existing Deployment Descriptor

1. Follow the procedure Using the Web Application Deployment Descriptor Editor
2. Select the deployment descriptor elements you wish to edit from the navigation tree in the left panel. Some elements require you to right click on their parent nodes in the left navigation tree and select the option to create the element. For more information, see Configuring a Web Application using the Web Application Deployment Descriptor Editor
3. Fill in or change the fields in the right panel as required.
4. Click the Create or Apply button as appropriate.
5. When you have finished editing the deployment descriptor, select the top node in the left navigation tree.
6. Click Validate in the right panel. The validation only checks to see that the deployment descriptor conforms to its document type definition (DTD). It does not verify the existence of any of the components you have configured nor perform any other type of validation.
7. Click Persist in the right panel. Your changes to the deployment descriptor are not written to the deployment descriptor files on disk until you click the Persist button.

Configuring a Web Application using the Web Application Deployment Descriptor Editor

The procedures in this section describe how to create the deployment information required for a Web Application. You should have created a basic deployment descriptor and have that Web Application open in the Web Application Deployment Descriptor Editor as described in Creating a New Web Application and Using the Web Application Deployment Descriptor Editor.

The sections that follow provide procedures for configuring the components of a Web Application. Tables of configurable attributes follow the procedures and are also linked from the appropriate procedures.

Configuring Basic Web Application Attributes

1. Click on the Web App Descriptor node in the left panel
2. Configure the following elements on the Web App Descriptor panel as required:
    - Description
    - Display Name
    - Large Icon File Name
    - Small Icon File Name
    - Distributable

Configuring Context Parameters

Context parameters are used to pass name/value pairs to the servlet container. You can enter any number of context parameter name/value pairs.

1. Right click on the Parameters node in the left panel and select Configure a new Parameter.
2. Enter a description for the Parameter in the Description field. (optional)
3. Enter the name of the parameter in the Param Name field.
4. Enter the value of the parameter in the Param Value field.
5. Click Create

You can access these parameters in your code using the `javax.servlet.ServletContext.getInitParameter()` and `javax.servlet.ServletContext.getInitParameterNames()` methods.

Configuring Filters

Filters intercept requests for Web Application resources to perform additional functions. To configure a filter, you create a filter and then map it to a servlet or URL pattern. You can configure any number of filters or filter mappings. For more information, see Filters at `http://e-docs.bea.com/wls/docs61/webapp/filters.html`.

Filters are a part of the proposed final draft of the Servlet 2.3 specification from Sun Microsystems. If you are planning to use filters in your application, note that the specification has not been finalized and could change in the future.

1. Right click on the Filters node in the left panel and select Configure a new Filter.
2. Fill in the following fields:
   - Filter Name (required) The name of this filter.
   - Description (optional)
   - Display Name (optional)
   - Small Icon File Name (optional)
   - Large Icon File Name (optional)
   - Filter Class (required) The full class name of the Java class that executes the filter. For example, `myApp.filters.myFilter`.
3. Click Create
4. Right click on the Filter Mappings node in the left panel and select Configure a new FilterMapping. You can map a filter to a URL pattern or a servlet that is already configured in this Web Application. You may create more than one mapping for each filter.
5. Select the filter you are mapping from the Filter drop-down list.
6. If you are mapping this filter to a URL pattern, enter it in the Url Pattern field.
7. If you are mapping this filter to a servlet, select a servlet from the Servlet drop-down list. You must configure a servlet before you can map it to a filter.
8. Click Create

9. Create Filter initialization parameters. Filter initialization parameters can be read by your filter class using the `FilterConfig.getInitParameter()` or `FilterConfig.getInitParameters()` methods.

    a. Expand the Filters node in the left panel.

    b. Expand the node of the filter for which you want to create initialization parameters.

    c. Right click on the Parameters node and select Configure a new Parameter.

    d. Enter a description for the Parameter in the Description field (optional).

    e. Enter the name of the parameter in the Param Name field.

    f. Enter the value of the parameter in the Param Value field.

    g. Click Create.

10. Repeat these steps to create additional filter mappings.

Configuring Listeners

Listeners are Java classes that respond to HTTP session or servlet context events. You can create any number of listeners. For more information, see Application Events and Listeners at `http://e-docs.bea.com/wls/docs61/webapp/app_events.html`.

Application events and listeners are a part of the proposed final draft of the Servlet 2.3 specification from Sun Microsystems. If you are planning to use application events and listeners in your application, note that the specification has not been finalized and could change in the future.

1. Right click on the Listeners node in the left panel and select Configure a new Listener

2. Enter the full name of the listener class in the Listener Class Name field.

3. Click Create.

Configuring Servlets

To configure a servlet, define the servlet and then map the servlet to a URL pattern. You can create any number of servlets. For more information, see Configuring Servlets `http://e-docs.bea.com/wls/docs61/webapp/components.html#configuring-servlets`.

1. Right click on the Servlet node in the left panel and select Configure a new Servlet.
2. Fill in the following fields:
   - Description (optional)
   - Display Name (optional)
   - Small Icon File Name (optional)
   - Large Icon File Name (optional)
   - Servlet Name (required)
   - Servlet Class. The Java class that executes this servlet.
     or
     Jsp File. A JSP file that is executed this servlet.
   - Load On Startup. (optional)
3. Click Create.
4. Right Click on Servlet Mappings and select Configure a new ServletMapping.
5. Select the servlet you want to map from the Servlet drop-down list.
6. Enter a URL pattern for this servlet. For more information about URL patterns, See Servlet Mapping at http://e-docs.bea.com/wls/docs61/webapp/components.html#servlet-mapping.
7. Click Create
8. Create Servlet initialization parameters. (optional).
   a. Expand the Servlet node in the left panel.
   b. Expand the node of the servlet for which you want to create initialization parameters.

c. Right click on the Parameters node in the left panel and select Configure a new Parameter.

d. Enter a description for the Parameter in the Description field (optional).

e. Enter the name of the parameter in the Param Name field.

f. Enter the value of the parameter in the Param Value field.

g. Click Create.

To retrieve initialization parameters, call the `getInitParameter(String name)` method from the parent `javax.servlet.GenericServlet` class. When passed the name of the parameter, this method returns the parameter’s value as a `String`.

9. Create Servlet Security Role Refs.

   a. Expand the Servlet node in the left panel.

   b. Expand the node of the servlet for which you want to create Security Role Refs.

   c. Right click on the Security Role Refs node and select Configure a new SecurityRoleRef.

   d. Enter a description for the Security Role Ref in the Description field (optional).

   e. Select a security role from the Role Link drop-down list. The security role must be previously defined on the Security Roles panel.

   f. Enter the role name used in your servlet code in the Role Name field.

Configuring Welcome Pages

When the HTTP request is for a directory name, WebLogic Server serves the first file specified in this list. If that file is not found, the server then tries the next file in the list. For more information, see Configuring Welcome Pages at `http://e-docs.bea.com/wls/docs61/webapp/components.html#welcome_pages`.

1. Right click on the Web App Descriptor node in the left panel and select Configure a new WelcomeFileList.

2. Enter Welcome Files names, one per line.

3. Click Create.

Configuring the Session Timeout

Session Timeout set the time after which HTTP sessions in this Web Application expire.

1. Select the Session Config node in the left panel.
2. Edit the Session Timeout value.
3. Click Apply.

Configuring MIME Mappings

MIME mappings defines a mapping between a file extension and a MIME type. You can create any number of MIME mappings.

1. Right click on MIME Mappings in the left panel and select Configure a new MIME Mapping.
2. Enter a valid MIME type in the Mime Type field.
3. Enter the file extension you want to map in the Extension field.
4. Click Create

Configuring Error Pages

An error page is a JSP or HTML page that you configure to be displayed in response to an HTTP error code or Java exception. You can configure different error pages to respond to various HTTP error codes or Java exceptions. For more information, see Customizing HTTP Error Responses at `http://e-docs.bea.com/wls/docs61/webapp/components.html#error-page`.

1. Right Click on Error Pages in the left panel and select Configure a new ErrorPage.

2. Enter an HTTP Error Code in the Error Code field
   or
   Enter a Java Exception class in the Exception Type field.

3. Enter the name of the resource to display in response to the error. The entry must include a leading `/`. For example `/myErrorPg.html`.

4. Click Create.

Configuring JSP Tag Libraries

JSP tag libraries contain the Java classes and descriptors that define a user-written JSP tag. You can create any number of tag libraries. You can configure one or more JSP tag libraries. For more information, see *Programming WebLogic JSP Tag Extensions* at `http://e-docs.bea.com/wls/docs61/taglib/index.html`.

1. Right click on the Tag Libs node in the left panel and select Configure a new TagLib.

2. Enter the name (relative to the `WEB-INF` directory of the Web Application) you will use to refer to this tag library in the JSP `taglib` directive in the URI field.

3. Enter the location (relative to the root directory of the Web Application) of the tag library or tag library jar file in the Location field.

4. Click Create.

Configuring Resource Environment References

Resource environment references are a part of the proposed final draft of the Servlet 2.3 specification from Sun Microsystems and are not currently implemented in WebLogic Server.

1. Right click on the Resource Env Refs node in the left panel and select Configure a new ResourceEnvRef.

2. Enter a description for this Resource Env-Ref in the Description (optional).

3. Enter the name of this Resource Env-Ref in the Ref Name field.

4. Enter the Java type for this Resource Env-Ref in the Ref Type field.
5. Click Create

Configuring Resource References

A resource reference defines a lookup name for an external resource. Your servlet code can look up a resource by this "virtual" name that is mapped to the actual location at deployment time. You can create any number of resource references.

1. Right click on the Resource Refs node in the left panel and select Configure a new ResourceRef.
2. Enter a description for this Resource Ref in the Description field (optional).
3. Enter the name of this Resource Ref in the Ref Name field.
4. Enter the Java type of the Resource Ref in the Ref Type field. Enter the full package name of the Java type.
5. Enter the authorization type in the Auth field. Valid values are `APPLICATION` or `CONTAINER.`
6. Enter the sharing scope in the Sharing Scope field.Valid values are `Sharable` and `Unsharable`.
7. Click Create.

Configuring Security Constraints

Security constraints apply security to specified resources. You can create any number of security constraints. For more information, see Restricting Access to Resources in a Web Application at `http://e-docs.bea.com/wls/docs61/webapp/security.html#resources-webapp`.

1. Right click on the Security Constraints node in the left panel and select Configure a new SecurityConstraint.
2. Enter a name for this Security Constraint in the Display Name field.

3. Click Create.

4. Expand the Security Constraint node in the left panel

5. Expand the node for the Security Constraint you are adding.

6. Create a Web Resource Collection. A Web Resource Collection defines the URL pattern and HTTP method that is constrained by this security constraint.

   a. Right click on Web Resource Collection and select Configure a new WebResourceCollection.

   b. Enter a name for this Web Resource Collection in the Web Resource Name field.

   c. Enter a description for this Web Resource Collection in the Description field. (optional)

   d. Enter one or more URL patterns, one per line, in the URL Pattern field. The URL pattern defines which resources are subject to this security constraint.

   e. Enter a HTTP method in the HTTP Method box. This entry is usually `GET` or `POST`.

   f. Click Create.

7. Create an Auth Constraint

   a. Right click on the name of the Security Constraint you are editing and select Configure a new AuthConstraint.

   b. Enter a description of this auth constraint in the Description field. (optional)

   c. Click Create. A list of previously defined security roles appears in the Available panel.

   d. Use the arrow buttons to move security roles from the available panel to the chosen panel. Note: You must first Create one or more security roles before you can assign them here. For more information, see Configuring Security Roles.

   e. Click Apply.

8. Create a User Data Constraint

   a. Right click on the name of the Security Constraint you are editing and select Configure a new UserDataConstraint.

b. Enter a description of this User Data Constraint in the Description field. (optional)

c. Select the transport guarantee from the Transport Guarantee drop-down list. Valid values are `NONE`, `INTEGRAL`, or `CONFIDENTIAL`.

d. Click Create.

Configuring Logins

Login configuration describes how the user is authenticated, the realm name that should be used for this application, and the attributes that are needed by the form login mechanism.

If this element is present, the user must be authenticated in order to access any resource that is constrained by Security Constraints defined in the Web application. Once authenticated, the user can be authorized to access other resources with access privileges.

1. Right click on the Web App Descriptor node and select Configure a new LoginConfig.
2. Select the Authorization method from the Auth Method drop-down list. Valid values are `BASIC`, `FORM`, or `CLIENT-CERT`.
3. Fill in the name of the authentication realm in the Realm Name field.
4. If you set the Auth Method to `FORM`, enter the URI of the file containing the authentication form in the Login Page field.
5. If you set the Auth Method to `FORM`, enter the URI of the file containing the page a user is redirected to when authentication fails in the Error Page field.

Configuring Security Roles

A Security Role defines a security context for principals (usually a user name) assigned to the role. You can create any number of security roles. For more information, see Restricting Access to Resources in a Web Application at `http://e-docs.bea.com/wls/docs61/webapp/security.html#resources-webapp`.

1. Right click on the Security Roles node in the left panel and select Configure a new SecurityRole.
2. Enter a description for this Security Role in the Description field.(optional).
3. Enter the name of this security role in the Role Name field.
4. Click Create.

Configuring Environment Entries

Declares an environment entry for an application. You can create any number of environment entries.

1. Right click on the Env Entries node in the left panel and select Configure a new EnvEntry.
2. Enter a description for this Environment entry in the Description field.(optional).
3. Enter the name of this Env entry in the Env Entry Name field.
4. Enter the value of this Env Entry in the Env Entry Value field
5. Enter the type of this Env Entry in the Env Entry Type field.

Configuring EJB References

The ejb-ref defines a reference to an EJB resource. This reference is mapped to the actual location of the EJB at deployment time by defining the mapping in the WebLogic-specific deployment descriptor file, `weblogic.xml`.You can create any number of EJB references.

1. Right click on the Ejb refs node in the left panel and select Configure a new EjbRef.
2. Enter a description for this EJB Reference in the Description field.(optional)
3. Enter the name of this EJB Reference in the EJBRef Name field.
4. Enter the Java class of the referenced EJB in the EJBRef Type field.
5. Enter the name of the home interface of the EJB in the Home Interface Name field.
6. Enter the name of the remote interface of the EJB in the Remote Interface Name field.
7. If the EJB is packaged in an Enterprise Application Archive (`EAR`), enter the name of the EJB as it is referenced in the `<ejb-name>` element of the `EAR's` deployment descriptor in the EJBLink Name field.
8. Enter the name of a security role defined in this web application in the Run As field. (optional)

Configuring a New Webapp Ext

The WebApp Ext node in the left pane represents the WebLogic-specific deployment descriptor, `weblogic.xml`. The `weblogic.xml` descriptor is used to configure attributes unique to WebLogic Server. To configure attributes in `weblogic.xml`, you must first create its node in the deployment descriptor editor.

To create the `weblogic.xml` descriptor node:

1. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)
2. Select Configure a new WebAppExtDescriptor
3. In the right pane, enter a description of the Web Application in the Description field (optional).
4. Enter the version of WebLogic Server that your application is running on (optional).

5. Click Create.

Configuring the Session Descriptor

The Session Descriptor configures HTTP session-related parameters.

1. If it does not exist in the left pane, create the WebApp Ext node:
   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)
   b. Select Configure a new WebAppExtDescriptor
   c. In the right pane, enter a description of the Web Application in the Description field (optional).
   d. Enter the version of WebLogic Server that your application is running on (optional).
   e. Click Create.
   f. Expand the WebApp Ext node in the left panel.
2. Click on the Session Descriptor node in the left panel.
3. Fill in the fields in the right panel according to the information in Session Descriptor table.
4. Click Apply.

Configuring the JSP Descriptor

The JSP Descriptor configures JSP-related parameters.

1. If it does not exist in the left pane, create the WebApp Ext node:
   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

   b. Select Configure a new WebAppExtDescriptor

   c. In the right pane, enter a description of the Web Application in the Description field (optional).

   d. Enter the version of WebLogic Server that your application is running on (optional).

   e. Click Create.

   f. Expand the WebApp Ext node in the left panel.

2. Click on the JSP Descriptor node in the left panel.

3. Fill in the fields in the right panel according to the information in JSP Descriptor table.

4. Click Apply.

Configuring Security Role Assignments

Security role assignments map a security role to one or more principals. The principals must be defined in your security realm.

1. If it does not exist in the left pane, create the WebApp Ext node:

   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

   b. Select Configure a new WebAppExtDescriptor

   c. In the right pane, enter a description of the Web Application in the Description field (optional).

   d. Enter the version of WebLogic Server that your application is running on (optional).

   e. Click Create.

   f. Expand the WebApp Ext node in the left panel.

2. Right click on the Security Role Assignment node in the left panel and select Configure a new SecurityRoleAssignment.

3. Select a security role from the Role drop-down list.
4. Add one or more Principal Names to the text box, one per line. The principals must be valid in your security realm.
5. Click Create.

Configuring Character Set Parameters

You can define codeset behavior for non-unicode operations by configuring Character Set Parameters. You can also specify a mapping of Java character sets to IANA character sets. For more information, see Determining the Encoding of an HTTP Request at `http://e-docs.bea.com/wls/docs61/webapp/components.html#encoding-http`, and Mapping IANA Character Sets to Java Character Sets at `http://e-docs.bea.com/wls/docs61/webapp/components.html#map-iana`.

1. If it does not exist in the left pane, create the WebApp Ext node:
   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)
   b. Select Configure a new WebAppExtDescriptor
   c. In the right pane, enter a description of the Web Application in the Description field (optional).
   d. Enter the version of WebLogic Server that your application is running on (optional).
   e. Click Create.
   f. Expand the WebApp Ext node in the left panel.
2. Right click on the WebApp Ext node in the left panel and select Configure a new Charset Params.
3. Create Input Charset Descriptors
   a. Expand the Charset Params node.

b. Right click on the Input Charset Descriptors node and select Configure a new InputCharsetDescriptor.

c. Enter the path that applies to this Charset Descriptor in the Resource Path field.

d. Enter a valid Java character set name in the Java Charset Name field.

e. Click Create

4. Create a Character Set Mapping

a. Expand the Charset Params node.

b. Right click on the Charset Mappings node and select Configure a new CharsetMapping.

c. Enter a valid IANA character set name in the IANA Charset Name field.

d. Enter a valid Java character set name in the Java Charset Name field.

e. Click Create

Configuring a Reference Descriptor

The Reference Descriptor maps the JNDI name of a server resource to a name used in the Web application. The Resource Description panel maps a resource, for example, a DataSource, to its JNDI name. The Ejb Reference panel maps an EJB to its JNDI name.

1. If it does not exist in the left pane, create the WebApp Ext node:

a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

b. Select Configure a new WebAppExtDescriptor

c. In the right pane, enter a description of the Web Application in the Description field (optional).

d. Enter the version of WebLogic Server that your application is running on (optional).

e. Click Create.

   f. Expand the WebApp Ext node in the left panel.

2. Right click on the WebApp Ext node in the left panel and select Configure a new Reference Descriptor.

3. Create Resource Descriptions. You can create one or more Resource Descriptions.

   a. Expand the Resource Descriptor node.

   b. Right click on the Resource Descriptions node and select Configure a new ResourceDescription.

   c. Enter the name of an object from the JNDI tree in the Jndi Name field.

   d. Select a Resource Reference from the Resource Reference drop-down list. The Resource Reference should be already configured under the Resource Refs tab.

   e. Click Create

4. Create an EJB Reference Description

   a. Expand the Resource Descriptor node.

   b. Right click on the Ejb Reference Descriptions node and select Configure a new EjbReferenceDescription.

   c. Enter the name of an object from the JNDI tree in the Jndi Name field.

   d. Select an EJB Reference from the Ejb Reference drop-down list. The EJB Reference should be already configured under the Ejb refs tab.

   e. Click Create

Configuring a Container Descriptor

Enable Check Auth on Forward when you want to require authentication of forwarded requests from a servlet or JSP.

1. If it does not exist in the left pane, create the WebApp Ext node:

   a. Right click on the top level node in the left pane. (The node is labeled *myWebApp* Web Application, where *myWebApp* is the name of the Web Application you are editing.)

   b. Select Configure a new WebAppExtDescriptor

   c. In the right pane, enter a description of the Web Application in the Description field (optional).

   d. Enter the version of WebLogic Server that your application is running on (optional).

   e. Click Create.

   f. Expand the WebApp Ext node in the left panel.

2. Right click on the WebApp Ext node and select Configure a new Container Descriptor.

3. To enable authentication of forwarded requests, check the Check Auth on Forward Enabled box.

4. To configure redirect behavior, check the redirect-with-absoute-url box.

5. Click Create.

Web App Descriptor

Attribute	**Description**	**Range of Values**	**Default Value**
Description	A text description of the servlet.	String	none
Display Name	This element is not used by WebLogic Server.	String	none
Large Icon File Name	Specifies the location for a large (32x32 pixel) `.gif` or `.jpg` image used to represent the Web application in a GUI tool. Currently, this element is not used by WebLogic Server.	String	none
Small Icon File Name	Specifies the location for a small (16x16 pixel) `.gif` or `.jpg` image used to represent the Web application in a GUI tool. Currently, this is not used by WebLogic Server.	String	none
Distributable	This element is not used by WebLogic Server.	Empty	none

Filters

Attribute	Description	Range of Values	Default Value
Filter Name	The name for this filter	String	MyFilter
Description	A text description of the servlet.	String	none
Display Name	This element is not used by WebLogic Server.	String	none
Small Icon File Name	This element is not used by WebLogic Server.	String	none
Large Icon File Name	This element is not used by WebLogic Server.	String	none
Filter Class	The fully-qualified class name of the filter.	Class name	none

Filter Mappings

Attribute	Description	Range of Values	Default Value
Filter	The name of the filter to which you are mapping a URL pattern. This name corresponds to the name you assigned a filter in a `<filter-name>` element.	String	none

Url Pattern	Describes a pattern used to resolve URLs. The portion of the URL after the `http://host:port` + `WebAppName` is compared to the `<url-pattern>` by WebLogic Server. If the patterns match, the filter mapped in this element will be called.
	Example patterns:
	`/soda/grape/*`
	`/foo/*`
	`/contents`
	`*.foo`
	The URL must follow the rules specified in Section 10 of the Servlet 2.2 Specification.
		String	none

Attribute	Description	Range of Values	Default Value
Servlet	The name of a servlet which, if called, causes this filter to execute.	A valid servlet name defined in this deployment descriptor.	none

Listeners

Attribute	Description	Range of Values	Default Value
Listener Class Name	Name of the class that responds to a Web Application event.	Class name	none

Servlets

Attribute	**Description**	**Range of Values**	**Default Value**
Description	A text description of the servlet.	String	None
Display Name	This element is not used by WebLogic Server.	String	None
Small Icon File Name	This element is not used by WebLogic Server.	String	None
Large Icon File Name	This element is not used by WebLogic Server.	String	None
Servlet Name	Defines the canonical name of the servlet, used to reference the servlet definition elsewhere in the deployment descriptor.	String	MyServlet
Servlet Class	The fully-qualified class name of the servlet. You may use only one of either the `<servlet-class>`tags or `<jsp-file>` tags in your servlet body.	Class name	None

Attribute	Description	Range of Values	Default Value
Jsp File	The full path to a JSP file within the Web application, relative to the Web application root directory. You may use only one of either the `<servlet-class>`tags or `<jsp-file>` tags in your servlet body.	String	None
Load On Startup	WebLogic Server initializes this servlet when WebLogic Server starts up. The optional contents of this element must be a positive integer indicating the order in which the servlet should be loaded. Lower integers are loaded before higher integers.	Positive Integer If no value is specified, or if the value specified is not a positive integer, WebLogic Server can load the servlet in any order in the startup sequence.	0

Parameters

Attribute	Description	Range of Values	Default Value
Description	Text description of the initialization parameter.		MyError Page
Param Name	Defines the name of this initialization parameter.	String	None

Attribute	Description	Range of Values	Default Value
Param Value	Defines a `String` value for this initialization parameter.	String	None

Security Role Refs

Attribute	Description	Range of Values	Default Value
Description	Text description of the role.	String	MyError Page
Role Name	Defines the name of the security role or principal that is used in the servlet code.	String	None
Role Link	Defines the name of the security role.	String - a security role that is defined in a `<security-role>` element later in the deployment descriptor.	None

Servlet Mappings

Attribute	Description	Range of Values	Default Value
Servlet	The name of the servlet to which you are mapping a URL pattern. This name corresponds to the name you assigned a servlet in a `<servlet>` declaration tag.	A declared servlet.	none

URL Pattern	Describes a pattern used to resolve URLs. The portion of the URL after the `http://host:port` + `WebAppName` is compared to the `<url-pattern>` by WebLogic Server. If the patterns match, the servlet mapped in this element will be called.
	Example patterns:
	`/soda/grape/*`
	`/foo/*`
	`/contents`
	`*.foo`
	The URL must follow the rules specified in Section 10 of the Servlet 2.2 Specification.
		String	MyServlet Mapping

Mime Mappings

Attribute	Description	Range of Values	Default Value
Mime Type	A string describing the defined mime type, for example: `text/plain.`	String	MyMime Mapping
Extension	A string describing an extension, for example: `txt.`	String	None

Session Config

Attribute	Description	Range of Values	Default Value
Session Timeout	The number of minutes after which sessions in this Web Application expire. The value set in this element overrides the value set in the `TimeoutSecs` parameter of the Session Descriptor in the WebLogic-specific deployment descriptor `weblogic.xml`, unless one of the special values listed here is entered.	Maximum value: Integer.MAX_VALUE ÷ 60 Special values: ■ -2 = Use the value set by `TimeoutSecs` in Session Descriptor ■ -1 = Sessions do not timeout. The value set in Session Descriptor is ignored.	-2

Welcome Files

Attribute	Description	Range of Values	Default Value
Welcome File	File name to use as a default welcome file, such as index.html. You may specify one or more welcome files.	String	None

Error Pages

Attribute	Description	Range of Values	Default Value
Error Code	A valid HTTP error code, for example `404.`	String	MyError Page
Exception Type	A fully-qualified class name of a Java exception type, for example `java.lang.`	String	None
Location	The location of the resource to display in response to the error. Must include a leading `/`. For example `/myErrorPg.html.`	String	None

Tag Libs

Attribute	Description	Range of Values	Default Value
URI	Describes a URI, relative to the location of the web.xml document, identifying a Tag Library used in the Web application. If the URI matches the URI string used in the taglib directive on the JSP page, this taglib is used.	String	MyTag Lib
Location	Gives the file name of the tag library descriptor relative to the root of the Web application. It is a good idea to store the tag library descriptor file under the `WEB-INF` directory so it is not publicly available over an HTTP request.	String	None

Resource Env Refs

The resource-env-ref element contains a declaration of an component's reference to an administered object associated with a resource in the component's environment.

Attribute	Description	Range of Values	Default Value
Description	A text description.	String	none
Ref Name	The name of a resource environment reference; its value is the environment entry name used in code.	String	MyResource Env Ref
Ref Type	The type of a resource environment reference.	String	none

Resource Refs

Attribute	**Description**	**Range of Values**	**Default Value**
Description	A text description.	String	none
Ref Name	The name of the resource used in the JNDI tree. Servlets in the Web application use this name to look up a reference to the resource.	String	MyResource Ref
Ref Type	The Java type of the resource that corresponds to the reference name. Use the full package name of the Java type.	Java type	none
Auth	Used to control the resource sign on for security.	`APPLICATION:` The application component code performs resource sign on programmatically. `CONTAINER:` WebLogic Server uses the security context established with login config element.	none
Sharing Scope	Specifies whether connections obtained through the given resource manager connection factory reference can be shared.	`Sharable` `Unsharable`	`Sharable`

Security Constraints

Attribute	Description	Range of Values	Default Value
Display Name	Name of this constraint.	String	MySecurity Constraint

Auth Constraint

Defines which groups or principals have access to the collection of web resources defined in this security constraint.

Attribute	Description	Range of Values	Default Value
Description	A text description of this security constraint.	String	none
Role Name	Defines which security roles can access resources defined in this security-constraint. Security role names are mapped to principals using the `Role Name.`	A defined security role.	none

User Data Constraint

Defines how the client should communicate with the server.

Attribute	Description	Range of Values	Default Value
Description	A text description.	String	none
Transport Guarantee	Specifies that the communication between client and server. WebLogic Server establishes a Secure Sockets Layer (SSL) connection when the user is authenticated using the `INTEGRAL` or `CONFIDENTIAL` constraint.s	■ `NONE`—the application does not require any transport guarantees. ■ `INTEGRAL`—the application requires that the data be sent between the client and server in such a way that it cannot be changed in transit. ■ `CONFIDENTIAL`—the application requires that the data be transmitted in a fashion that prevents other entities from observing the contents of the transmission.	none

Web Resource Collection

Defines the components of the Web application to which this security constraint is applied.

Attribute	Description	Range of Values	Default Value
Web Resource Name	The name of this Web resource collection.	String	none
Description	A text description of this security constraint.	String	none

URL Pattern	Use one or more of the URL Pattern elements to declare to which URL patterns this security constraint applies. If you do not use at least one of these elements, this web resource collection is ignored by WebLogic Server.	String	none
HTTP Method	Use one or more of the HTTP Method elements to declare which HTTP methods (`GET | POST |...`) are subject to the authorization constraint. If you omit the HTTP Method element, the default behavior is to apply the security constraint to all HTTP methods.	`GET` `POST`	none

LoginConfig

Attribute	**Description**	**Range of Values**	**Default Value**
Auth Method	Specifies the method used to authenticate the user. Possible values: `BASIC` - uses browser authentication `FORM` - uses a user-written HTML form `CLIENT-CERT`	`BASIC` - uses browser authentication `FORM` - uses a user-written HTML form `CLIENT-CERT`	BASIC

Realm Name	The name of the realm that is referenced to authenticate the user credentials. If omitted, the WebLogic realm is used by default. For more information, see Specifying a Security Realm at http://e-docs.bea.com/wls/docs61/adminguide/cnfgsec.html#cnfgsec004.	String	None
Login Page	The URI of a Web resource relative to the document root, used to authenticate the user. This can be an HTML page, JSP, or HTTP servlet, and must return an HTML page containing a FORM that conforms to a specific naming convention.	String	None
Error Page	The URI of a Web resource relative to the document root, sent to the user in response to a failed authentication login.	String	None

Security Roles

Attribute	Description	Range of Values	Default Value
Description	A text description of this security role.	String	none
Role Name	The role name. The name you use here must have a corresponding entry in the WebLogic-specific deployment descriptor, `weblogic.xml`, which maps roles to principals in the security realm.	String	system

Env Entries

Attribute	**Description**	**Range of Values**	**Default Value**
Description	A textual description.	String	none
Env Entry Name	The name of the environment entry.	String	MyEnvironment Entry
Env Entry Value	The value of the environment entry.	String	none
Env Entry Type	The name of the environment entry.	String	none

Ejb refs

Attribute	Description	Range of Values	Default Value
Description	A text description of the reference.	String	none
EJBRef Name	The name of the EJB used in the Web application. This name is mapped to the JNDI tree in the WebLogic-specific deployment descriptor `weblogic.xml`.	String	MyEJB Ref
EJBRef Type	The expected Java class type of the referenced EJB.	String	none
Home Interface Name	The fully qualified class name of the EJB home interface.	String	none
Remote Interface Name	The fully qualified class name of the EJB remote interface.	String	none
EJBLink Name	The `<ejb-name>` of an EJB in an encompassing J2EE application package.	String	none
Run As	A security role whose security context is applied to the referenced EJB.	A security role defined in this web application.	none

Security Role Assignment

Attribute	Description	Range of Values	Default Value
Role Name	Specifies the name of a security role.	A valid security role.	none
Principal Name	Specifies the name of a principal that is defined in the security realm. You can use multiple `<principal-name>` elements to map principals to a role. For more information on security realms, see the *Programming WebLogic Security* at http://e-docs.bea.com/wls/docs61/security/index.html.	A principal defined in the security realm.	none

Reference Descriptor

Resource Descriptions

Attribute	Description	Range of Values	Default Value
Res Ref Name	Specifies the name of a resource reference.	String	none
Jndi Name	Specifies a JNDI name for the resource.	Java character set name	none

EJB Reference Description

Attribute	Description	Range of Values	Default Value
Ejb Ref Name	Specifies the name of an EJB reference used in your Web application.	String	none
Jndi Name	Specifies a JNDI name for the reference.	Java character set name	none

Session Descriptor

:

Parameter Name	Default Value	Parameter Value
URLRewritingEnabled	true	Enables URL rewriting, which encodes the session ID into the URL and provides session tracking if cookies are disabled in the browser.
IDLength	52	Sets the size of the session ID. The minimum value is 8 bytes and the maximum value is `Integer.MAX_VALUE`. If you are writing a WAP application, you must use URL rewriting because the WAP protocol does not support cookies. Also, some WAP devices have a 128-character limit on URL length (including parameters), which limits the amount of data that can be transmitted using URL re-writing. To allow more space for parameters, use this parameter to limit the size of the session ID that is randomly generated by WebLogic Server

Parameter Name	Default Value	Parameter Value
CookieComment	Weblogic Server Session Tracking Cookie	Specifies the comment that identifies the session tracking cookie in the cookie file. If unset, this parameter defaults to `WebLogic Session Tracking Cookie`. You may provide a more specific name for your application.
CookieDomain	Null	Identifies the server to which the browser sends cookie information when the browser makes a request. For example, setting the CookieDomain to `.mydomain.com` returns cookies to any server in the `*.mydomain.com` domain. The domain name must have at least two components; setting a name to `*.com` or `*.net` is invalid. If unset, this parameter defaults to the server that issued the cookie.
CookieMaxAgeSecs	-1	Sets the life span of the session cookie, in seconds, after which it expires on the client. If the value is 0, the cookie expires immediately. The maximum value is MAX_VALUE, where the cookie lasts forever. If set to `-1`, the cookie expires when the user exits the browser.
CookieName	`JSESSIONID`	Defines the session cookie name. Defaults to `JSESSIONID` if unset. You may set this to a more specific name for your application.
CookiePath	Null	Specifies the pathname to which the browser sends cookies. If unset, this parameter defaults to `/` (slash), where the browser sends cookies to all URLs served by WebLogic Server. You may set the path to a narrower mapping, to limit the request URLs to which the browser sends cookies.

Parameter Name	Default Value	Parameter Value
InvalidationIntervalSecs	`60`	Sets the time, in seconds, that WebLogic Server waits between doing house-cleaning checks for timed-out and invalid sessions, and deleting the old sessions and freeing up memory. Use this parameter to tune WebLogic Server for best performance on high traffic sites. The minimum value is every second (1). The maximum value is once a week (604,800 seconds). If unset, the parameter defaults to 60 seconds.
JDBConnectionTimeoutSecs	120	Sets the time, in seconds, that WebLogic Server waits before timing out a JDBC connection, where x is the number of seconds between.
PersistenceStoreDir	`session_db`	If you have set `PersistentStoreType` to `file`, this parameter sets the directory path where WebLogic Server will store the sessions. The directory path is either relative to the temp directory or an absolute path. The temp directory is either a generated directory under the `WEB-INF` directory of the Web application, or a directory specified by the context-param `javax.servlet.context.tmpdir`. Ensure that you have enough disk space to store the *number of valid sessions* multiplied by the *size of each session*. You can find the size of a session by looking at the files created in the `PersistenceStoreDir`. You can make file-persistent sessions clusterable by making this directory a shared directory among different servers. You must create this directory manually.
PersistenceStorePool	None	Specifies the name of a JDBC connection pool to be used for persistence storage.

Parameter Name	Default Value	Parameter Value
PersistentStoreType	`memory`	Sets the persistent store method to one of the following options: ■ `memory`—disables persistent session storage ■ `file`—uses file-based persistence (See also `PersistenceStoreDir`, above) ■ `jdbc`—uses a database to store persistent sessions. (see also `PersistenceStorePool`, above) ■ `replicated`—same as `memory`, but session data is replicated across the clustered servers ■ cookie—all session data is stored in a cookie in the user's browser
CookiesEnabled	True	Use of session cookies is enabled by default and is recommended, but you can disable them by setting this property to `false`. You might turn this option off to test using URL rewriting.
TrackingEnabled	True	When set to true, session tracking is enabled.
TimeoutSecs	3600	Sets the time, in seconds, that WebLogic Server waits before timing out a session, where x is the number of seconds between a session's activity. Minimum value is 1, default is 3600, and maximum value is `integer MAX_VALUE`. On busy sites, you can tune your application by adjusting the timeout of sessions. While you want to give a browser client every opportunity to finish a session, you do not want to tie up the server needlessly if the user has left the site or otherwise abandoned the session. This parameter can be overridden by the `session-timeout` element (defined in minutes) in `web.xml`. For more information, see `Session Timeout`.
ConsoleMainAttribute		If you enable Session Monitoring in the WebLogic Server Administration Console, set this parameter to the name of the session parameter you will use to identify each session that is monitored.

Parameter Name	Default Value	Parameter Value
PersistentStoreCookieName	`WLCOOKIE`	Sets the name of the cookie used for cookie-based persistence.

JSP Descriptor

Parameter Name	Default Value	Parameter Value
Compile Command	`javac`, or the Java compiler defined for a server under the configuration /tuning tab of the WebLogic Server Administration Console	Specifies the full pathname of the standard Java compiler used to compile the generated JSP servlets. For faster performance, specify a different compiler, such as IBM's Jikes or Symantec's `sj`.
Compiler Class	`None`	Name of a Java compiler that is executed in WebLogic Servers's virtual machine. (Used in place of an executable compiler such as `javac` or `sj`.)
Compile Flags	`None`	Passes one or more command-line flags to the compiler. Enclose multiple flags in quotes, separated by a space. For example: `java weblogic.jspc -compileFlags "-g -v" myFile.jsp`
Working Dir	`internally generated directory`	The name of a directory where WebLogic Server saves the generated Java and compiled class files for a JSP.

Parameter Name	Default Value	Parameter Value
Verbose	`true`	When set to `true`, debugging information is printed out to the browser, the command prompt, and WebLogic Server log file.
keepgenerated	`false`	Saves the Java files that are generated as an intermediary step in the JSP compilation process. Unless this parameter is set to `true`, the intermediate Java files are deleted after they are compiled.
Page Check Seconds	`1`	Sets the interval, in seconds, at which WebLogic Server checks to see if JSP files have changed and need recompiling. Dependencies are also checked and recursively reloaded if changed. If set to 0, pages are checked on every request. If set to `-1`, page checking and recompiling is disabled.
Encoding	Default encoding of your platform	Specifies the default character set used in the JSP page. Use standard Java character set names (see http://java.sun.com/j2se/1.3/docs/guide/intl/encoding.doc.htm). If unset, this parameter defaults to the encoding for your platform. A JSP page directive (included in the JSP code) overrides this setting. For example: `<%@ page contentType="text/html; charset=custom-encoding”%>`
Package Prefix	`jsp_servlet`	Specifies the package into which all JSP pages are compiled.
No Try Blocks	`false`	If a JSP file has numerous or deeply nested custom JSP tags and you receive a `java.lang.VerifyError` exception when compiling, use this flag to allow the JSPs to compile correctly.
Precompile	`false`	When set to true WebLogic Server automatically compiles all JSPs on startup.
Compiler Supports Encoding	`False`	Specifies the encoding used by the WebLogic JSP compiler to create the intermediate.`java` file. When set to true, the JSP compiler uses the encoding specified with the `contentType` attribute contained in the `page` directive on the JSP page, or, if a `contentType` is not specified, the encoding defined with the `encoding` parameter in the jsp-descriptor. When set to false, the JSP compiler uses the default encoding for the JVM when creating the intermediate `.java` file.

Container Descriptor

Attribute	Description	Range of Values	Default Value
Check Auth on Forward Enabled	When enabled, requires authentication of forwarded requests from a servlet or JSP.	Checked, Unchecked	Unchecked
redirect-with-absoute-url	Controls whether the `javax.servlet.http.HttpServletResponse.SendRedirect()` method redirects using a relative or absolute URL. Set this element to `false` if you are using a proxy HTTP server and do not want the URL converted to a non-relative link. The default behavior is to convert the URL to a non-relative link.	boolean	true

Charset Params

Input Charset Descriptors

Attribute	Description	Range of Values	Default Value
Resource Path	A path which, if included in the URL of a request, signals WebLogic Server to use the Java character set specified by the Java Charset Name field.	String	none
Java Charset Name	Specifies the Java characters set to use.	Java character set name	none

Character Set Mapping

Attribute	Description	Range of Values	Default Value
IANA Charset Name	Specifies the IANA character set name that is to be mapped to the Java character set specified by the Java Charset Name field.	IANA character set name	none
Java Charset Name	Specifies the Java characters set to use.	Java character set name	none

Configuring a WebLogic Web Service

The Web Service information in the Deployment Descriptor Editor is provided for monitoring purposes only; changing any of the values might result in the deployed Web Service not functioning correctly.

Web Services

RPC Services

Attribute	Description	Range of Values	Default Value
JNDI Name	JNDI name of the stateless session EJB that comprises the RPC-style Web service.	String.	None.
Home Interface	The Home interface of the stateless session EJB.	String.	None.
Remote Interface	The Remote interface of the stateless session EJB.	String.	None.
URI	URI used by client applications to invoke the Web service.	String.	None.

Message Services

Attribute	**Description**	**Range of Values**	**Default Value**
Service Name	Name of the SOAP Servlet that handles SOAP messages between WebLogic Server and client applications.	String.	Null.
Destination	JNDI name of the JMS topic or queue which receives or sends data between WebLogic Server and the client application.	String.	None.
Destination Type	The type of JMS destination: topic or queue.	`Topic` or `Queue`.	None.
Action	Specifies whether the client application that invokes this message-style Web service sends or receives data to the JMS destination. Specify `send` if the client sends data to the JMS destination and `receive` if the client receives data from the JMS destination.	`Send` or `Receive`.	None.
Connection Factory	JNDI name of the ConnectionFactory used to create a connection to the JMS destination.	String.	None.

Attribute	Description	Range of Values	Default Value
URI	URI used by clients to invoke the Web service.	String.	None.

117 WLEC Connection Pool Runtime

By default, this pane allows the user to sort the objects by the following criteria:

- Name
- Domain
- Primary Addresses
- Failover Addresses
- Minimum Pool Size
- Maximum Pool Size

118 WLEC Connection Pool

These procedures describe how to use the Administration Console to set the attributes for configuring and managing WLEC connection pools. For more information, see Using WebLogic Enterprise Connectivity.

Configure a New WLEC Connection Pool

1. Click the WLEC node in the left pane. The WLEC Connection Pools table displays in the right pane showing all the WLEC connection pools defined in the domain.
2. Click the Configure a New WLEC Connection Pool text link. A dialog displays in the right pane showing the tabs associated with configuring a new WLEC connection pool.
3. Enter values in the Name, Primary Address, Failover Address, Domain, Minimum Pool Size, and Maximum Pool Size attribute fields.
4. Click the Create button in the lower right corner to create a WLEC Connection Pool instance with the name you specified in the Name field. The new instance is added under the WLEC node in the left pane.
5. Click Apply.
6. Click the Security tab and change the attribute fields or accept the default values as assigned.
7. Click Apply to save any changes you made.

Clone a WLEC Connection Pool

1. Click the WLEC node in the left pane. The WLEC Connection Pools table displays in the right pane showing all the WLEC connection pools defined in the domain.
2. Click the Clone icon in the row of the WLEC connection pool you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a WLEC connection pool.
3. Enter values in the Name, Primary Address, Failover Address, Domain, Minimum Pool Size, and Maximum Pool Size attribute fields.
4. Click the Clone button in the lower right corner to create a WLEC Connection Pool instance with the name you specified in the Name field. The new instance is added under the WLEC node in the left pane.
5. Click Apply.
6. Click the Security tab and change the attribute fields or accept the default values as assigned.
7. Click Apply to save any changes you made.

Delete a WLEC Connection Pool

1. Click the WLEC node in the left pane. The WLEC Connection Pools table displays in the right pane showing all the WLEC connection pools defined in the domain.
2. Click the Delete icon in the row of the WLEC connection pool you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click Yes to delete the WLEC connection pool. The WLEC connection pool icon under the WLEC node is deleted.

Monitor All Instances of a WLEC Connection Pool

1. Click the WLEC node in the left pane. The WLEC Connection Pools table displays in the right pane showing all the WLEC connection pools defined in the domain.
2. Click the Monitor All Instances icon in the row of the WLEC connection pool you want to monitor. A dialog displays in the right pane showing all instances of the WLEC connection pool deployed across the server domain.

Assign a WLEC Connection Pool

1. Click the instance node in the left pane under WLEC Connection Pools for the pool you want to assign. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Targets tab.
3. Complete the following steps for the Servers and Clusters tabs:
   a. Select one or more targets in the Available column that you want to assign to the data source.
   b. Click the mover control to move the targets you selected to the Chosen column.
   c. Click Apply to save your assignments.

Configuration

These tables describe the attributes you use in the Administration Console to configure and manage WLEC connection pools.For more information, see Using WebLogic Enterprise Connectivity.

General

Attribute	Description	Range of Values	Default Value
Name	This attribute returns the name of the WLEC connection pool. The name must be unique for each WLEC connection pool.	The name can be up to 256 alphanumeric characters, but may not contain commas or spaces.	Null
Primary Addresses	This attribute returns the list of addresses for IIOP Listener/Handlers that can be used to establish a connection between the WLEC connection pool and the WLE domain. The format of each address is `//`*`hostname`*`:`*`port`*. The addresses must match the ISL addresses defined in the `UBBCONFIG` file. Multiple addresses are seperated by commas. For example: `//main1.com:1024, //main2.com:1044.` To configure the WLEC connection pool to use the SSL protocol, use the `corbalocs` prefix with the address of the IIOP Listener/Handler. For example: `corbalocs:`*`//hostname`*`:`*`port`*.	String	Null

Attribute	Description	Range of Values	Default Value
Failover Addresses	This attribute returns the list of addresses for IIOP Listener/Handlers that are used if connections cannot be established with the addresses defined in the Primary Addresses field. Multiple addresses are separated by commas.	String	Null
WLEC Domain	This attribute returns the name of the WLEC domain to which the pool is connected. You can have only one WLEC connection pool per WLE domain. The domain name must match the `domainid` parameter in the `RESOURCES` section of the `UBBCONFIG` file for the WLE domain.	String	Null
Minimum Pool Size	This attribute returns the number of IIOP connections to be added to the WLEC connection pool when WebLogic Server starts.	Integer	1
Maximum Pool Size	This attribute returns the maximum number of IIOP connections that can be made from the WLEC connection pool.	Integer	1

Security

Attribute	Description	Range of Values	Default Value
User Name	This attribute returns the name of a qualified user. This field is required only when the security level in the WLE domain is `USER_AUTH`, `ACL` or `MANDATORY_ACL`.	Valid user name as assigned by the administrator.	Null
User Password	This attribute returns the password of the qualified user specified in the User Name field. This field is required only when you define the `User Name` field.	Valid user password as assigned by the administrator.	Null
User Role	This attribute sets the user role for this connection pool. This field is required when the security level in the WLE domain is `APP_PW`, `USER_AUTH`, `ACL`, or `MANDATORY_ACL.`	Valid user role as defined by the administrator or the developer.	Null

Attribute	Description	Range of Values	Default Value
Application Password	This attribute returns the password for the application. This field is required when the security level in the WLE domain is `APP_PW`, `USER_AUTH`, `ACL`, or `MANDATORY_ACL`.	String	Null
Minimum Encryption Level	This attribute sets the minimum SSL encryption level used between the WLE domain and WebLogic Server. Zero (0) indicates that the data is signed but not sealed. 40, 56, and 128 specify the length, in bits, of the encryption key. If this minimum level of encryption is not met, the SSL connection between WLE and WebLogic Server fails.	The possible values are 0, 40, 56, and 128.	40

Attribute	Description	Range of Values	Default Value
Maximum Encryption Level	This attribute sets the maximum SSL encryption level used between the WLE domain and WebLogic Server. Zero (0) indicates that the data is signed but not sealed. 40, 56, and 128 specify the length, in bits, of the encryption key. If this minimum level of encryption is not met, the SSL connection between WLE and WebLogic Server fails.	The possible values are 0, 40, 56, and 128.	The default is the maximum level allowed by the Encryption Package kit license.
Enable Certificate Authentication	This attribute enables the use of certificate authentication. When you use certificate authentication, WLEC uses the values for the `User Name` and `Application Password` fields to create a certificate for WLEC. If you do not use certificate authentication, WLEC uses password authentication or no authentication, depending on the security level of the WLE domain. If password authentication is required, WLEC uses the values for the `User Name` and `User Password` fields to authenticate.	Boolean Selected = enabled or Not Selected = not enabled	Not Selected

Attribute	Description	Range of Values	Default Value
Enable Security Context	This attribute determines the state of the security context of the WebLogic Server User passed to the WLE domain.	Boolean Selected = enabled or Not Selected = not enabled	Not Selected

Targets

Servers

Attribute	Description	Range of Values	Default Value
Servers	This attribute provides a list from which the assigned servers may be chosen.	List	Null

Clusters

Attribute	Description	Range of Values	Default Value
Clusters	This attribute provides a list from which the assigned clusters may be chosen.	List	Null

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for user supplied information.	The value must be an alphanumeric string.	Null

For more information, see Using WebLogic Enterprise Connectivity.

119 WLEC Connection

By default, this pane allows the user to sort the objects by the following criteria:

- Name
- Domain
- Primary Addresses
- Failover Addresses
- Minimum Pool Size
- Maximum Pool Size

120 XML Entity Spec Registry Entry

Configuration

Attribute	Description	Range of Values	Default Value
Public Id	This attribute is used to define the public ID of either the entity to be resolved or the document type definition (DTD) file to be used to parse the document.		MYXMLEntrySpecRegistryEntry
System Id	This attribute is used to define the system ID of either the entity to be resolved or the document type definition (DTD) file to be used to parse the document		Null

Attribute	Description	Range of Values	Default Value
EntityURI	This attribute is used to define the path of the file containing the external entity. The path is relative to the registry's entity directory.		Null
When To Cache	Specifies whether you want WebLogic Server to cache an external entity referenced by a URL the first time the entity is referenced in an XML document, when WebLogic Server starts, or to use the global caching setting.	`cache-on-reference, cache-at-initialization, defer-to-registry-setting`	defer-to-registry-setting
Cache Timeout Interval	Number of seconds after which external entities in the cache become stale after they have been cached by WebLogic Server.	Integer	-1

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides space for optional user-supplied information.	This attribute value must be an alphanumeric string.	Null

121 XML Registry

Create an XML Registry

1. Click the XML Registries node in the left pane. The XML Registries table displays in the right pane showing all the XML registries defined in the domain.
2. Click the Configure a New XML Registry text link. A dialog displays in the right pane showing the tabs associated with configuring a new registry.
3. Enter values in the Name, Documentation Builder, and SAX Parser Factory attribute fields.
4. Click Create to create an XML-registry instance with the name you specified in the Name field. The new registry is added under the XML Registries node in the left pane.

Clone an XML Registry

1. Click the XML Registries node in the left pane. The XML Registries table displays in the right pane showing all the XML registries defined in the domain.
2. Click the Clone icon in the row of the registry you want to clone. A dialog displays in the right pane showing the tabs associated with cloning a new registry.
3. Enter values in the Name, Documentation Builder, and SAX Parser Factory attribute fields.

4. Click Create to create an XML-registry instance with the name you specified in the Name field. The new registry is added under the XML Registries node in the left pane.

Delete an XML Registry

1. Click the XML Registries node in the left pane. The XML Registries table displays in the right pane showing all the XML registries defined in the domain.
2. Click the Delete icon in the row of the registry you want to delete. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click Yes to delete the registry. The registry icon under the XML Registries node is deleted.

Assign an XML Registry

1. Click the instance node in the left pane under XML Registries for the registry you want to assign. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Targets tab.
3. Select one or more targets in the Available column that you want to assign to the registry.
4. Click the mover control to move the targets you selected to the Chosen column.
5. Click Apply to save your assignments.

Configuration

Attribute	Description	Range of Values	Default Value
Name	This attribute allows the user to set the name of the registry.	The value must be an alphanumeric string.	MyXML Registry
DocumentBuilderFactory	This attribute allows the user to set the default Document Builder Factory for the server. This attribute determines the default factory for all server applications that use DOM to parse documents. To override the default factory for a particular document type definition (DTD), use the Parser Name Class attribute in the XML Registry Entries dialog Configuration tab.	String	weblogic.apache.xerces.jaxp.DocumentBuilderFactoryImpl

Attribute	Description	Range of Values	Default Value
SAX Parser Factory	This attribute allows the user to set the default SAX Parser Factory for the server. This attribute determines the default factory for all server applications that use SAX to parse documents. To override the default factory for a particular document type definition (DTD) or to use a custom-generated SAX parser, use the Parser Name Class attribute on the XML Registry Entries dialog Configuration tab.	String	weblogic.apache.xerces.jaxp.SAXParserFactoryImpl
Transformer Factory	The TransformerFactory classes that implement the JAXP interfaces for transforming XML documents.	String	`weblogic.apache.xalan.processor.TransformerFactoryImpl`
When To Cache	Specifies whether you want WebLogic Server to cache an external entity referenced by a URL the first time the entity is referenced in an XML document or when WebLogic Server starts.	cache-on-reference, cache-at-initialization	`cache-on-reference`

Targets

Attribute	Description	Range of Values	Default Value
Servers	Select from a drop-down list of available server instances. To select multiple instances, hold the control key down while selecting the servers.	List	None Selected

Notes

Attribute	Description	Range of Values	Default Value
Notes	This attribute provides a space for user supplied information.	The value must be an alphanumeric string.	Null

122 XML Registry Entry

Configuration

Attribute	Description	Range of Values	Default Value
Public Id	This attribute is used to define the public ID of either the entity to be resolved or the document type definition (DTD) file to be used to parse the document.		Null
System Id	This attribute is used to define the system ID of either the entity to be resolved or the document type definition (DTD) file to be used to parse the document		Null
Root Element Tag	This attribute is used to define the root of the document to be parsed.		

Attribute	Description	Range of Values	Default Value
Entity Path	This attribute is used to define the path of the file containing the external entity. The path is relative to the registry's entity directory.		Null
Parser Class Name	This attribute is used to define a DOM parser or a SAX parser to be used to parse a document using the DTD defined in the Entity Path attribute. If you are using a custom-generated parser, this attribute is used to define the generated parser. For a particular DTD (or, in the case of a custom-generated parser, a particular application), this attribute overrides the default DOM or SAX parser that is specified using the DocumentBuiderFactory and SAX Parser Factory attributes in the XML Registry dialog Configuration tab.	Valid class name	Null

Notes

Attribut e	Description	Range of Values	Default Value
Notes	This attribute provides space for optional user-supplied information.	This attribute value must be an alphanumeric string.	Null