
Common System Administration Tasks	2
Console --> Version	1
Console --> Preferences	1
Display Deployment Descriptor	1
Customizing a Table View	1
Using the Administration Console	1
Overview	1
Uses of the System Administration Console	2
System Administration Architecture	2
Starting and Logging Out of the Administration Console	3
Starting the Administration Console	3
Logging Out of the Administration Console	6
Configuring Your Domain Using the Administration Console	7
Navigating Through the Administration Console	7
Creating New Configuration Objects	10
Deleting Configuration Objects	10
Cloning Configuration Objects	11
Editing Configuration Objects	12
Monitoring Using the Administration Console	12
Changing the Monitoring Interval	13
Changing the Monitoring Graph Polling Interval	13
Customizing the Administration Console	14
Changing the Default Language of the Administration Console	14
Changing Administration Console Behavior	15
Customizing Table Views	15
Viewing the WebLogic Server Version Number	16
Viewing Server and Browser Information	16
Getting Help	17
Navigating Through the Documentation	17
Help for Configuration Attributes	18
Additional BEA Documentation Available on the Internet	18
Attributes and Console Screen Reference for Administration Console	1
Application --> Configuration --> General	1
Application > Configuration --> Descriptor	1
Application --> Deploy	1

Application --> Notes 1
Enterprise Application Deployment Assistant --> Step 2 - Select Targets 1
Application --> Targets 1
Enterprise Application Deployment Assistant --> Step 3 - Review Choices and Deploy 1
Enterprise Application Deployment Assistant --> Step 1 - Select Archive 1
Applications 1
Application Modules 1
Application-Scoped JDBC Connection Pools --> Testing 1
Application-Scoped JDBC Connection Pools --> Targets 1
Application-Scoped JDBC Connection Pools --> Notes 1
Application-Scoped JDBC Connection Pools --> Monitoring 1
Application-Scoped JDBC Connection Pools --> Control 1
Application-Scoped JDBC Connection Pools --> Configuration --> Descriptors 1
Application-Scoped JDBC Connection Pools --> Configuration --> General 1
Application-Scoped JDBC Connection Pool --> Deploy 1
Enterprise Applications 1
Overview 1
Tasks 2
Configuring and Deploying a New Enterprise Application or Web Service 2
Viewing Deployed Enterprise Applications 3
Stopping Deployed Enterprise Applications 3
Adding or Editing Enterprise Application Deployment Notes 4
Deleting an Enterprise Application or Application Module 4
Monitoring Enterprise Applications 5
Viewing and Updating Run-Time Deployment Descriptors 5
Attributes and Console Screen Reference for Applications 1
Cluster --> Configuration --> General 1
Cluster --> Configuration --> Multicast 1
Cluster --> Configuration --> Servers 1
Clusters --> Control 1
Clusters --> Deployments --> Applications 1
Clusters --> Deployments --> Classes 1
Clusters --> Deployments --> Connector 1
Cluster --> Deployments --> EJB Modules 1

Clusters --> Deployments --> Web Modules 1
Cluster --> Notes 1
Active Clusters 1
Cluster 1
Cluster --> Monitoring 1
Clusters 1
Tasks 1
Configuring a Cluster 1
Cloning a Cluster 3
Deleting a Cluster 4
Assigning Servers to a Cluster 4
Monitoring a Cluster 4
Adding a Note to a Cluster Configuration 5
Specifying a Server's Cluster Replication Group 5
Specifying a Server's Cluster Weight 6
Specifying the Address of a Server's NIC Card for Cluster Communication 6
Start all Managed Servers 7
Resume all Managed Servers 7
Graceful Shutdown of all Servers 8
Force Shutdown of all Servers 9
Start/Stop a Server 9
Cluster --> Protocols --> HTTP 1
Attributes and Console Screen Reference for Clusters 1
Connector Component --> Configuration --> General 1
Connector Component --> Configuration --> Descriptor 1
ConnectorComponent --> Deploy 1
ConnectorComponent --> Monitoring 1
Connector Component --> Notes 1
Connector Module Deployment Assistant --> Step 2 - Select Targets 1
Connector Component --> Targets 1
Connector Component Deployment Assistant --> Step 3 - Review Choices and Deploy
1
Connector Module Deployment Assistant --> Step 1 - Start 1
Resource Connectors 1
Connector Connection Pool Idle/Leaked Connections 1

- ConnectorConnectionPoolRuntime 1
- ConnectorConnectionRuntime 1
- Connectors 1
- Overview 1
- Tasks 2
- Deploying New Connector (Resource Adapter) Modules 2
- Viewing Deployed Connectors 3
- Stopping a Deployed Connectors 4
- Adding or Editing Connector Deployment Notes 4
- Deleting a Connector 4
- Monitoring Connectors 5
- Viewing and Editing Run-Time Deployment Descriptors 5
- Attributes and Console Screen Reference for Connectors 1
- Install or Update an Application 1
- Deployment Order 1
- Change Deployment Order 1
- Deploying Applications and Modules 1
- Tasks 2
- Preparing Applications and Modules for Deployment 2
- Selecting a Deployment Staging Mode 3
- Setting the Application Staging Mode 4
- Setting the Server Staging Mode 5
- Deploying New Applications and Modules 6
- Changing the Order of Deployment 6
- Changing the Target Servers for a Deployment 7
- Deploying, Redeploying, and Stopping Applications 8
- Removing an Application or Module from the Domain 9
- Attributes and Console Screen Reference for Deployment 1
- Domain --> Control 1
- Domain --> Configuration --> Applications 1
- Domain --> Configuration --> General 1
- Domain --> Configuration --> JTA 1
- Domain --> Configuration --> Logging 1
- Domain --> Configuration --> SNMP 1
- JCOM --> General 1

Domain --> Monitor --> Clusters 1
Domain --> Monitor --> Servers 1
Domain --> Notes 1
Domain 1
Tasks 1
Enabling the Domain-Wide Administration Port 1
Converting the weblogic.properties File 1
SSL Security Files 1
Servlets 2
EJB JAR files and Web App WAR files 3
Attributes and Console Screen Reference for Domains 1
Domain Log Filter --> Configuration 1
Domain Log Filter --> Notes 1
Domain Log Filters --> Target 1
Domain Log Filters 1
Domain Log Filters 1
Specifying the Messages That a Server Forwards to the Domain Log 1
Cloning a Domain Log Filter 3
Deleting a Domain Log Filter 4
Attributes and Console Screen Reference for Domain Log Filters 1
EJB --> Configuration --> Descriptors 1
EJB --> Configuration --> Compiler options 1
EJB --> Configuration --> General 1
EJB --> Deploy 1
EJB --> Monitoring 1
EJB --> Monitoring --> Stateless EJBs 1
EJB --> Monitoring --> Stateful EJBs 1
EJB --> Monitoring --> Message Driven EJBs 1
EJB --> Monitoring --> Entity EJBs 1
EJB Module Deployment Assistant --> Step 3 - Review Choices and Deploy 1
EJB Module Deployment Assistant --> Step 1 - Select Archive 1
EJB --> Targets 1
EJB Module Deployment Assistant --> Step 2 - Select Targets 1
EJB --> Testing 1
EJB --> Notes 1

EJB Modules 1

EJB --> Configuration --> Descriptors --> Stateless Session EJBs 1

EJB --> Configuration --> Descriptors --> Stateful Session EJBs 1

EJB --> Configuration --> Descriptors --> Message-Driven EJBs 1

EJB --> Configuration --> Descriptors --> Entity EJBs 1

EJB 1

Tasks 2

Deploying a New EJB Module 2

Configuring an EJB Module 3

Displaying and Configuring General Information 4

Configuring Deployment Descriptor Values 4

Configuring Compiler Options 5

Viewing an EJB Module's Deployment Status 6

Stopping or Redeploying an EJB Module 7

Setting an EJB Module's Target Server and/or Cluster 7

Monitoring Stateless Session EJBs 8

Monitoring Stateful Session EJBs 9

Monitoring Entity EJBs 9

Monitoring Message-Driven EJBs 9

Testing an EJB Module 10

Adding or Editing EJB Module Deployment Notes 10

Attributes and Console Screen Reference for EJB 1

FileT3 (Deprecated)--> Configuration 1

FileT3 (Deprecated) --> Notes 1

Domain --> FileT3 --> Target and Deploy 1

FileT3 (Deprecated) 1

FileT3 (Deprecated) 1

Tasks 1

Create a File System 1

Modify a File System Path NAME 2

Assign Servers for a FileT3 File System 2

Assign Clusters for a FileT3 File System 3

Attributes and Console Screen Reference for FileT3 1

JDBC 1

Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data

Sources 1

JDBC Objects in a Cluster 3

Application Scoped JDBC Objects 3

JDBC Configuration Guidelines 4

Overview of JDBC Administration 6

About the Administration Console 6

Related Information 7

Attributes and Console Screen Reference for JDBC 1

JDBC Connection Pool --> Configuration --> Connections 1

JDBC Connection Pool --> Configuration --> General 1

JDBC Connection Pool --> Control 1

JDBC Connection Pool --> Testing 1

JDBC Connection Pool --> Monitoring 1

JDBC Connection Pool --> Notes 1

JDBC Connection Pool --> Target and Deploy 1

JDBC Connection Pools 1

Configure a JDBC Connection Pool --> Test Database Connection 1

Configure a JDBC Connection Pool --> Choose database 1

Configure a JDBC Connection Pool --> Define and test connection 1

JDBC Connection Pool Assistant --> Create and deploy 1

Configure a JDBC Connection Pool --> Define connection properties 1

Active JDBC Connections 1

JDBC Connection Pools 1

Configuring JDBC Connection Pools 1

Using the JDBC Connection Pool Assistant 3

Creating and Configuring a JDBC Connection Pool 4

Database Passwords in Connection Pool Configuration 5

Cloning a JDBC Connection Pool 6

Deploying a JDBC Connection Pool to One or More Servers or Clusters 7

Configuring the Statement Cache for a JDBC Connection Pool 7

Adding a Note to a JDBC Connection Pool 8

Application-Scoped JDBC Data Sources and Connection Pools 9

Configuring Application-Scoped Connection Pool Attributes 10

Deploying Application-Scoped Connection Pools 10

Monitoring Application-Scoped Connection Pools 12

Manually Administering an Application-Scoped Connection Pool	13
Testing an Application-Scoped Connection Pool	14
Adding a Note to an Application-Scoped Connection Pool	15
Connection Pool and Data Source Configuration Guidelines	15
Drivers Supported for Local Transactions	16
Drivers Supported for Distributed Transactions Using XA	16
Drivers Supported for Distributed Transactions without XA	16
Configuring JDBC Drivers for Local Transactions	16
Configuring XA JDBC Drivers for Distributed Transactions	19
WebLogic jDriver for Oracle/XA Data Source Properties	24
Additional XA Connection Pool Properties	26
Configuring Non-XA JDBC Drivers for Distributed Transactions	27
Security for JDBC Connection Pools	30
Managing JDBC Connection Pools	30
Testing a JDBC Connection Pool	30
Resetting All Connections in a JDBC Connection Pool	31
Shrinking a JDBC Connection Pool	32
Suspending a JDBC Connection Pool	32
Resuming a JDBC Connection Pool	33
Shutting Down a JDBC Connection Pool	34
Restarting a JDBC Connection Pool	34
Destroying or Deleting a JDBC Connection Pool	35
Clearing the Statement Cache for a JDBC Connection Pool	36
Monitoring Connections in a JDBC Connection Pool	36
Tuning Connection Pools	37
Enabling Connection Requests to Wait for a Connection	37
Automatically Recovering Leaked Connections	38
Initializing Database Connections with SQL Code	40
Connection Testing Options	40
Increasing Performance with the Statement Cache	43
Attributes and Console Screen Reference for JDBC Connection Pools	1
JDBC Data Source --> Configuration	1
JDBC Data Source --> Notes	1
JDBC Data Source --> Target and Deploy	1
JDBC Data Source	1

JDBC Data Source Assistant --> Target the data source 1
JDBC Data Source Assistant --> Configure the data source 1
JDBC Data Source Assistant --> Connect to connection pool 1
JDBC Data Source --> Configuration 1
JDBC Data Source --> Notes 1
JDBC Data Source --> Target and Deploy 1
JDBC Data Source Factory --> Configuration 1
JDBC Data Source Factory 1
JDBC DataSources 1
Configuring JDBC DataSources 1
When to Enable Global Transactions in a Data Source 2
Emulating Two-Phase Commit 3
Creating and Configuring a JDBC Data Source 5
Cloning a JDBC Data Source 6
Deploying a JDBC Data Source to a Server or Cluster 7
Adding a Note to a Data Source 7
Deleting a Data Source 8
JDBC Data Source Factories 8
Creating and Configuring a JDBC Data Source Factory 8
Application-Scoped JDBC Data Sources 9
Monitoring Data Sources 9
Attributes and Console Screen Reference for JDBC Data Sources 1
JDBC MultiPool --> Configuration --> General 1
JDBC MultiPool --> Configuration --> Pools 1
JDBC MultiPool --> Notes 1
JDBC MultiPool --> Target and Deploy 1
JDBC MultiPool 1
JDBC Connection Leak Profile 1
JDBC MultiPools 1
Configuring MultiPools 1
Creating and Configuring a JDBC MultiPool 2
Cloning a JDBC MultiPool 3
Deploying a JDBC MultiPool on One or More Servers and Clusters 3
Adding a Note to a JDBC MultiPool 4
Deleting a JDBC MultiPool 4

Attributes and Console Screen Reference for JDBC MultiPools 1
JMS Connection Consumer --> Configuration 1
JMS Connection Consumer --> Notes 1
JMS Connection Consumer 1
JMS Connection Factory --> Configuration --> Flow Control 1
JMS Connection Factory --> Configuration --> General 1
JMS Connection Factory --> Configuration --> Transactions 1
JMS Connection Factory --> Notes 1
JMS Connection Factory --> Target and Deploy 1
JMS Connection Factory 1
JMS Connection Runtime 1
JMS Destination Key --> Configuration 1
JMS Destination Key --> Notes 1
JMS Destination Key 1
Active JMS Destinations 1
JMS Destination 1
JMS Distributed Queue --> Auto Deploy 1
JMS Distributed Queue --> Configuration --> General 1
Distributed Queue --> Configuration --> Members 5
JMS Distributed Queue --> Configuration --> Thresholds and Quotas 1
JMS Distributed Queue --> Notes 1
JMS Distributed Queue Member --> Configuration 1
JMS Distributed Queue 1
JMS Distributed Topic --> Auto Deploy 1
JMS Distributed Topic --> Configuration --> General 1
Distributed Topic --> Configuration --> Members 5
JMS Distributed Topic --> Configuration --> Thresholds and Quotas 1
JMS Distributed Topic --> Notes 1
JMS Distributed Topic Member --> Configuration 1
JMS Distributed Topic 1
JMS Durable Subscriber Runtime 1
JMS File Store --> Configuration 1
JMS File Store --> Notes 1
JMS JDBC Store --> Configuration 1
JMS JDBC Store --> Notes 1

JMS Pooled Connections 1
JMS Queue --> Configuration --> General 1
JMS Queue --> Configuration --> Overrides 1
JMS Queue --> Configuration --> Redelivery 1
JMS Queue --> Configuration --> Thresholds & Quotas 1
JMS Queue --> Monitoring 1
JMS Queue --> Notes 1
JMS Server --> Configuration --> General 1
JMS Server --> Configuration --> Thresholds & Quotas 1
JMS Server --> Notes 1
JMS Server --> Monitoring 1
JMS Server --> Target and Deploy 1
Active JMS Servers 1
JMS Server 1
JMS Session Pool --> Configuration 1
JMS Session Pool --> Notes 1
Active JMS Session Pools 1
JMS Session Pool 1
JMS Session Runtime 1
JMS Store 1
JMS Template --> Configuration --> General 1
JMS Template --> Configuration --> Override 1
JMS Template --> Configuration --> Redelivery 1
JMS Template --> Configuration --> Thresholds & Quotas 1
JMS Template --> Notes 1
JMS Template 1
JMS Topic --> Configuration --> General 1
JMS Topic --> Configuration --> Multicast 1
JMS Topic --> Configuration --> Overrides 1
JMS Topic --> Configuration --> Redelivery 1
JMS Topic --> Configuration --> Thresholds & Quotas 1
JMS Topic --> Monitoring 1
JMS Topic --> Notes 1
JMS Distributed Destinations 1
Foreign JMS Destination --> Configuration --> Notes 1

Foreign JMS Connection Factory --> Configuration --> Notes 1
Foreign JMS Connection Factory 1
Foreign JMS Destination --> Configuration --> General 1
Foreign JMS Connection Factory --> Configuration --> General 1
Foreign JMS Destination 1
Foreign JMS Server --> Configuration --> General 1
Foreign JMS Server --> Notes 1
Foreign JMS Server --> Target and Deploy 1
Foreign JMS Server 1
JMS Queue --> Configuration --> Expiration Policy 1
JMS Template --> Configuration --> Expiration Policy 1
JMS Topic --> Configuration --> Expiration Policy 1
JMS: Configuring 1
JMS and WebLogic Server 2
Configuring WebLogic JMS 2
JMS Configuration Naming Rules 4
JMS Server Tasks 5
Configuring a JMS Server 5
Targeting and Deploying a JMS Server 7
Monitoring a JMS Server 8
JMS Connection Factory Tasks 8
Using a Default Connection Factory 9
Configuring a JMS Connection Factory 10
Deploying a Connection Factory on Multiple Individual Servers 12
Deploying a Connection Factory on a Cluster 13
JMS Queue and Topic Destination Tasks 14
Creating a JMS Queue 14
Creating a JMS Topic 16
JMS Template Tasks 18
Creating a JMS Template 19
Destination Keys Tasks 21
 Creating a JMS Destination Key 21
JMS Store Tasks 22
JMS File Store Tasks 23
JMS JDBC Store Tasks 25

Session Pools Tasks	28
Creating a JMS Session Pool	29
Connection Consumers Tasks	30
Creating a JMS Connection Consumer	30
JMS Distributed Destination Tasks	31
Guidelines for Configuring Distributed Destinations	31
Creating a Distributed Topic and Creating Members Automatically	32
Creating a Distributed Topic and Adding Existing Physical Topics as Members Manually	35
Creating a Distributed Queue and Creating Members Automatically	37
Creating a Distributed Queue and Adding Existing Physical Queues as Members Manually	40
Creating a JMS Distributed Queue Member	42
Deleting a JMS Distributed Queue Member	43
Creating a JMS Distributed Topic Member	44
Deleting a JMS Distributed Topic Member	45
Monitoring Distributed Destinations	45
Simple Access to Remote or Foreign JMS Providers	46
How WebLogic JMS Accesses Foreign JMS Providers	46
Creating a Foreign JMS Server	47
Creating a Foreign JMS Connection Factory	48
Creating a Foreign JMS Destination	49
Sample Configuration for MQSeries JNDI	51
Attributes and Console Screen Reference for JMS	1
Connection Factories	1
Stores	2
JMS Servers	2
Destinations	2
Session Pools	3
Connection Consumers	4
Templates	4
Destination Keys	4
Distributed Destinations	5
Foreign JMS Servers	5
Monitoring JMS Connections and Sessions	6

JMS: Monitoring	1
Overview	2
Monitoring Active JMS Connections	2
Monitoring Pooled JMS Connections	2
Monitoring Active JMS Servers	3
Monitoring Active JMS Destinations	3
Monitoring Active JMS Session Pools	4
Monitoring Durable Subscribers for Topics	5
Monitoring Distributed Destination System Subscriptions and Proxy Topic Members	5
JMS: Tuning	1
Overview	2
Improving JMS File Store Performance	2
Disabling the Default Synchronous Writes Policy	3
Comparing Synchronous Write Policy Settings	4
Tuning Distributed Destinations	5
Configuring Message Load Balancing on Distributed Destinations	5
Configuring Server Affinity for Distributed Destinations	7
Paging Out Messages To Free Up Memory	8
Configuring Message Paging	8
JMS Message Paging Attributes	14
Related Topics	19
Controlling the Flow of Messages on JMS Servers and Destinations	19
How Flow Control Works	19
Configuring Flow Control	20
Flow Control Thresholds	22
Avoiding Quota Exceptions by Blocking Message Producers	23
Defining a Send Timeout on a JMS Connection Factory	24
Specifying a Blocking Send Policy on a JMS Server	25
Related Topics	25
Handling Expired Messages	26
Defining a Message Expiration Policy	26
Enabling Active Message Expiration	32
Related Topics	32
Transaction Details	1
JTA	1

Configuring Transactions	1
Configuring JTA	2
Configuring Domains for Inter-Domain Transactions	3
Limitations for Inter-Domain Transactions	3
Inter-Domain Transactions for WebLogic Server 8.x and 7.x Domains	3
Inter-Domain Transactions Between WebLogic Server 7.x/8.x and WebLogic Server 6.x Domains	4
Monitoring Transactions	5
Viewing Transaction Statistics for a Server	5
Viewing Transaction Statistics for Named Transactions	5
Viewing Transaction Statistics for Server Resources	6
Viewing Current (Inflight) Transactions for a Server	6
Manually Resolving Current (Inflight) Transactions	6
Transaction Log Files	10
Setting the Transaction Log File Location (Prefix)	11
Setting the Transaction Log File Write Policy	12
Heuristic Log Files	13
Handling Heuristic Completions	14
Abandoning Transactions	15
Moving a Server to Another Machine	16
Transaction Recovery After a Server Fails	16
Transaction Recovery Service Actions After a Crash	17
Recovering Transactions for a Failed Non-Clustered Server	18
Recovering Transactions for a Failed Clustered Server	19
Overview of Transaction Management	24
Attributes and Console Screen Reference for JTA	1
JNDI	1
Overview of JNDI Management	1
What Do JNDI and Naming Services Do?	1
Viewing Objects in the JNDI Tree	2
Loading Objects in the JNDI Tree	2
Attributes and Console Screen Reference for JNDI	1
Jolt Connection Pool --> Configuration --> Addresses	1
Jolt Connection Pool --> Configuration --> General	1
Jolt Connection Pool --> Configuration --> User	1

Jolt Connection Pool --> Notes	1
Jolt Connection Pool --> Target and Deploy	1
Jolt Connection Pool	1
Jolt Connection Pool --> Monitoring	1
Active Jolt Connection Pool	1
Jolt	1
Configuring Jolt for WebLogic Server	1
Set Your Environment	1
Create a Jolt Startup & Shutdown Class	2
Configuring a Jolt Connection Pool	3
Create a Jolt Connection Pool	3
Assign Jolt Connection Pools to a Server	3
Assign Jolt Connection Pools to a Cluster	4
Configuring Connection Failure Handling	4
Configuring a User Security Identity	5
Administering Active Jolt Connection Pools	5
Monitor Active Instances of a Jolt Connection Pool	6
Resetting a Jolt Connection Pool	6
Attributes and Console Screen Reference for Jolt	1
Customize Log View	1
Search Log	1
Server Log	1
Overview of WebLogic Server Log Messages and Log Files	2
Message Attributes	3
Format of Message Output	6
Local Log Files and Domain Log Files	7
Viewing Server Logs	9
Viewing Server Logs from the Administration Console	9
Viewing Server Logs from a Text Editor	11
Viewing the Domain Log	12
Viewing Domain Logs from the Administration Console	13
Viewing Domain Logs from a Text Editor	15
Rotating Log Files	16
Notification of Rotation	19
Specifying Which Messages a Server Sends to Standard Out	20

Printing Messages to Standard Error 21
Viewing Standard Out for a Server Started by the Node Manager 21
Other Logging Tasks 22
Viewing Standard Out for a Server Set Up as a Windows Service 23
Viewing Localized Time Stamps on Windows 23
Redirecting System.out and System.err to a File 24
Disabling a Server from Forwarding Messages to the Domain Log 25
Changing the Name and Location of the Server Log File 25
Changing the Name and Location of the Domain Log File 26
Enabling JDBC Logging 27
Enabling and Configuring an HTTP Log 28
Attributes and Console Screen Reference for Logging 1
Machine --> Configuration --> General 1
Machine --> Configuration --> Node Manager 1
Machine --> Configuration --> Servers 1
Machine --> Monitoring --> Node Manager Log 1
Machine --> Notes 1
Machine 1
Unix Machine --> Configuration --> General 1
Unix Machine --> Configuration --> Node Manager 1
Unix Machine --> Configuration --> Servers 1
Machine --> Monitoring --> Node Manager Status 1
UNIX Machine --> Monitoring --> Node Manager Status 1
UNIX Machine --> Monitoring --> Node Manager Log 1
Unix Machine --> Notes 1
Machines 1
Tasks 1
Configuring a Machine 1
Cloning a Machine 3
Deleting a Machine 3
Assign a WebLogic Server Instance to a Machine 3
Monitoring Node Manager Status 4
Monitoring Node Manager Log 4
Binding to Protected Ports on UNIX 5
Attributes and Console Screen Reference for Machines 1

Mail Session --> Configuration 1
Mail Session --> Notes 1
Mail Session --> Target and Deploy 1
Mail Session 1
Mail 1
Tasks 1
Configuring a New Mail Session 1
Cloning a Mail Session 2
Deleting a Mail Session 3
Assigning a Mail Session 3
Attributes and Console Screen Reference for Mail Sessions 1
General Bridge Destination --> Configuration 1
General Bridge Destination --> Notes 1
General Bridge Destination 1
Messaging Bridge --> Configuration --> Connection Retry 1
Messaging Bridge --> Configuration --> General 1
Messaging Bridge --> Configuration --> Transactions 1
Messaging Bridge --> Notes 1
Messaging Bridge --> Target and Deploy 1
Messaging Bridge 1
Messaging Bridge Runtime 1
JMS Bridge Destination --> Notes 1
JMS Bridge Destination 1
JMS Bridge Destination --> Configuration 1
Messaging Bridge 1
What Is a Messaging Bridge? 2
Messaging Bridge Configuration Tasks 3
About the Bridge's Resource Adapters 3
Deploying the Bridge's Resource Adapters 5
Configuring Source and Target Bridge Destinations 6
Configuring a Messaging Bridge Instance 11
Using the Messaging Bridge to Interoperate with Different WebLogic Server Releases and Domains 17
Naming Guidelines for WebLogic Servers and Domains 17
Enabling Security Interoperability for WebLogic Domains 18

Using the Messaging Bridge To Access Destinations In a Release 6.1 or Later Domain	19
Using the Messaging Bridging To Access Destinations In a Release 6.0 Domain	20
Using the Messaging Bridging To Access Destinations In a Release 5.1 Domain	21
Using the Messaging Bridge to Access a Third-Party Messaging Provider	22
Managing a Messaging Bridge	23
Monitoring All Messaging Bridges	23
Stopping and Restarting a Messaging Bridge	23
Configuring the Messaging Bridge Execute Thread Pool Size	24
Attributes and Console Screen Reference for Messaging Bridge	1
Messaging Bridge	1
JMS Bridge Destination	1
General Bridge Destination	1
Create a New ACL	1
ACL Permission	1
ACL	1
Caching Realm-->ACL	1
Caching Realm --> Authentication	1
Caching Realm --> General	1
Caching Realm --> Groups	1
Caching Realm --> Permissions	1
Caching Realm --> Users	1
Caching Realm --> Notes	1
Caching Realm	1
Custom Realm --> Configuration	1
Custom Realm --> Notes	1
Groups	1
Group-->Group	1
LDAP Security Realm-->Configuration-->General	1
LDAP Security Realm-->Groups	1
LDAP Security Realm-->LDAP Server	1
LDAP Security Realm-->Users	1
LDAP Security Realm-->Notes	1
Windows NT Realm-->Configuration	1
Windows NT Realm-->Notes	1

RDBMS Realm-->Database 1
RDBMS Realm-->General 1
RDBMS Realm-->Schema 1
RDBMS Realm-->Notes 1
Basic Realm 1
Unix Realm-->Configuration 1
Unix Realm-->Notes 1
Unlock User Accounts 1
Users 1
Compatibility Security 1
Tasks 1
Setting Up Compatibility Security: Main Steps 1
Configuring the Identity Assertion Provider in the Realm Adapter Authentication Provider 2
Configuring a Realm Adapter Auditing Provider 3
Changing the System Password 3
Configuring the File Realm 4
Configuring the Caching Realm 5
Enabling the ACL Cache 6
Enabling the Authentication Cache 6
Enabling the Group Cache 7
Enabling the User Cache 7
Enabling the Permission Cache 7
Adding a Note to the Caching Realm 7
Configuring an LDAP V1 Security Realm 8
Enabled Communication between the LDAP Server and WebLogic Server 9
Specifying How Users Are Located in the LDAP V1 Security Realm 9
Specifying How Groups Are Located in the LDAP V1 Security Realm 10
Adding a Note to the LDAP V1 Security Realm 10
Configuring an LDAP Realm V2 10
Adding a Note to the LDAP V2 Security Realm 15
Configuring the Windows NT Security Realm 16
Updating Users Permissions for Windows NT and Windows 2000 17
Adding a Note to the Windows NT Security Realm 19
Configuring the wlauth Program for the UNIX Security Realm 19

Configuring the UNIX Security Realm 21
Adding a Note to the UNIX Security Realm 21
Configuring the RDBMS Security Realm 22
Defining Database Attributes for the RDBMS Security Realm 23
Defining Database Schema for the RDBMS Security Realm 24
Adding A Note to the RDBMS Security Realm 25
Installing a Custom Security Realm 25
Adding A Note To A Custom Security Realm 26
Defining Users 26
Deleting Users 27
Changing the Password of a User 27
Unlocking A User Account 27
Disabling the Guest User 28
Defining Groups 28
Removing Users from a Group 29
Deleting Groups 29
Defining ACLs 30
Protecting User Accounts 31
Installing an Audit Provider 31
Attributes and Console Screen Reference for Compatibility Security 1
Active Directory Authentication Provider-->Active Directory 1
Active Directory Authentication Provider-->General 1
Active Directory Authentication Provider-->Groups 1
Active Directory Authentication Provider-->Membership 1
Active Directory Authentication Provider-->Users 1
Active Directory Authentication Provider-->Details 1
Adjudication Provider 1
Auditing Provider 1
Authentication Providers 1
Authorization Provider 1
Change Password 1
Credential Mapping Provider 1
WebLogic Adjudication Provider-->General 1
WebLogic Adjudication Provider-->Details 1
WebLogic Auditing Provider-->General 1

WebLogic Auditing Provider-->Details 1
WebLogic Authentication Provider-->Details 1
Weblogic Authentication Provider-->General 1
WebLogic Authentication Provider-->Export 1
WebLogic Authentication Provider-->Import 1
WebLogic Authorization Provider-->Details 1
Weblogic Authorization Provider-->General 1
WebLogic Authorization Provider-->Export 1
WebLogic Authorization Provider-->Import 1
WebLogic Credential Mapping Provider-->Details 1
WebLogic Credential Mapping Provider-->General 1
WebLogic Credential Mapping Provider-->Export 1
WebLogic Credential Mapping Provider-->Import 1
Weblogic Identity Assertion Provider-->General 1
WebLogic Identity Assertion Provider-->Details 1
Weblogic Keystore Provider-->Details 1
WebLogic Keystore Provider-->General 1
WebLogic Role Mapping Provider-->Details 1
WebLogic Role Mapping Provider-->General 1
WebLogic Role Mapping-->Export 1
WebLogic Role Mapping Provider-->Import 1
Domain-->Security-->General 1
Domain-->Security-->Filter 1
Domain-->Security-->Embedded LDAP 1
Domain-->Security-->Advanced 1
Domain-->Compatibility Security-->Passwords 1
Domain-->Compatibility Security-->General 1
Domain -->Compatibility Security-->File Realm 1
Domain-->Compatibility Security-->Advanced 1
EJB Policies and Roles 1
Groups 1
Groups-->General 1
Group-->Details 1
Group-->Membership 1
iPlanet Authentication Provider-->Details 1

iPlanet Authentication Provider-->General 1
iPlanet Authentication Provider-->Groups 1
iPlanet Authentication Provider-->iPlanet LDAP 1
iPlanet Authentication Provider-->Membership 1
iPlanet Authentication Provider-->Users 1
Keystore Provider 1
Novell Authentication Provider-->General 1
Novell Authentication Provider-->Details 1
Novell Authentication Provider-->Groups 1
Novell Authentication Provider-->Membership 1
Novell Authentication Provider-->Novell LDAP 1
Novell Authentication Provider-->Users 1
Open LDAP Authentication Provider-->General 1
Open LDAP Authentication Provider-->Details 1
Open LDAP Authentication Provider-->Groups 1
Open LDAP Authentication Provider-->Membership 1
Open LDAP Authentication Provider-->Open LDAP 1
Open LDAP Authentication Provider-->Users 1
Define Policy 1
Security Realm-->User Management 1
Security Realm-->General 1
Security Realm-->Migration-->Export 1
Security Realm-->Migration-->Import 1
Adjudication Provider 1
Auditing Provider 1
Authentication Provider 1
Authorization Provider 1
Credential Mapping Provider 1
Keystores 1
Role Mapping Provider 1
Security Realm-->Testing 1
Security Realm-->UserLockout 1
Realm Adapter Adjudication Provider-->General 1
Realm Adapter Adjudication Provider-->Details 1
Realm Adapter Authentication Provider-->Details 1

Realm Adapter Authentication Provider-->General 1
Realm Adapter Authorization Provider-->General 1
Realm Adapter Authorization Provider-->Details 1
Security Realms 1
Global Roles 1
Security Role-->Conditions 1
Security Role-->General 1
Security Role-->Details 1
Security Role-->General 1
Role Mapping Provider 1
Security Realm-->Providers 1
Users 1
User-->Details 1
User-->General 1
User-->Groups 1
War Policies and Roles 1
Security-->URL Resource-->General 1
Credential Maps 1
Security-->Credential Map 1
Security 1
Tasks 1
The Default Security Configuration in WebLogic Server 1
Defining Groups 2
Deleting Groups 3
Defining Users 3
Deleting Users 4
Changing the Password of a User 5
Protecting User Accounts 5
Unlocking a User Account 6
Defining Global Roles 6
Deleting Global Roles 8
Defining Scoped Roles 9
Deleting Scoped Roles 16
Protecting WebLogic Resources 17
Configuring the Embedded LDAP Server 17

Configuring Backups for the Embedded LDAP Server 18
Configuring a New Security Realm 18
Testing a New Security Realm 21
Configuring an Authentication Provider: Main Steps 21
Setting the JAAS Control Flag 23
Configuring the WebLogic Authentication Provider 24
Configuring an LDAP Authentication Provider 25
Setting LDAP Server and Caching Information 25
Locating Users in the LDAP Directory 26
Locating Groups in the LDAP Directory 27
Locating Members of a Group in the LDAP Directory 27
Configuring Failover for LDAP Authentication Providers 28
Configuring the Realm Adapter Authentication Provider 29
Changing the Order of Authentication Providers 30
Configuring the WebLogic Authorization Provider 30
Configuring the WebLogic Credential Mapping Provider 31
Configuring the WebLogic Role Mapping Provider 32
Configuring a WebLogic Identity Assertion Provider 32
Configuring the WebLogic Adjudication Provider 34
Configuring a WebLogic Auditing Provider 34
Configuring a Custom Security Provider 35
Deleting a Security Provider 36
Configuring a User Name Mapper 36
Configuring a Custom User Name Mapper 38
Importing and Exporting Security Data from Security Realms 38
Importing and Exporting Security Data from Security Providers 40
Changing the Default Security Realm 41
Deleting A Security Realm 42
Creating Credential Maps 42
Configuring Keystores and SSL 43
Configuring Two-Way SSL 46
Enabling Trust Between WebLogic Domains 47
Configuring Connection Filtering 47
Attributes and Console Screen Reference for Security 1
Server --> Notes 1

Configure SSL-->Determine Compatibility Level 1
Server --> Configuration --> Cluster 1
Server --> Configuration --> Deployment 1
Server --> Configuration --> General 1
Server --> Configuration --> Health Monitoring 1
Servers-->Configuration-->Keystores and SSL 1
Server --> Configuration --> Remote Start 1
Server --> Configuration --> Tuning 1
Server --> Control --> Start-Stop 1
Deprecated SSL Identity and Trust Configuration 1
Connection 1
Execute Thread 1
Server --> Deployments --> Startup/Shutdown 1
SSL Identity Configuration 1
Server --> Logging --> Domain 1
Server --> Logging --> HTTP 1
Server --> Logging --> JDBC 1
Server --> Logging --> JTA 1
Socket 1
Inflight JTA Transactions 1
Server --> Services --> Bridge 1
Server --> Services --> File T3 1
Server --> Services --> JDBC 1
Server --> Services --> JMS 1
Server --> Services --> Jolt 1
Server --> Services --> Mail 1
Server --> Services --> WTC 1
Server --> Services --> XML 1
Server 1
ServletSessionRuntime 1
Server --> Protocols --> jCOM 1
Server --> Services --> Virtual Hosts 1
Server --> Services --> Web Services 1
Server --> Protocols --> IIOP 1
Server --> Protocols --> HTTP 1

Servers --> Protocols --> Channels 1
Server --> Monitoring --> Security 1
Server --> Monitoring --> Performance 1
Server --> Monitoring --> JTA 1
Server --> Monitoring --> JMS 1
Server --> Monitoring --> General 1
Server --> Monitoring --> JRockit 1
Server --> Logging --> Server 1
Server --> Deployments --> Web Modules 1
Server --> Deployments --> EJB Modules 1
Server --> Deployments --> Connector 1
Server --> Deployments --> Applications 1
Server --> Control --> Remote Start Output 1
Server --> Control --> JTA Migration Config. 1
Server --> Control --> JMS Migrate 1
Server --> Control --> JMS Migration Config. 1
Server --> Control --> JTA Migrate 1
Servers --> Protocols --> Channels 1
Server --> Protocols --> General 1
Specify Keystore Type 1
Review SSL Private Key Settings 1
Configure Keystore Properties 1
Restart Your Server 1
Execute Queue --> Configuration 1
Execute Queue --> Notes 1
Active Execute Queue 1
Execute Queue 1
Transactions By Name 1
Transactions by Resource 1
Creating, Configuring, and Monitoring Servers 1
Adding and Removing Servers in an Existing Domain 2
Creating a Managed Server in an Existing Domain 3
Cloning a Server 4
Deleting a Managed Server 5
Deleting an Administration Server 5

Configuring the Default Network Connections	6
Configuring Protocols	7
Configuring the Listen Address	11
Configuring the Listen Ports	13
Configuring a Custom Network Channel for a Non-Clustered Server	15
Transitioning Domains from Development to Production Environments: Main Steps	16
Changing the Runtime Mode	17
Other Configuration Tasks	18
Replicating a Domain's Configuration Files for Managed Server Independence	19
Disabling Managed Server Independence	19
Monitoring a Server	20
Monitoring a Server Instance from the Administration Console	21
Determining the Platform on Which a Server Is Running	21
Monitoring the JRockit Virtual Machine	22
Configure Self-Health Monitoring, Shutdown, and Restart for Managed Servers	22
Attributes and Console Screen Reference for Servers	1
Server --> Configuration	1
Servers --> Protocols	2
Servers --> Logging	2
Servers --> Monitoring	2
Servers --> Control	3
Servers --> Deployments	3
Server --> Services	3
Server --> Notes	4
Miscellaneous	4
Starting and Stopping Servers	1
Version Requirements for Starting Servers	2
Starting Administration Servers	2
Alternate Ways to Start Administration Servers	3
Starting an Administration Server from the Windows Start Menu	4
Starting an Administration Server When the Host Computer Boots	4
Starting an Administration Server With the java weblogic.Server Command	4
Starting Managed Servers from the Administration Console	5
Configure Startup Arguments for Managed Servers	7
Starting Managed Servers From a WebLogic Server Script	8

Alternate Ways to Start Managed Servers 10

Starting All Managed Servers in a Domain 11

Starting a Managed Server in the STANDBY State 12

Creating Scripts That Use the Node Manager 13

Starting a Managed Server With the java weblogic.Server Command 13

Starting a Managed Server When the Host Computer Boots 14

Starting a Managed Server If the Administration Server is Unavailable 14

Providing Usernames and Passwords to Start and Stop a Server 14

Specifying an Initial Administrative User for a Domain 15

Boot Identity Files 15

Specifying User Credentials When Starting a Server with the Node Manager 21

Other Startup Tasks 22

Configuring a Connection to the Administration Server 22

Resuming a Server 24

Specifying Java Options for a WebLogic Server Instance 25

Changing the JVM that Runs Servers 27

Shutting Down Instances of WebLogic Server 28

Shutting Down a Server 28

Shutting Down All Managed Servers in a Domain 29

Controlling Graceful Shutdowns 30

Killing the JVM 31

SMNP --> Traps --> Attribute Change --> Configuration 1

SMNP --> Traps --> Attribute Change 1

SMNP --> Traps --> Counter Monitor --> Configuration 1

SMNP --> Traps --> Counter Monitor --> Servers 1

SMNP --> Traps --> Counter Monitor 1

SMNP --> Traps --> Gauge Monitor --> General 1

SMNP --> Traps --> Gauge Monitor --> Servers 1

SMNP --> Traps --> Gauge Monitor 1

SMNP --> Traps --> SNMP Monitors 1

SMNP --> Traps --> Log Filter --> General 1

SMNP --> Traps --> Log Filter 1

SMNP --> Proxies --> General 1

SMNP --> Proxies 1

SMNP --> Traps --> String Monitor --> General 1

SNMP --> Traps --> String Monitor --> Servers 1
SNMP --> Traps --> String Monitor 1
SNMP --> Trap Destinations --> General 1
SNMP --> Trap Destinations 1
Configuring SNMP and WebLogic Server 1
Configuring SNMP and WebLogic Server: Main Steps 2
Enabling and Configuring the WebLogic SNMP Agent 3
Creating a Trap Destination 4
Determining Which WebLogic Server Attributes to Monitor 6
Determining the Scope of an SNMP Monitor 7
Commonly Monitored Attributes 7
Determining the Names of Other Attributes 10
Configuring WebLogic Server to Generate Traps 12
Configuring a Notification Log Filter 14
Configuring an Attribute Change 15
Configuring a String Monitor 16
Configuring a Gauge Monitor 18
Configuring a Counter Monitor 20
Disabling Trap Generation 22
Configuring an SNMP Proxy 23
Attributes and Console Screen Reference for SNMP 1
Startup or Shutdown Class 1
Shutdown Class --> Configuration 1
Shutdown Class --> Target and Deploy 1
Shutdown Class --> Notes 1
Startup Class --> Configuration 1
Startup Class --> Target and Deploy 1
Startup Class --> Notes 1
Startup and Shutdown Classes 1
Configuring a Server to Use a Startup or Shutdown Class: Main Steps 2
Configure a Startup or Shutdown Class 2
Clone a Configuration of a Startup or Shutdown Class 4
Assign a Startup or Shutdown Configuration to Servers or Clusters 5
Add the Class to the Server's Classpath 6
Modify an Existing Startup or Shutdown Configuration 8

Delete a Startup or Shutdown Class Configuration 8

Attributes and Console Screen Reference for Startup and Shutdown 1

Tasks --> Status 1

Tasks --> Details 1

Migration Task Runtime --> migration 1

MigrationTaskRuntime --> Details 1

Tasks 1

Tasks Status 1

View Details about a Task 1

Attributes and Console Screen Reference for Tasks 1

Virtual Host --> Configuration --> General 1

Virtual Host --> Configuration --> HTTP 1

Virtual Host --> Configuration --> Logging 1

Virtual Host --> Notes 1

Domain --> VirtualHost --> Target and Deploy 1

Virtual Host 1

Virtual Hosts 1

Tasks 1

Configuring a VirtualHost 1

Cloning a VirtualHost 2

Specifying HTTP Log File Settings for a Virtual Host 3

Deleting a VirtualHost 4

Assigning a VirtualHost 5

Targeting Web Applications to the Virtual Host. 5

Associating a Virtual Host with a Server 5

Removing an Associated Virtual Host 6

Attributes and Console Screen Reference for Virtual Hosts 1

Web Application --> Configuration --> Descriptor 1

Web Application --> Configuration --> General 1

Web Applications --> Configuration --> Other 1

Web Applications --> Configuration --> Files 1

Web Application --> Deploy 1

Web Application --> Monitor --> Web Application 1

Web Application --> Monitor --> Sessions 1

Web Application --> Monitor --> Servlets 1

Web Application --> Notes 1
Web Application Deployment Assistant --> Step 2 - Select Targets 1
Web Application --> Targets 1
Web Application --> Configuration --> Testing 1
Web Application Deployment Assistant --> Step 1 - Select Archive 1
Web Application Deployment Assistant --> Step 3 - Review Choices and Deploy 1
Web Applications 1
ServletRuntime 1
Web App Component Runtime 1
Web Applications 1
Overview 1
Tasks 2
Designating a Default Web Application 2
Deploying a New Web Application 2
Testing the Deployment 3
Viewing Deployed Web Applications 4
Stopping Deployed Web Applications 4
Adding or Editing Web Application Deployment Notes 4
Deleting a Web Application 5
Monitoring Web Applications and Servlets 5
Viewing and Updating Run-Time Descriptor Elements 6
Attributes and Console Screen Reference for Web Applications 1
Web Service Component Runtime 1
Web Services 1
Web Service --> Testing 1
Web Service --> Targets 1
Web Service --> Notes 1
Web Service --> Monitoring --> Web Services 1
Web Service --> Monitoring --> Sessions 1
Web Service --> Monitoring --> Servlets 1
Web Service --> Deploy 1
Web Service --> Configuration --> Other 1
Web Service --> Configuration --> General 1
Web Service --> Configuration --> Files 1
Web Service --> Configuration --> Descriptor 1

- Web Services 1
 - Overview 1
 - Tasks 2
 - Configuring and Deploying a New Web Service 2
 - Viewing Deployed Web Services 4
 - Undeploying Deployed Web Services 5
 - Deleting a Web Service 5
 - Viewing Web Service Deployment Descriptors 6
 - Testing a Web Service From Its Home Page 7
 - Configuring Web Service Reliable SOAP Messaging 8
 - Attributes and Console Screen Reference for Web Services 1
 - WLEC Connection Pool --> Configuration --> General 1
 - WLEC Connection Pool --> Configuration --> Security 1
 - WLEC Connection Pool --> Notes 1
 - WLEC Connection Pool --> Target and Deploy 1
 - WLEC Connection Pool 1
 - WLEC Connection Pool --> Monitoring 1
 - WLEC 1
 - Configuring WLEC for WebLogic Server 1
 - Configuring WLEC Connection Pools 2
 - Configure a New WLEC Connection Pool 2
 - Monitor Active Instances of a WLEC Connection Pool 2
 - Assign WLEC Connection Pools to a Server 3
 - Assign WLEC Connection Pools to a Cluster 3
 - Configuring User Security 4
 - Attributes and Console Screen Reference for WLEC 1
 - Exported Services --> General 1
 - Exported Services 1
 - Imported Services --> General 1
 - Imported Services 1
 - Local Tuxedo Access Points --> Connections 1
 - Local Tuxedo Access Points --> General 1
 - Local Tuxedo Access Points --> Security 1
 - Local Tuxedo Access Points 1
 - Passwords --> Configuration 1

Passwords 1
Remote Tuxedo Access Points --> Connections 1
Remote Tuxedo Access Points --> General 1
Remote Tuxedo Access Points --> Security 1
Remote Tuxedo Access Points 1
Resources --> Configuration 1
Resources 1
WTC Service --> General 1
WTC Service --> Contents --> Exported 1
WTC Service --> Contents --> Imported 1
WTC Service --> Contents --> Local Access Points 1
WTC Service --> Contents --> Passwords 1
WTC Service --> Contents --> Remote Access Points 1
WTC Service --> Contents --> Queuing Bridge 1
WTC Service --> Contents --> Redirections 1
WTC Service --> Contents --> Resources 1
WTC Service --> Notes 3
WTC Service --> Target and Deploy 1
WTC Service 1
TUXEDO Queuing Bridge --> Connections 1
TUXEDO Queuing Bridge --> Factories 1
TUXEDO Queuing Bridge --> General 1
TUXEDO Queuing Bridge --> Priority Mapping 1
TUXEDO Queuing Bridge 1
Redirection --> General 1
Redirection 1
WebLogic Tuxedo Connector (WTC) 1
WTC Service Tasks 1
Creating a WTC Service 1
Deleting a WTC Service 2
Assign a WTC Service to a Server 2
Local Tuxedo Access Point Tasks 3
Creating a Local Tuxedo Access Point 3
Configuring Connection Attributes for Local Tuxedo Access Points 3
Configuring Security Attributes for Local Tuxedo Access Points 4

Deleting a Local Tuxedo Access Point 4
Remote Tuxedo Access Point Tasks 5
Creating a Remote Tuxedo Access Point 5
Configuring Connection Attributes for Remote Tuxedo Access Points 5
Configuring Security Attributes for Remote Tuxedo Access Points 6
Deleting a Remote Tuxedo Access Point 6
Exported Services Tasks 7
Creating an Exported Service 7
Deleting an Exported Service 7
Imported Services Tasks 8
Creating Imported Services 8
Deleting an Imported Service 8
Password Tasks 9
Creating a Password Configuration 9
Deleting a Password Configuration 10
Resource Tasks 10
Creating a Resource 10
Removing a Resource 10
Tuxedo Queuing Bridge Tasks 11
Creating a Tuxedo Queuing Bridge Connection 11
Remove a Tuxedo Queuing Bridge Connection 11
Configuring Connection Attributes for a Tuxedo Queuing Bridge Connection 12
Configuring Connection Factories for a Tuxedo Queuing Bridge Connection 12
Configuring Priority Mapping for a Tuxedo Queuing Bridge Connection 12
Creating a Tuxedo Queuing Bridge Redirection 13
Deleting a Tuxedo Queuing Bridge Redirection 13
Attributes and Console Screen Reference for WebLogic Tuxedo Connector 1
XML Entity Spec Registry Entry --> Configuration 1
XML Entity Spec Registry Entry --> Notes 1
XML Entity Spec Registry Entry 1
XML Parser Select Registry Entry --> Configuration 1
XML Parser Select Registry Entry --> Notes 1
XML Parser Select Registry Entry 1
XML Registry --> Configuration 1
XML Registry --> Notes 1

XML Registry --> Target and Deploy 1
XML Registry 1
XMLEntity Cache --> Session --> Rejections 1
XMLEntity Cache --> Current --> General 1
XMLEntity Cache --> Current --> Entry Resource Usage 5
XMLEntity Cache --> Session --> General 9
XMLEntity Cache --> Session --> Entry Resource Usage 13
XMLEntity Cache --> Historical --> Rejections 17
XMLEntity Cache --> Current --> Total Resource Usage 1
XMLEntity Cache --> Historical --> Entry Resource Usage 1
XMLEntity Cache --> Historical --> General 1
XML 1
Tasks 2
Configuring a Parser or Transformer Other Than the Built-In 2
Configuring a Parser for a Particular Document Type 3
Configuring External Entity Resolution 5
Configuring the External Entity Cache 7
Monitoring the External Entity Cache 8
Attributes and Console Screen Reference for XML 1

1 Common System Administration Tasks

Follow the following links for information on performing the following common tasks. Links labeled **e-docs** require an Internet connection.

How do I configure a new Server?

- See “Adding and Removing Servers in an Existing Domain” on page 495-2

How do I deploy an application?

- *Deployment Overview*: See “Deploying Applications and Modules” on page 62-1
- *Enterprise Applications*: See “Configuring and Deploying a New Enterprise Application or Web Service” on page 26-2
- *Enterprise JavaBeans (EJB)*: See “Deploying a New EJB Module” on page 103-2
- *Web Applications*: See “Deploying a New Web Application” on page 559-2
- *J2EE Connectors*: See “Deploying New Connector (Resource Adapter) Modules” on page 57-2

How do I create a JDBC connection pool?

- “Configuring JDBC Connection Pools” on page 127-1

How do I secure my application?

- See [Securing WebLogic Resources](#)

How do I configure a default Web Application?

(A Web Application served in response to `www.myhost.com/`)

- See “Designating a Default Web Application” on page 559-2

How do I change the Node Manager port?

- See [Configuring, Starting, and Stopping Node Manager](#)

How do I enable the Administration port?

- See “Enabling the Domain-Wide Administration Port” on page 74-1

How do I create a cluster?

- See “Configuring a Cluster” on page 41-1

WebLogic Server

Administration Console Online Help

8.1

Console --> Version

This tab displays the version of the WebLogic Server instance that is hosting the Administration Console. Information on other active modules is also displayed.

For more information, see “Using the Administration Console” on page 6-1



Console --> Preferences

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use the Preferences tab to customize the behavior of the Administration Console. These attributes are not saved with the domain's configuration in the `config.xml` file. They are saved as Web browser cookies (except for Remember Last Tab). If you use the same browser and machine, your settings are retained each time you open the Administration Console.

For more information, see “Using the Administration Console” on page 6-1.

Tasks

“Changing Administration Console Behavior” on page 6-15

“Changing the Default Language of the Administration Console” on page 6-14

“Changing the Monitoring Interval” on page 6-13

Related Topics

[Overview of WebLogic System Administration](#) in the *BEA WebLogic Server Administration Guide*

[“Starting and Stopping Servers”](#) on page 497-1

Attributes

Table 3-1

Attribute	Description	Range of Values	Default Value
Language	Selects the language used for the console and the help system.	English, Japanese	English
Auto-refresh Every	Interval for refreshing the data displayed on monitoring screens.	Integer, in seconds	10
Poll for Graph Data Every	Interval for refreshing the data displayed in monitoring graphs.	Integer, in milliseconds	500
Display Advanced features by default	Specifies whether the Console automatically displays advanced attributes. If you do not select this option, you can still display advanced attributes by clicking the Advanced Options... Show link.	Boolean Enabled = selected Disabled = not selected	Not Selected
Remember Last Tab	Specifies whether the Administration Console displays the tab that was most recently viewed when you return to a console node.	Boolean Enabled = selected Disabled = not selected	Selected
Display help text for each attribute	Specifies whether the Administration Console should show help text for each console screen and attribute.	Boolean Enabled = selected Disabled = not selected	Selected

Attribute	Description	Range of Values	Default Value
Use Navigation Tree	Enables or disables the ability to use the navigation tree on the console.	Boolean Enabled = selected Disabled = not selected	Selected



Display Deployment Descriptor

[Tasks](#) [Related Topics](#)

Overview

This window displays the content of a deployment descriptor. Deployment descriptors are packaged with application modules such as Web Applications, EJB jar files, Resource Adaptors (Connectors), and Enterprise Applications (EAR files). Deployment descriptors contain data that define deployment parameters for an application module.

You cannot modify a deployment descriptor from this screen.

Tasks

“Preparing Applications and Modules for Deployment” on page 62-2

Related Topics

“Deploying Applications and Modules” on page 62-1

Packaging and Deploying WebLogic Server Applications

[Deployment](#)

[WebLogic Builder Online Help](#) (The WebLogic Builder tool helps you create application modules and deployment descriptors.)



Customizing a Table View

[Tasks](#) [Related Topics](#)

Overview

When you select a node in the Administration Console, the console initially displays tabular lists of configured objects such as servers or application modules. These tables also contain columns that display some of the configuration attributes for the object. You can select which columns appear in the table and which column is used to sort the list.

Tasks

“Customizing Table Views” on page 6-15

Related Topics

“Using the Administration Console” on page 6-1



1 Using the Administration Console

[“Attributes and Console Screen Reference for Administration Console” on page 7-1]

Overview

The *System Administration Console* is a Web browser-based, graphical user interface you use to manage a *WebLogic Server domain*. A WebLogic Server domain is a logically related group of WebLogic Server resources that you manage as a unit. A domain includes one or more WebLogic Servers and may also include WebLogic Server *clusters*. Clusters are groups of WebLogic Servers that work together to provide scalability and high-availability for applications. You deploy and manage your applications as part of a domain.

One instance of WebLogic Server in each domain is configured as an *Administration Server*. The Administration Server provides a central point for managing a WebLogic Server domain. All other WebLogic Server instances in a domain are called *Managed Servers*. In a domain with only a single WebLogic Server instance, that server functions both as Administration Server and Managed Server. The Administration Server hosts the Administration Console, which is a Web Application accessible from any supported Web browser with network access to the Administration Server.

For more information on WebLogic Server domains, see [Overview of WebLogic System Administration](#).

Uses of the System Administration Console

You can use the System Administration Console to:

- Configure, start, and stop WebLogic Server Instances
- Configure WebLogic Server Clusters
- Configure WebLogic Server Services, such as database connectivity (JDBC), and messaging (JMS).
- Configure security parameters, including managing users, groups, and roles.
- Configure and Deploy your applications.
- Monitor server and application performance.
- View server and domain log files.
- View application deployment descriptors.
- Edit selected runtime application deployment descriptor elements.

System Administration Architecture

The Administration Console uses the Java Management Extension (JMX) API as its underlying architecture. The JMX API models system administration functions using Java objects called *MBeans*. Each MBean contains a set of *attributes* that define parameters for various management functions and *operations* that define administrative actions. The Administration Console allows you to access these MBean attributes and operations using a convenient graphical user interface. The Administration Server persists the values of MBean attributes in a domain-wide configuration file called `config.xml`. (Some security attributes, however, are persisted in other files.)

You can also perform nearly all of the management tasks that you perform using the Administration Console using the WebLogic Server command-line interface. The command-line interface is useful for scripting of WebLogic Server system administration operations or for system administrators who prefer to use a command-line environment. For more information, see the [WebLogic Server Command Reference](#).

Additional documentation on WebLogic Server system administration is available on BEA's e-docs Web site. For a list of these documents, see [System Administration](#).

Starting and Logging Out of the Administration Console

This section describes how to begin using the Administration Console and how to end your console session.

Starting the Administration Console

1. Start a WebLogic Administration Server. For more information, see “Starting Administration Servers” on page 497-2.
2. Open a supported Web browser (See [Browser Support for the WebLogic Server Console](#) for a list of supported Web browsers.) and open the following URL:

```
http://hostname:port/console
```

Where *hostname* is the DNS name or IP address of the Administration Server and *port* is the address of the port on which the Administration Server is listening for requests (7001 by default). If you start the Administration Server using Secure Socket Layer (SSL), you must add *s* after *http* as follows:

```
https://hostname:port/console
```

For more information about setting up SSL for system administration, see “Servers-->Configuration-->Keystores and SSL” on page 436-1.

Note: If your Web browser is configured to send HTTP requests to a proxy server, you may need to configure your browser so that it does not send Administration Server HTTP requests to the proxy. If the Administration Server is running on the same machine as your Web browser, configure your browser so that requests sent to `localhost` or IP address `127.0.0.1` are not sent to the proxy server.

3. When the login page appears, enter the user name and the password you used to start the Administration Server, (you may have specified this user name and password during the installation process) or enter a user name that is granted one of the default global security roles.

The WebLogic Server Administration Console allows users to edit configurations or to perform other operations based on the default global security role they are granted. If this security role does not permit editing of configuration data, for example, the data is displayed in the Administration Console but is not editable. If the user attempts to perform a control operation that is not permitted, such as starting or stopping servers, the Administration Console displays an Access Denied error.

In this release of WebLogic Server, users granted the `Admin` default global role can perform any function using the Administration Console. The other default global roles (`Deployer`, `Monitor`, and `Operator`) primarily have read-only permissions, except for functions specific to their security role. [Table 1-1](#) lists the default global roles and describes the view of the Administration Console for users granted these security roles.

4. Click the **Sign In** button.

Table 1-1 Security Role-Based Administration Console Views

Default Global Role	Administration Console View
Admin	<p>Everything is visible, including all:</p> <ul style="list-style-type: none"> ■ Apply buttons (when viewing the configuration of an existing entity) ■ "Create a new..." links ■ Clone and Delete icons (part of table views) ■ Create, Clone, and Delete operations in the navigation tree <p>Additionally, users in the Admin security role can specify security information and configure jCOM services. More specifically, they can see the Security, CompatibilitySecurity, and Services —> jCOM nodes in the navigation tree. For security, they can also right-click on WebLogic resources to define a security policy, scoped role, or credential mapping.</p>
Deployer	<p>Everything is visible <i>except</i>:</p> <ul style="list-style-type: none"> ■ Apply buttons (typically appear when viewing the configuration of an existing entity) ■ Clone and Delete icons for <i>non-deployable</i> objects (typically part of table views) ■ "Create a new..." links for <i>non-deployable</i> objects ■ Create, Clone, and Delete operations for <i>non-deployable</i> objects (typically part of the navigation tree) <p>Note: <i>Deployable</i> objects include applications, Web application modules, EJB modules, Connector modules, JDBC connection pools, JDBC data sources, JDBC MultiPools, JDBC data source factories, JMS servers, and WTC Services. Applications do not have Clone icons.</p> <p>Additionally, users in the Deployer security role can edit the Load Order field of an application.</p>
Monitor or Operator	<p>Everything is visible <i>except</i>:</p> <ul style="list-style-type: none"> ■ Apply buttons (typically appear when viewing the configuration of an existing entity) ■ Clone and Delete icons (typically part of table views) ■ "Create a new..." links ■ Create, Clone, and Delete operations (typically part of the navigation tree) <p>In addition, all fields are read-only for users in the Monitor and Operator security roles.</p>

Notes: For fields that are read-only and do not have an assigned value, the text “(No Value Assigned)” is shown.

No user, regardless of the security role they are granted, can view the non-encrypted version of an encrypted attribute in the Administration Console.

For more information on using security roles to control access to the Administration Console, see "[Security Roles](#)" in *Securing WebLogic Resources*.

To add or modify users see “Defining Users” on page 428-3.

Logging Out of the Administration Console

To log out of the Administration Console:

1. Right-click on the Console node in the Navigation Tree.
2. Select **Logout...**

-or-

1. Click **Logout** on the banner near the top of the right panel of the Administration Console, as shown in Figure 6-1.

Figure 6-1 Logout Function



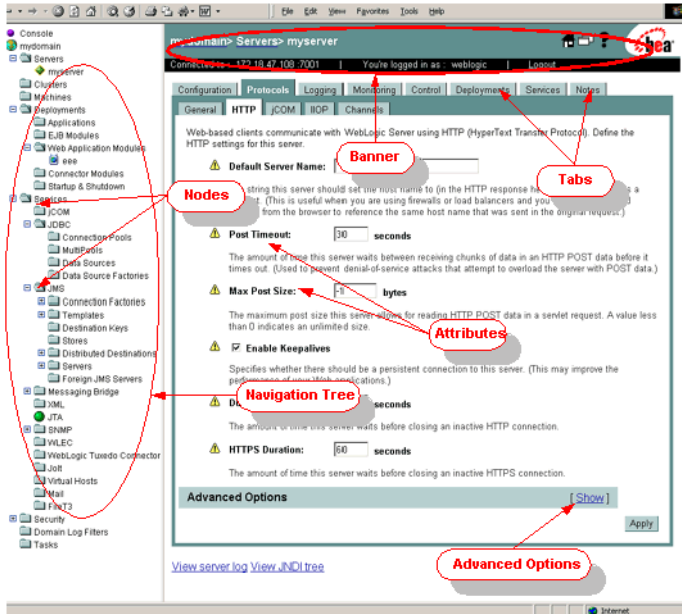
Configuring Your Domain Using the Administration Console

To configure your domain using the Administration Console, you browse to a console page that you use to view or modify configuration attributes or to execute system administration operations.

Navigating Through the Administration Console




This section discusses the various components of the Administration Console user interface and how to use them to manage your domain.

Figure 6-2 Administration Console Layout



Banner

The banner area of the console appears in the top portion of the right panel of the console. In the banner you can:

- See the name and type of configuration or monitoring object currently displayed in the console.
- See the host name or IP address of the Administration Server.
- See the username used to log in to the console.
- Log out of the console by clicking the [Logout](#) link.
- Click the  icon to display the console home page.
- Click the  icon to get help on console operations.
- Click the  icon to display the contents of the right panel of the console in a new browser window.

Navigation Tree and Nodes

The left panel in the Administration Console contains a navigation tree that you use to navigate to the console pages you use to manage your domain. By selecting (left-clicking) a *node* in the tree, you can access console pages related to the node, which are displayed in the right panel of the console. If a node in the tree is preceded by a plus sign, you can click on the plus sign to expand the tree to access additional resources.


By right-clicking on a node, you can also access a variety of operations such as viewing log files, viewing the JNDI tree, creating security policies, viewing monitoring data, creating, deleting or cloning configuration objects, and opening the configuration page in a new window.

Console Screens

Once you select a node from the navigation tree, a list of objects that you can configure appears as branches of the node and as a tabular list displayed in the right panel of the console. To select an object, click the branch for the object you want to configure or click its link in the table in the right panel.

After you select the object, you will see a tabbed interface in the right panel. Clicking on a tab (Some tabs also contain sub-tabs.) displays a console page containing configuration attributes, controls for administrative operations, or displays you use to monitor the current state of the object.

You change configuration data by editing the attribute fields displayed in the right panel. After you make your changes, click the Apply button. Changing some attributes requires that you restart one or more WebLogic Server instances. These attributes are

marked with the  icon.

Some attributes are displayed in a light grey color and are not editable. These attributes may or may not be editable depending on the value of another attribute. For example, on the Servers --> Configuration --> General tab, the attribute called **SSL listen Port** is only editable if the **SSL Listen Port Enabled** box is checked.


Advanced Options

Some console screens contain a section labeled **Advanced Options**. Click the [\[Show\]](#) link to display the Advanced Options section, or click the [\[Hide\]](#) link if you do not want to see the Advanced Options. Advanced options contain seldom used attributes that usually do not require editing.

Figure 6-3 Advanced Options



Home Page

The Administration Console home page is the first page displayed in the right panel when you log into the Administration Console. You can always display this page by clicking the  icon in the banner area of the console. From this page you can navigate to configuration and monitoring pages you use to manage your domain. (You can also use the Navigation Tree to access these pages.)

Editing More Than One Domain

The Administration Server and Console can manage only a single, active domain. To manage another domain, start the Administration Console hosted by the Administration Server of that domain.

Creating New Configuration Objects

To manage a WebLogic Server instance, service, or application, you create or edit configuration objects using the Administration Console.

To create a new configuration object:

1. Select the type of object you want to create using the Navigation Tree.
2. Click the link in the right panel labeled Create a new *object*, where *object* is the name of a WebLogic Server instance, service, resource, or application.
3. Follow the prompts on the screen to create the object. Once the object is created you can edit its configuration by navigating through the tabs displayed in the right panel.


Deleting Configuration Objects

To delete a configuration object:

1. Select the object you want to delete using the Navigation Tree.
2. Right-click on the object name in the Navigation Tree and select Delete. The Administration Console asks you to confirm the deletion.
3. Click the Yes button.

-Or-

1. Select the parent object (such as Servers, or EJB Modules) in the Navigation Tree. A table of configured objects appears in the right panel.

2. Click the  icon in the row of the table containing the object you want to delete. The Administration Console asks you to confirm the deletion.
3. Click the Yes button.


Cloning Configuration Objects

Cloning an object allows you to create an identical object with a new name. This feature can be useful when you need to create many objects, such as server instances, with identical configurations. Note that each new cloned object may require additional editing after cloning.

To clone an object:

1. Select the object you want to clone using the Navigation Tree.
2. Right-click on the object name in the Navigation Tree and select Clone. The Administration Console displays the Configuration page for the object.
3. Enter a unique name for your object and edit any other configuration attributes that differ from the object you are cloning.
4. Click the Clone button.


-Or-

1. Select the parent node of the object you want to clone (such as Servers, or EJB Modules) in the Navigation Tree. A table of configured objects appears in the right panel.
2. Click the  icon in the row of the table containing the object you want to clone. The Administration Console displays the Configuration page for the object.
3. Enter a unique name for your object and edit any other configuration attributes that differ from the object you are cloning.
4. Click the Clone button.

Editing Configuration Objects


To access the configuration data you want to edit:

1. Select the node in the Navigation Tree for the server instance, service, or application you want to configure.
2. Select the tab in the right panel of the console that contains the configuration parameters you want to edit.
3. Edit the configuration data by changing the appropriate fields in the right panel.
4. Click Apply. Many configuration changes require you to restart the affected server(s). Non-dynamic attributes that require you to restart a server are labeled

with a  icon. When you apply changes to a non-dynamic attribute, the icon blinks to remind you to restart the affected server(s).

Monitoring Using the Administration Console

To monitor a domain resource either right click on the resource in the navigation tree and select a monitoring option, or navigate to the resource and select the monitoring tab from the right panel. The data displayed represents the current state of the resource.

To update the information click the  icon in the upper right section of the screen. The data will refresh regularly until you click the icon again. The icon displays a circular animation to that the Administration Server is working to refresh the data. By default, the data refreshes every 10 seconds, or you can specify the refresh interval. See “Changing the Monitoring Interval” on page 6-13.

Changing the Monitoring Interval

When you use the Administration Console to monitor a resource of a WebLogic Server domain, you can set the interval at which the console refreshes the data from the monitored resource.

To set the refresh interval:

1. Select the Console node in the navigation tree.
2. Click the Preferences tab.
3. Set the refresh interval, in seconds in the **Auto-refresh every:** field.
4. Click Apply.

Note: The value of the **Auto-refresh every** field is not persisted in the domain's configuration file (`config.xml`). It is, however, persisted as a browser cookie. If you access the Administration Console using the same computer and Web browser, this setting is retained.

Changing the Monitoring Graph Polling Interval

Some monitoring screens in the Administration Console display real-time graphs of performance data. To change the interval at which monitoring graphs refresh their display:

1. Select the Console node in the navigation tree.
2. Click the Preferences tab.
3. Set the interval, in milliseconds in the **Poll for graph data every:** field.
4. Click Apply.

Customizing the Administration Console

This section discusses how you can customize the appearance and functionality of the Administration Console.

Note: The attributes described in this section are not persisted with the domain configuration in the `config.xml` file.

The following attributes are persisted as browser cookies -- if you access the Administration Console using the same computer and Web browser, these settings are retained:

- Use Navigation Tree
- Display help text for each attribute
- Auto Refresh every

Changing the Default Language of the Administration Console

If you have a Japanese or English language version of WebLogic Server, you can change the default language used in the Administration Console and the online help. Normally, the console uses your browser's language settings to determine which language to display, and you do not need to change this setting. To change the default language:

1. Select the Console node in the navigation tree.
2. Click the Preferences tab.
3. Set the language from the drop-down list labeled Language.
4. Click Apply.

Changing Administration Console Behavior

To change the way the Administration Console displays data:

1. Select the Console node in the navigation tree.
2. Click the Preferences tab.
3. If you want the console to remember the last tab displayed when navigating from node to node using the navigation tree, check the **Remember Last Tab** check box.
4. If you want the console to display descriptions of each screen and attribute, check the **Display help text for each attribute** box.
5. If you want the console to display the navigation tree (recommended), check the **Use Navigation Tree** check box.
6. If you want the console to display “Advanced Options” on page 6-10 by default, check the **Display Advanced Features by Default** box. If you do not select this option, you can still display advanced fields by clicking the [\[Show\]](#) link on any screen containing Advanced Options.
7. Click Apply.

Customizing Table Views

When the data displayed in the right panel is a table of data listing objects of a particular type, you can customize the table by adding or subtracting columns. You can also sort the data tables by clicking on the column headers.

To customize the *columns* that appear in a table:

1. Navigate to an object table.
2. Click on the **Customize this view** link at the top of the table.
3. Choose one or more attributes to display by selecting the attribute name in the **Available** section (on the left) of the **Columns to display** box.
4. Click the right arrow button to move the column to the Chosen box.

5. Repeat steps 1 through 4 for each column you want to display.

6. Click **Apply**

To customize the *order* in which the rows are sorted for display:

1. Navigate to an object table.

2. Click on the **Customize this view** link at the top of the table.

3. Choose the attribute to use when sorting the display by selecting the attribute name in the **Available** section (on the left) of the **Sort rows by** box.

4. Click the right arrow button to move the column to the Chosen box.

5. Click **Apply**

Viewing the WebLogic Server Version Number

To view information on the version number of your Administration server:

1. Select the Console node in the navigation tree.

2. Click the Versions tab. Version information appears in the right panel.

Viewing Server and Browser Information

To view WebLogic Server version, system properties, and browser header information:

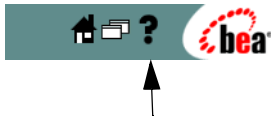
1. Right-click on the Console node in the navigation tree.

2. Select View server & browser info. Server and browser information appears in the right panel.

Getting Help

Documentation on how to use the Administration Console is included with the console application. To get help about an Administration Console screen, click the help icon in the upper right corner of the console, as shown in Figure 6-4.

Figure 6-4 Help Icon



Navigating Through the Documentation

When you click the help link, a new browser window opens containing help for the Administration Console. The text displayed in the right frame of the help window describes the functionality of the console screen you are using and contains links to related tasks.

Use the left frame of this window to navigate to other help topics using the Table of **Contents**, the alphabetical **Index**, or the **Search** function.

For list of general topics, click the **Topic List** button and select a topic from the list displayed. The text of that topic appears in the right frame and a table of contents containing links to headings under that topic appears in the left frame.

Use the <<<**Back** or **Fwd**>>> buttons to step through previously viewed pages.

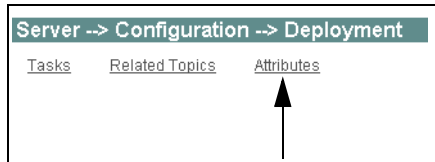
Use the **Print** button to print the current topic.

Help for Configuration Attributes

The console, by default, displays descriptive text for each configuration attribute or control. You can choose whether or not to display this text. For more information, see “Changing Administration Console Behavior” on page 6-15. You can access

additional information, such as default and allowable values for each attribute by clicking the [Attributes](#) link at the top of the right frame in the help window, as shown in Figure 6-5.

Figure 6-5 Attributes Link



Additional BEA Documentation Available on the Internet

Additional documentation is also available on the BEA e-docs Web site. Many help screen include links to related topics on the e-docs Web site. These links are preceded with the label (**e-docs**). An Internet connection is necessary to view this documentation.

The following links display general BEA WebLogic Server documentation:

- [WebLogic Server 8.1 System Administration](#)
- [WebLogic Server 8.1 Documentation](#) includes overviews, programming guides, tutorials, and sample applications.
- [Search](#) the WebLogic Server documentation on the e-docs Web site.

Attributes and Console Screen Reference for Administration Console

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“Console --> Preferences” on page 3-1](#)

[“Console --> Version” on page 2-1](#)

[“Customizing a Table View” on page 5-1](#)

[“Common System Administration Tasks” on page 1-2](#)



Application --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

In the Application > Configuration tab, you configure a new Application to be deployed to WebLogic Server or cluster.

Tasks

“Configuring and Deploying a New Enterprise Application or Web Service” on page 26-2

“Viewing Deployed Enterprise Applications” on page 26-3

“Stopping Deployed Enterprise Applications” on page 26-3

“Adding or Editing Enterprise Application Deployment Notes” on page 26-4

“Deleting an Enterprise Application or Application Module” on page 26-4

“Monitoring Enterprise Applications” on page 26-5

“Viewing and Updating Run-Time Deployment Descriptors” on page 26-5

Related Topics

“Enterprise Applications” on page 26-1

See [Developing WebLogic Server Applications](#).

Attributes

Table 8-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> weblogic.management.configuration.ApplicationMBean</p> <p><i>Attribute:</i> Name</p>	

Table 8-1

Attribute Label	Description	Value Constraints
Path	<p>The location of the original source application files on the Administration Server. Relative paths are based on the root of the Administration Server installation directory. It is highly recommended that you use absolute paths to minimize possible issues when upgrading the server.</p> <p>If the application is not being staged (StagingMode==nostage) then the path must be valid on the target server.</p> <p>The path to an Enterprise application (EAR) is the location of the EAR file or the root of the EAR if it is unarchived. e.g. Path="myapps/app.ear" is valid. If the application is a standalone module, then the path is the parent directory of the module. For example, if the module is located at myapps/webapp/webapp.war, the Path="myapps/webapp" is correct, whereas Path="myapps/webapp/webapp.war" is incorrect.</p> <p>Note: The path displayed in the console may include extra spaces inserted for display purposes. You must remove these spaces if you select and copy this path for use in another context.</p> <p><i>MBean:</i> weblogic.management.configuration.ApplicationMBean</p> <p><i>Attribute:</i> Path</p>	
Deployment Protocol	The deployment protocol to use for this here application.	

Table 8-1

Attribute Label	Description	Value Constraints
Staging Mode	<p>Indicates whether this application is being staged. Staging involves distributing the application files from the Administration server to the targeted managed servers staging directory. This attribute is used to override the managed server's StagingMode attribute.</p> <p><i>MBean:</i> weblogic.management.configuration.ApplicationMBean</p> <p><i>Attribute:</i> StagingMode</p>	
Staging Path	<p>Identifies the directory path on the managed server relative to the server's StagingDirectoryName. This is the path used to prepare and activate an application on a managed server.</p> <p>This attribute is derived from the Path attribute, and depends on whether the application is being staged. If the Path attribute for application myapp is foo.ear, the staging path is set to myapp/foo.ear. If the path is C:\myapp.ear, the staging path is myapp/myapp.ear.</p> <p>If the application is not being staged (StagingMode==nostage), then the staging path is the same as the Path attribute. If this application is not being staged, the staging path is equivalent to source path (Path attribute)</p> <p><i>MBean:</i> weblogic.management.configuration.ApplicationMBean</p> <p><i>Attribute:</i> StagingPath</p>	

Table 8-1

Attribute Label	Description	Value Constraints
Load Order	<p>Specifies the order applications are loaded at server startup. Applications with the lowest values are loaded first.</p> <p>Application ordering is only supported for applications deployed with the 2 phase protocol.</p> <p><i>MBean:</i> weblogic.management.configuration.ApplicationMBean</p> <p><i>Attribute:</i> LoadOrder</p>	<i>Default:</i> 100



Application > Configuration --> Descriptor

[Tasks](#) [Related Topics](#)

Overview

This release of WebLogic Server has deprecated the Administration Console Deployment Descriptor Editor. A new Descriptor tab in the Administration Console has replaced it. Using the Descriptor tab, you can view, modify, and persist deployment descriptor elements to the descriptor file within the Application in the same manner that they were persisted using the Deployment Descriptor Editor.

These descriptor element changes take place dynamically at runtime without requiring redeployment of the application. The descriptor elements contained in the Descriptor tab are limited to only those descriptor elements that may be dynamically changed at runtime.

The Administration Console allows you to modify certain deployment descriptor elements and their attributes for Applications that are deployed as exploded archive files (you cannot edit these descriptors for applications packaged as .EAR archives). If you alter the value of any of these elements and select Apply, then the Application deployment descriptor file is updated and deployed to all of the appropriate server machines.

Tasks

“Configuring and Deploying a New Enterprise Application or Web Service” on page 26-2

“Viewing Deployed Enterprise Applications” on page 26-3

“Stopping Deployed Enterprise Applications” on page 26-3

“Adding or Editing Enterprise Application Deployment Notes” on page 26-4

“Deleting an Enterprise Application or Application Module” on page 26-4

“Monitoring Enterprise Applications” on page 26-5

“Viewing and Updating Run-Time Deployment Descriptors” on page 26-5

Related Topics

“Enterprise Applications” on page 26-1

See [Developing WebLogic Server Applications](#).

Application --> Deploy

Overview

The Application > Deploy tab displays the deployment status of each module in the Enterprise Application.

To deploy or redeploy individual application modules, use the Deploy or Redeploy button next to the module names in the table. You can also use the Deploy All or Redeploy All buttons to manage all application modules shown on this page. (When deploying or redeploying a module, the target server uses the application files located at the path shown in the Configuration tab.)

To stop application modules shown on this page, making them unavailable to clients, use the Stop or Stop All button. Note that when you stop an application module, the source files remain at the path specified in the Configuration tab; you can deploy the modules later without having to reconfigure the application.

(To configure additional deployment targets for this application, click the Targets tab.)

Tasks

“Configuring and Deploying a New Enterprise Application or Web Service” on page 26-2

“Viewing Deployed Enterprise Applications” on page 26-3

“Stopping Deployed Enterprise Applications” on page 26-3

“Adding or Editing Enterprise Application Deployment Notes” on page 26-4

“Deleting an Enterprise Application or Application Module” on page 26-4

“Monitoring Enterprise Applications” on page 26-5

“Viewing and Updating Run-Time Deployment Descriptors” on page 26-5

Related Topics

“Enterprise Applications” on page 26-1

“Deploying Applications and Modules” on page 62-1

[Deploying WebLogic Server Applications](#)

Application --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

In the Application > Notes tab, include any additional information that describes the configuration of the application.

Tasks

“Configuring and Deploying a New Enterprise Application or Web Service” on page 26-2

“Viewing Deployed Enterprise Applications” on page 26-3

“Stopping Deployed Enterprise Applications” on page 26-3

“Adding or Editing Enterprise Application Deployment Notes” on page 26-4

“Deleting an Enterprise Application or Application Module” on page 26-4

“Monitoring Enterprise Applications” on page 26-5

“Viewing and Updating Run-Time Deployment Descriptors” on page 26-5

Related Topics

“Enterprise Applications” on page 26-1

See [Developing WebLogic Server Applications](#).

Attributes

Table 11-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.ApplicationMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

Enterprise Application Deployment Assistant --> Step 2 - Select Targets

[Tasks](#) [Related Topics](#)

Overview

This page displays the available servers and clusters to which you can deploy an Enterprise Application module or Web Service.

To deploy to individual servers, select one or more server instances from the Independent Servers list and click Continue.

To deploy to a cluster of servers, select the name of the cluster from the Clusters list. By default, the assistant deploys an application or module to all server instances in the cluster (the All servers in the cluster option). If you want to deploy to only a subset of the servers in a cluster, select Parts of the cluster and then select the individual server instances to which you want to deploy the application or module.

If you are targeting individual modules of an Enterprise Application, your selections apply only to the module name displayed in the header of the console page (for example, Step 2 - Select targets for module “*module_name*”). If the application has additional modules to deploy, the console re-displays the Select Targets page after you click Continue, allowing you to target the next module to different server instances.

Tasks

“Configuring and Deploying a New Enterprise Application or Web Service” on page 26-2

“Viewing Deployed Enterprise Applications” on page 26-3

“Stopping Deployed Enterprise Applications” on page 26-3

“Adding or Editing Enterprise Application Deployment Notes” on page 26-4

“Deleting an Enterprise Application or Application Module” on page 26-4

“Monitoring Enterprise Applications” on page 26-5

“Viewing and Updating Run-Time Deployment Descriptors” on page 26-5

Related Topics

“Enterprise Applications” on page 26-1

See [Developing WebLogic Server Applications](#).

See [Deploying WebLogic Server Applications](#)

Application --> Targets

[Tasks](#) [Related Topics](#)

In the Application > Targets tab, define the servers or clusters on which you will deploy this application. You can reconfigure deployment targets later if needed.

Tasks

“Configuring and Deploying a New Enterprise Application or Web Service” on page 26-2

“Viewing Deployed Enterprise Applications” on page 26-3

“Stopping Deployed Enterprise Applications” on page 26-3

“Adding or Editing Enterprise Application Deployment Notes” on page 26-4

“Deleting an Enterprise Application or Application Module” on page 26-4

“Monitoring Enterprise Applications” on page 26-5

“Viewing and Updating Run-Time Deployment Descriptors” on page 26-5

Related Topics

“Enterprise Applications” on page 26-1

See [Developing WebLogic Server Applications](#).

See [Deploying WebLogic Server Applications](#)



Enterprise Application Deployment Assistant --> Step 3 - Review Choices and Deploy

[Tasks](#) [Related Topics](#)

Overview

This page displays a list of the targeted clusters and servers for the Enterprise Application or Web Service. Review the entries under the Deployment Targets heading. If you need to change a target, click your browser's Back button.

The Source accessibility header displays the selected staging mode for deploying the Web Application source files:

- **Copy this application onto every target for me**—This option is selected by default if you targeted the application to a cluster or to multiple server instances. This corresponds to “stage” mode where the Administration Server copies the application files to each targeted server; the targeted servers then deploy the application using their copy of the source files.
- **I will make the application accessible from the following location**—This option is selected by default if you targeted the application to a single server instance. This corresponds to the “nostage” mode where the server deploys an application from a single directory; all targeted servers must be able to access the directory to deploy the application. Select this option if you are deploying to a cluster that resides on a single physical machine.

In the Identity header, the Name field specifies a unique name to refer to this Enterprise Application deployment in the Administration Console. Accept the default name or enter a new name to describe the application or Web Service.

Click Deploy to accept the values on this page and deploy the Enterprise Application or Web Service to the listed server instances.

Tasks

“Configuring and Deploying a New Enterprise Application or Web Service” on page 26-2

“Viewing Deployed Enterprise Applications” on page 26-3

“Stopping Deployed Enterprise Applications” on page 26-3

“Adding or Editing Enterprise Application Deployment Notes” on page 26-4

“Deleting an Enterprise Application or Application Module” on page 26-4

“Monitoring Enterprise Applications” on page 26-5

“Viewing and Updating Run-Time Deployment Descriptors” on page 26-5

Related Topics

“Enterprise Applications” on page 26-1

See [Developing WebLogic Server Applications](#).

Enterprise Application Deployment Assistant --> Step 1 - Select Archive

[Tasks](#) [Related Topics](#)

Overview

The Enterprise Application Deployment Assistant helps you deploy a new Enterprise Application or Web Service (in `.ear` format) to one or more servers in the domain. You can select either an archived application (`.ear` file), or an application in exploded archive format.

Use the links in the Location field to browse directories on the Administration Server machine and locate the Enterprise Application or Web Service to deploy. If the application does not reside on the Administration Server machine, first use the upload link to upload the application `.ear` file. This places the application archive in the Administration Server's configured upload directory, and automatically opens that directory in the Location field.

When the assistant detects an `.ear` file or exploded `.ear` directory in the current directory, it lists the archive or directory name as a selection beneath the Location field. Select the name of the archive or directory you want to configure for deployment.

If your domain contains multiple WebLogic Server instances, you have the option to target all components of an Enterprise Application to a single server, or to target different components of the application to different server instances. Click the Target Application or Target Each Module button, respectively.

In a single server domain, click Continue to automatically target the EJB to the available server instance.

Tasks

“Configuring and Deploying a New Enterprise Application or Web Service” on page 26-2

“Viewing Deployed Enterprise Applications” on page 26-3

“Stopping Deployed Enterprise Applications” on page 26-3

“Adding or Editing Enterprise Application Deployment Notes” on page 26-4

“Deleting an Enterprise Application or Application Module” on page 26-4

“Monitoring Enterprise Applications” on page 26-5

“Viewing and Updating Run-Time Deployment Descriptors” on page 26-5

Related Topics

“Enterprise Applications” on page 26-1

See [Developing WebLogic Server Applications](#).

See [Deploying WebLogic Server Applications](#)

Applications

The Applications page displays a list of Enterprise Applications and Web Services (configured as `.ear` files or exploded `.ear` directories) that have been deployed in this domain. After you have deployed an application to one or more servers in the domain, you can later deploy, redeploy, or stop the application by selecting its name on this page.

To deploy a new Enterprise Application or Web Service on servers in this domain, click the Deploy a New Application link.

- For more information about configuring an Enterprise Application or Web Service for deployment, see [“Deploying New Applications and Modules” on page 62-6](#).
- To change the order of automatic deployment for Enterprise Applications or Web Services, see [“Changing the Order of Deployment” on page 62-6](#).



Application Modules

The Application Modules page displays the list of modules that make up the selected Enterprise Application. When deploying the Enterprise Application, you can choose to deploy all of the modules on the same servers, or deploy modules to different servers in the domain..

- For more information about configuring an Enterprise Application for deployment, see [“Deploying New Applications and Modules” on page 62-6](#).
- To change the order of automatic deployment for multiple Enterprise Applications, see [“Changing the Order of Deployment” on page 62-6](#).



Application-Scoped JDBC Connection Pools

--> Testing

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

In the JDBC Connection Pool—Testing tab, you can test a JDBC connection in a connection pool on each server on which the connection pool is deployed.

When you test a connection pool, WebLogic Server reserves and releases a connection from the connection pool.

Note: To make the test more meaningful, make sure that Check On Reserve Enabled or Check On Release Enabled is selected on the Configuration—Descriptor tab. If either of these options is selected, WebLogic Server not only reserves and releases a connection, but also tests the physical database connection. See Check On Reserve Enabled in [Attributes](#).

Tasks

“Testing an Application-Scoped Connection Pool” on page 127-14

Related Topics

“Application-Scoped JDBC Data Sources and Connection Pools” on page 127-9

“Deploying Applications and Modules” on page 62-1

Attributes

Server Name—The name of the server on which the connection pool is deployed. Click the Test Pool button for this server to test a connection from this instance of the connection pool.

Pool State—The current state of the instance of the connection pool: RUNNING, SUSPENDED, or UNAVAILABLE. Each instance of the connection is independent and contains its own physical database connections.

Test Pool—Click this button to test a connection from the selected instance of the connection pool. Make sure that Test Reserved Connections or Test Released Connections is selected on the Configuration—Connections tab (under Advanced Options). Test results are displayed at the top of the pane.

Test pool on all servers—Click this button to test a connection from all instances of the connection pool. Test results are displayed at the top of the pane.

Application-Scoped JDBC Connection Pools

--> Targets

[Tasks](#) [Related Topics](#)

Overview

Use this page to select the servers or clusters on which you would like to deploy this application-scoped connection pool. This connection pool should be deployed on the same targets as other modules in the application that use connections from the pool. Applications look up the data source on the local component environment (`java:comp/env`). The data source must be available on the local server.

Tasks

“Selecting Deployment Targets for an Application-Scoped Connection Pool” on page 127-11

Related Topics

“Deploying Application-Scoped Connection Pools” on page 127-10

“Application-Scoped JDBC Data Sources and Connection Pools” on page 127-9

“Deploying Applications and Modules” on page 62-1



Application-Scoped JDBC Connection Pools

--> Notes

[Tasks](#) [Related Topics](#)

Overview

You can use the Application-Scoped JDBC Connection Pool—Notes tab to store notes about the selected connection pool.

Tasks

“Adding a Note to an Application-Scoped Connection Pool” on page 127-15

Related Topics

“Application-Scoped JDBC Data Sources and Connection Pools” on page 127-9

“Deploying Applications and Modules” on page 62-1



Application-Scoped JDBC Connection Pools

--> Monitoring

[Tasks](#) [Related Topics](#)

Overview

The Application-Scoped Connection Pool—Monitoring tab shows the statistics for each deployed instance of the application-scoped connection pool. You can select which information to display on the page. See “JDBC Connection Pool --> Monitoring” on page 117-1 for a description of default columns displayed.

Tasks

“Monitoring Application-Scoped Connection Pools” on page 127-12

Related Topics

“Application-Scoped JDBC Data Sources and Connection Pools” on page 127-9

“Deploying Applications and Modules” on page 62-1



Application-Scoped JDBC Connection Pools

--> Control

[Tasks](#) [Related Topics](#)

Overview

On the Application-Scoped Connection Pool—Control tab, you can manually control each instance of the application-scoped connection pool, including the following options:

Shrink—Shrinks the connection pool to the maximum of the currently reserved connections or the initial size. Shrinking must be enabled on the Configuration—Descriptor tab or this operation will fail.

Reset—Resets the database connection pool by shutting down and re-establishing all physical database connections. This also clears the statement cache for each connection in the connection pool. You can only reset a normally running connection pool.

Clear Statement Cache—Clears the cache of prepared and callable statements maintained for each connection in the pool.

Tasks

“Shrinking an Application-Scoped Connection Pool” on page 127-13

“Resetting an Application-Scoped Connection Pool” on page 127-13

“Clearing the Statement Cache for an Application-Scoped Connection Pool” on page 127-14

Related Topics

[“Manually Administering an Application-Scoped Connection Pool”](#) on page 127-13

[“Application-Scoped JDBC Data Sources and Connection Pools”](#) on page 127-9

[“Deploying Applications and Modules”](#) on page 62-1

Application-Scoped JDBC Connection Pools

--> Configuration --> Descriptors

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

If you deployed your application as an *exploded* archive, you can view and edit application-scoped connection pool attributes on the Descriptors tab. If you deployed your application as an archived file, you cannot view or change attributes on this page.

Tasks

“Configuring Application-Scoped Connection Pool Attributes” on page 127-10

Related Topics

“Application-Scoped JDBC Data Sources and Connection Pools” on page 127-9

“Deploying Applications and Modules” on page 62-1

“Configuring JDBC Connection Pools” on page 127-1

Attributes

Attribute Label	Description	Value Constraints
Initial Capacity	The number of physical database connections to create when creating the connection pool.	<i>Minimum:</i> 0 <i>Maximum:</i> 2147483647 <i>Default:</i> 1 <i>Dynamic:</i> yes
Max Capacity	Maximum number of physical database connections that this connection pool can contain. Different JDBC Drivers and database servers may limit the number of possible physical connections.	<i>Minimum:</i> 1 <i>Maximum:</i> 2147483647 <i>Default:</i> 15 <i>Dynamic:</i> yes
Capacity Increment	Increment by which the connection pool capacity is expanded. When there are no more available physical connections to service requests, the connection pool will create this number of additional physical database connections and add them to the connection pool. The connection pool will ensure that it does not exceed the maximum number of physical connections as set by <code>MaxCapacity</code> .	<i>Minimum:</i> 1 <i>Maximum:</i> 2147483647 <i>Default:</i> 1 <i>Dynamic:</i> yes
Shrinking Enabled	<p>Indicates whether or not the pool can shrink back to its <code>InitialCapacity</code> when it is detected that connections created during increased traffic are not being used.</p> <p>When shrinking, the number of connections is reduced to the greater of either the initial capacity or the current number of connections in use.</p>	<i>Default:</i> true <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false <i>Dynamic:</i> yes

Attribute Label	Description	Value Constraints
Shrink Frequency Seconds	Number of seconds to wait before shrinking a connection pool that has incrementally increased to meet demand. ShrinkingEnabled must be set to true for a connection pool to shrink.	<i>Units:</i> seconds <i>Minimum:</i> 0 <i>Maximum:</i> 2147483647 <i>Default:</i> 900 <i>Dynamic:</i> yes
Highest Num Unavailable	The maximum number of connections in the pool that can be made unavailable (to an application) for purposes like refreshing the connection, etc. Note that in cases like the backend system being unavailable, this specified value could be exceeded due to factors outside the pool's control. When set to 0 (the default), this feature is disabled.	<i>Minimum:</i> 0 <i>Maximum:</i> 2147483647 <i>Default:</i> 0 <i>Dynamic:</i> yes
Highest Num Waiters	The maximum number of connection requests that can concurrently block waiting to reserve a connection from the pool.	<i>Minimum:</i> 0 <i>Maximum:</i> 2147483647 <i>Default:</i> 2147483647 <i>Dynamic:</i> yes
Check On Create Enabled	When selected (set to true), WebLogic Server tests a connection after creating it and before adding it to the list of connections available to the application. The test adds a small delay in creating the connection, but ensures that the application receives a working connection. TableName must be specified.	<i>Default:</i> false <i>Valid values:</i> <ul style="list-style-type: none"> ■ true ■ false <i>Dynamic:</i> yes
Check On Reserve Enabled	When selected (set to true), WebLogic Server tests a connection before giving it to the application. The test adds a small delay in serving the request for a connection from the pool, but ensures that the application receives a working connection. TableName must be set.	<i>Default:</i> false <i>Valid values:</i> <ul style="list-style-type: none"> ■ true ■ false <i>Dynamic:</i> yes

Attribute Label	Description	Value Constraints
Check On Release Enabled	When selected (set to true), 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. The attribute <code>TableName</code> must be set for Check On Release to be effective.	<i>Default:</i> false <i>Valid values:</i> <ul style="list-style-type: none"> ■ true ■ false <i>Dynamic:</i> yes
Inactive Connection Timeout Seconds	The number of seconds of inactivity after which reserved connections will forcibly be released back into the pool. When set to 0 (the default), this feature is disabled.	<i>Units:</i> seconds <i>Minimum:</i> 0 <i>Maximum:</i> 2147483647 <i>Default:</i> 0 <i>Dynamic:</i> yes
Connection Reserve Timeout Seconds	The number of seconds after which a call to reserve a connection from the pool will timeout. When set to -1, a call will never timeout.	<i>Units:</i> seconds <i>Minimum:</i> -1 <i>Maximum:</i> 2147483647 <i>Default:</i> 10 <i>Dynamic:</i> yes
Test Frequency Seconds	<p>The number of seconds between database connection tests. After every <code>TestFrequencySeconds</code> interval, unused database connections are tested using <code>TableName</code>. Connections that do not pass the test will be closed and reopened to re-establish a valid physical database connection. If the test fails again, the connection is closed.</p> <p>If <code>TableName</code> is not set, the test will not be performed.</p> <p>If set to 0 (the default), connections are not tested.</p>	<i>Units:</i> seconds <i>Minimum:</i> 0 <i>Maximum:</i> 2147483647 <i>Default:</i> 0 <i>Dynamic:</i> yes

Attribute Label	Description	Value Constraints
Connection Creation Retry Frequency Seconds	The frequency of retry attempts by the pool to establish connections to the database. When set to 0 (the default), this feature is disabled.	<i>Units:</i> seconds <i>Minimum:</i> 0 <i>Maximum:</i> 2147483647 <i>Default:</i> 0 <i>Dynamic:</i> yes
Table Name	<p>The name of the table used when testing a physical database connection. The default SQL code used to test a connection is</p> <pre>"select count(*) from TableName"</pre> <p>The <code>TableName</code> 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 <code>TableName</code> to the name of a table that is known to have few rows, or even no rows.</p> <p>If <code>TableName</code> begins with "SQL ", then the rest of the string following that leading token will be taken as a literal SQL statement that will be used to test a connection.</p>	
Init SQL	<p>The SQL code used to initialize a connection. If you specify a value for <code>Init SQL</code>, WebLogic Server will run the query when it creates a database connection. If no value is set for <code>Init SQL</code>, WebLogic Server does not run any SQL code to initialize the connection.</p> <p>Start the code with SQL followed by a space.</p>	

Attribute Label	Description	Value Constraints
Statement Cache Size	The number of Prepared and Callable Statements stored in the cache for further use. WebLogic Server can reuse statements in the cache without reloading them, which can increase server performance. Setting the size of the statement cache to 0 turns it off. Each connection in the pool has its own cache of statements.	<i>Default:</i> 10 <i>Dynamic:</i> yes

Application-Scoped JDBC Connection Pools

--> Configuration --> General

Overview

The configuration—General tab for an application-scoped data source shows the name of the data source as it is listed in the local component environment (`java:comp/env`). This information is read-only. You can change the data source name in the `weblogic-application.xml` supplemental descriptor file.

Related Topics

“Application-Scoped JDBC Data Sources and Connection Pools” on page 127-9

“Deploying Applications and Modules” on page 62-1



Application-Scoped JDBC Connection Pool

--> Deploy

[Tasks](#) [Related Topics](#)

Overview

The Application-Scoped Connection Pool—Deploy tab shows the status of each deployed instance of the application-scoped data source and connection pool. You can stop and redeploy active connection pools and deploy inactive deployed instances of the connection pool.

Tasks

“Stopping and Redeploying an Application-Scoped Connection Pool” on page 127-11

Related Topics

“Application-Scoped JDBC Data Sources and Connection Pools” on page 127-9

“Deploying Applications and Modules” on page 62-1



1 Enterprise Applications

[“Attributes and Console Screen Reference for Applications” on page 27-1]

Overview

An enterprise J2EE application can contain Web Application, EJB, and resource adapter components, deployment descriptors, and archive files. These components are packaged in an Enterprise Archive (EAR) file with an `.ear` extension, or can exist in exploded `.ear` format.

An EAR file contains all JAR, WAR, and RAR component archive files for an application and deployment descriptor that describes the bundled components. The `META-INF/application.xml` deployment descriptor contains an entry for each Web and EJB component, and additional entries to describe security roles and application resources such as databases.

You use the Administration Console or the `weblogic.Deployer` command line utility to deploy an EAR file on one or more WebLogic Server instances and clusters in a domain.

Tasks

Configuring and Deploying a New Enterprise Application or Web Service

The following procedure describes how to use the Administration Console to set the attributes for deploying and configuring a new Enterprise Application. For additional information about configuring and deploying Web Services, refer to “Web Services.”

To deploy a new Enterprise Application or Web Service (archived in an EAR file or exploded EAR format) using the Administration Console:

1. In the left pane of the Console, expand the Deployments folder, right-click Applications, and select Deploy a New Application. This initiates the Deployment Assistant in the right panel.
2. Using the Deployment Assistant, locate the EAR file you would like to configure for use with WebLogic Server. You can also configure an "exploded" Enterprise Application or component directory. Note that WebLogic Server will deploy all components it finds in and below the specified directory.
3. When you have located the archive file to configure, click Target Application.
4. If you have more than one server or cluster in your domain, select the one on which you want to deploy your new Enterprise Application and click Continue. If you have just one server in your domain, go to the next step.
5. Enter a name for the Enterprise Application in the Name field.
If you have more than one server or cluster in your domain, click whether you want to copy the file to each server.
6. Click Deploy. The Console will display the Deploy panel, which lists deployment status and deployment activities for the Enterprise Application.
7. Using the available tabs, enter the following information:
 - Configuration—Define the general configuration of this Enterprise Application.

- **Targets**—Define the servers or clusters on which you would like to deploy all of the modules in this Enterprise Application.
 - **Deploy**—View the deployment status of each module in the Enterprise Application, and to undeploy or redeploy individual modules.
 - **Notes**—Include any additional information that describes the configuration of this Enterprise Application.
8. Click **Apply**.

Viewing Deployed Enterprise Applications

To view a deployed Enterprise Application in the Administration Console:

1. In the left pane of the Console, expand the **Deployments** folder and click on the folder name **Applications**.
2. Use the links in the table to view information about the deployed applications.

Stopping Deployed Enterprise Applications

Stopping an Enterprise Application makes the application unavailable to WebLogic Server clients. Stopping an application does not remove the deployment files from the server; you can later redeploy a stopped application by clicking its redeploy button in the Administration Console.

To stop a deployed Enterprise Application:

1. In the left pane of the Console, expand **Deployments** and click on the folder name **Applications**.
2. In the displayed table, click the name of the Application you wish to stop.
3. On the **Deploy** tab, click **Stop** to stop an individual application module, or **Stop All** to stop all modules in the Enterprise Application.

Adding or Editing Enterprise Application Deployment Notes

1. In the left pane of the Console, expand Deployments and click on the folder name Applications. A table displays all deployed Applications.
2. Click the name of the .ear file for which you will add notes. The Deploy tab displays in the right pane.
3. Select the Notes tab.
4. Add or edit the optional information in the Notes scroll box.
5. Click Apply.

Deleting an Enterprise Application or Application Module

To delete an Enterprise Application, proceed as follows:

1. In the left pane of the Console, expand Deployments and click the folder name Application. A table displays all deployed Enterprise Applications and Application Modules.
2. In this table, locate the Enterprise Application you want to delete. (To delete a module of the application, click the Module link to display a table containing individual Application Modules.)
3. Click the Garbage Can icon to the far right of the Enterprise Application or Application Module.
4. Click Yes to confirm your decision.
5. Click Continue to return to the previous screen.

Monitoring Enterprise Applications

To monitor active Applications in the Console, proceed as follows:

1. In the left pane of the Console, expand Deployments and click the folder name Applications.
2. Under the Modules column, click the link for the application you wish to monitor
3. Click the name of the module you wish to monitor.
4. Select the Monitoring tab.
5. Use the Select Servers pull-down to select the server for which you wish to monitor active applications and application modules.
6. Click Customize This View to customize the available monitoring features. You can choose which items to monitor, determine how to sort these options, and set this view as your default view.
7. Click Apply to save your settings.

Viewing and Updating Run-Time Deployment Descriptors

This release of WebLogic Server deprecates the Administration Console Deployment Descriptor Editor and replaces it with a new Descriptor tab. Using the Descriptor tab, you can view, modify, and persist deployment certain descriptor elements to the descriptor file within an exploded Enterprise Application in the same manner that they were persisted using the Deployment Descriptor Editor.

These descriptor elements are engaged dynamically at runtime without requiring the Application to be redeployed. The descriptor elements contained in the Descriptor tab are limited to only those descriptor elements that may be dynamically changed at runtime.

The Administration Console allows you to modify these deployment descriptor elements and their attributes for applications that are deployed as exploded archive files. (You cannot edit these descriptors for applications packaged as .EAR archives.)

If you alter the value of any of these elements and select Apply, then the Application deployment descriptor file is updated and deployed to all of the appropriate server machines.

To view and edit descriptor information in the Console:

1. In the left pane of the Console, expand Deployments and click the Applications folder name.
2. Click on the name of the application in the Name column.
3. Select the Configuration tab followed by the Descriptor tab.
4. Click on a deployment descriptor link and define the configuration of the deployment descriptor file that is associated with this Application module by changing the provided attribute values as needed.

Attributes and Console Screen Reference for Applications

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“Application --> Configuration --> General” on page 8-1](#)

[“Application > Configuration --> Descriptor” on page 9-1](#)

[“Application --> Deploy” on page 10-1](#)

[“Application --> Notes” on page 11-1](#)

[“Application --> Targets” on page 13-1](#)

[“Applications” on page 16-1](#)

[“Application Modules” on page 17-1](#)

[“ServletRuntime” on page 557-1](#)

[“ServletSessionRuntime” on page 460-1](#)

Deployment Assistant:

[“Enterprise Application Deployment Assistant --> Step 1 - Select Archive” on page 15-1](#)

[“Enterprise Application Deployment Assistant --> Step 2 - Select Targets” on page 12-1](#)

[“Enterprise Application Deployment Assistant --> Step 3 - Review Choices and Deploy” on page 14-1](#)

Application-Scoped JDBC Connection Pools:

[“Application-Scoped JDBC Connection Pools --> Testing” on page 18-1](#)

[“Application-Scoped JDBC Connection Pools --> Targets” on page 19-1](#)

[“Application-Scoped JDBC Connection Pools --> Notes” on page 20-1](#)

“Application-Scoped JDBC Connection Pools --> Monitoring” on page 21-1

“Application-Scoped JDBC Connection Pools --> Control” on page 22-1

“Application-Scoped JDBC Connection Pools --> Configuration --> Descriptors” on page 23-1

“Application-Scoped JDBC Connection Pools --> Configuration --> General” on page 24-1

“Application-Scoped JDBC Connection Pool --> Deploy” on page 25-1

Cluster --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use the Cluster-->Configuration-->General tab to configure cluster-wide configuration parameters.

Tasks

“Configuring a Cluster” on page 41-1

Related Topics

For more information about the attributes you can set on this tab, see the following topics:

- ["Cluster Address"](#) in *Using WebLogic Server Clusters*
- ["Load Balancing Algorithms for RMI Objects and EJBs"](#) in *Using WebLogic Server Clusters*

Attributes

Table 28-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ClusterMBean</code></p> <p><i>Attribute:</i> <code>Name</code></p>	
Cluster Address	<p>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 environments, cluster address should be a DNS host name that maps to the addresses of the Managed Servers in the cluster. For development environments, cluster address may be defined as a comma separated list of single address host names or IP addresses. If network channels are configured, it is possible to set the cluster address on a per channel basis.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ClusterMBean</code></p> <p><i>Attribute:</i> <code>ClusterAddress</code></p>	<i>Configurable:</i> <code>yes</code>

Table 28-1

Attribute Label	Description	Value Constraints
Default Load Algorithm	<p>The algorithm to be used for load-balancing clustered EJBs and RMI objects if no algorithm is specified for a particular object.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ClusterMBean</code></p> <p><i>Attribute:</i> <code>DefaultLoadAlgorithm</code></p>	<p><i>Default:</i> "round-robin"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "round-robin"■ "weight-based"■ "random"■ "round-robin-affinity"■ "weighted-affinity"■ "random-affinity" <p><i>Configurable:</i> yes</p>
WebLogic Plug-In Enabled	<p>Set this attribute to true if the cluster will receive requests from a proxy plug-in or <code>HttpClusterServlet</code>. When <code>WeblogicPluginEnabled</code> is true, a call to <code>getRemoteAddr</code> will return the address of the browser client from the proprietary <code>WL-Proxy-Client-IP</code> header, instead of the web server.</p> <p>For non-clustered servers that will receive proxied requests, this attribute may be set at the server level, on the Server -->Configuration-->General tab.</p> <p><code>WeblogicPluginEnabled</code> is duplicated both in <code>ClusterMBean</code> and <code>ServerMBean</code>. <code>ClusterMBean</code> overrides <code>ServerMBean</code>.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ClusterMBean</code></p> <p><i>Attribute:</i> <code>WeblogicPluginEnabled</code></p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false
Service Age Threshold	<p>The number of seconds by which the age of two conflicting services must differ before one is considered older than the other.</p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 65534</p> <p><i>Default:</i> 180</p> <p><i>Configurable:</i> yes</p> <p><i>Dynamic:</i> yes</p>

Table 28-1

Attribute Label	Description	Value Constraints
Client Cert Proxy Enabled	<p>When set to true for a cluster, this attribute specifies that certs from clients of web applications hosted on the server instances in the cluster are provided in the special WL-Proxy-Client-Cert header sent by a proxy plug-in or HttpClusterServlet.</p> <p>(ClientCertProxyEnabled can be defined at the cluster level, at the server level, and at the web application level, in web.xml.)</p> <p>This setting is useful if user authentication is performed on the proxy server—setting clientCertProxy to true causes the proxy server to pass on the certs to the cluster in a special header, WL-Proxy-Client-Cert.</p> <p>A WL-Proxy-Client-Cert header could be provided by any client with access to WebLogic Server. WebLogic Server takes the certificate information from that header, trusting that it came from a secure source (the plug-in) and uses that information to authenticate the user.</p> <p>For this reason, if you set clientCertProxy to true, use a connection filter to ensure that WebLogic Server accepts connections only from the machine on which the plug-in is running. See “Using Network Connection Filters” in <i>Programming WebLogic Security</i>.</p> <p><i>MBean:</i> weblogic.management.configuration.ClusterMBean</p> <p><i>Attribute:</i> ClientCertProxyEnabled</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false

Cluster --> Configuration --> Multicast

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to configure the address and port upon which the server instances in a cluster broadcast service availability and heart beat messages. It is most efficient to dedicate the multicast address and port to a single cluster, although multiple WebLogic Server clusters can share a single multicast address and port. No other applications should broadcast or subscribe to the multicast address and port used by your cluster or clusters.

Tasks

“Configuring a Cluster” on page 41-1

Related Topics

- ["One-to-Many Communication Using IP Multicast"](#) in *Using WebLogic Server Clusters*
- ["Cluster Multicast Address and Port"](#) in *Using WebLogic Server Clusters*
- ["Configure Multicast Time-To-Live \(TTL\)"](#) in *Using WebLogic Server Clusters*
- ["Configure Multicast Buffer Size"](#) in *Using WebLogic Server Clusters*

Attributes

Table 29-1

Attribute Label	Description	Value Constraints
Multicast Address	The multicast address used by cluster members to communicate with each other. <i>MBean:</i> weblogic.management.configuration.ClusterMBean <i>Attribute:</i> MulticastAddress	<i>Minimum:</i> 224.0.0.0 <i>Maximum:</i> 239.255.255.255 <i>Default:</i> 237.0.0.1 <i>Configurable:</i> yes
Multicast Port	The multicast port used by cluster members to communicate with each other. <i>MBean:</i> weblogic.management.configuration.ClusterMBean <i>Attribute:</i> MulticastPort	<i>Minimum:</i> 1 <i>Maximum:</i> 65535 <i>Default:</i> 7777 <i>Configurable:</i> yes
Multicast Send Delay	The number of milliseconds to delay sending message fragments over multicast in order to avoid OS-level buffer overflow. <i>MBean:</i> weblogic.management.configuration.ClusterMBean <i>Attribute:</i> MulticastSendDelay	<i>Minimum:</i> 0 <i>Maximum:</i> 100 <i>Default:</i> 3 <i>Configurable:</i> yes
Multicast TTL	The time-to-live value for the cluster's multicast address. <i>MBean:</i> weblogic.management.configuration.ClusterMBean <i>Attribute:</i> MulticastTTL	<i>Minimum:</i> 1 <i>Maximum:</i> 255 <i>Default:</i> 1 <i>Configurable:</i> yes
Multicast Buffer Size	The multicast socket send/receive buffer size. <i>MBean:</i> weblogic.management.configuration.ClusterMBean <i>Attribute:</i> MulticastBufferSize	<i>Units:</i> kilobytes <i>Minimum:</i> 64 <i>Default:</i> 64 <i>Configurable:</i> yes

Cluster --> Configuration --> Servers

[Tasks](#) [Related Topics](#)

Overview

Use this tab to configure which Managed Servers belong to a cluster.

Tasks

“Assigning Servers to a Cluster” on page 41-4

Related Topics

- ["Configure Machine Names"](#)



Clusters --> Control

[Tasks](#) [Related Topics](#)

Overview

This page allows you to change the state of Managed Servers in the current cluster. To be able to control Managed Servers using the commands on this page, you must have Node Manager configured and running on the machine that host those Managed Servers

Tasks

“Start all Managed Servers” on page 41-7

“Resume all Managed Servers” on page 41-7

“Graceful Shutdown of all Servers” on page 41-8

“Force Shutdown of all Servers” on page 41-9

“Start/Stop a Server” on page 41-9

“Starting a Managed Server in the STANDBY State” on page 497-12

Related Topics

- [“Configuring, Starting, and Stopping Node Manager”](#) in *Configuring and Managing WebLogic Server*.
- [“Server Lifecycle”](#) in *Configuring and Managing WebLogic Server*



Clusters --> Deployments --> Applications

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This page lists J2EE Applications and Web Services that have been configured for deployment to this cluster. Select the name of a deployed application or Web Service to view the current deployment status, or to deploy, redeploy, or stop the application.

Tasks

[“Configuring and Deploying a New Enterprise Application or Web Service” on page 26-2](#)

[“Deploying a New Web Application” on page 559-2](#)

[“Stopping Deployed Enterprise Applications” on page 26-3](#)

Related Topics

- [“Application Deployment Topics”](#)
- [“Deploy Applications”](#)

Attributes

Table 32-1

Attribute Label	Description	Value Constraints
Name	Name of the application.	
Path	Path to the application.	
Modules	Number of modules.	

Clusters --> Deployments --> Classes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This page lists the startup and shutdown classes that have been configured for deployment to this cluster. Select the name of the class to view the current deployment status, or to deploy, redeploy, or stop the class.

Tasks

[“Configure a Startup or Shutdown Class” on page 525-2](#)

[“Assign a Startup or Shutdown Configuration to Servers or Clusters” on page 525-5](#)

Related Topics

- [“Application Deployment Topics”](#)
- [“Deploy Applications”](#)

Attributes

Table 33-1

Attribute Label	Description	Value Constraints
Name	The name of the configuration.	

Table 33-1

Attribute Label	Description	Value Constraints
Type	Type of class.	
Class Name	The fully qualified name of the class to deploy.	
Arguments	Arguments used to initialize the class.	

Clusters --> Deployments --> Connector

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This page lists the Connector modules that have been configured for deployment to this cluster. Select the name of the Connector module to view the current deployment status, or to deploy, redeploy, or stop the module.

Tasks

[“Deploying New Connector \(Resource Adapter\) Modules” on page 57-2](#)

[“Stopping a Deployed Connectors” on page 57-4](#)

Related Topics

- [“Application Deployment Topics”](#)
- [“Deploy Applications”](#)

Attributes

Table 34-1

Attribute Label	Description	Value Constraints
Name	Name of connector.	

Table 34-1

Attribute Label	Description	Value Constraints
URI	URI of connector.	
Application		

Cluster --> Deployments --> EJB Modules

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This page lists the EJB modules that have been configured for deployment to this cluster. Select the name of the EJB module to view the current deployment status, or to deploy, redeploy, or stop the EJB.

Tasks

[“Configuring an EJB Module” on page 103-3](#)

[“Deploying a New EJB Module” on page 103-2](#)

[“Deploying a New EJB Module” on page 103-2](#)

[“Viewing an EJB Module’s Deployment Status” on page 103-6](#)

[“Stopping or Redeploying an EJB Module” on page 103-7](#)

Related Topics

- ["Application Deployment Topics"](#)
- ["Deploy Applications"](#)

Attributes

Table 35-1

Attribute Label	Description	Value Constraints
Name	Name of the EJB.	
URI	URI of the EJB.	

Clusters --> Deployments --> Web Modules

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This page lists the Web Applications that have been configured for deployment to this cluster. Select the name of the Web Application to view the current deployment status, or to deploy, redeploy, or stop the application.

Tasks

[“Deploying a New Web Application” on page 559-2](#)

[“Stopping Deployed Web Applications” on page 559-4](#)

Related Topics

- [“Application Deployment Topics”](#)
- [“Deploy Applications”](#)

Attributes

Table 36-1

Attribute Label	Description	Value Constraints
Name	Name of the web module.	

Table 36-1

Attribute Label	Description	Value Constraints
URI	URI of the web module.	

Cluster --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use the Notes tab to record comments or notes about a cluster. An example would be to record performance observations with different cluster settings, such as using different Default Load Algorithms.

Tasks

Enter free form text notes to describe your cluster configuration.

Related Topics

[Using WebLogic Server Clusters](#)

Attributes

Table 37-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.ClusterMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes



Active Clusters

The Active Clusters table shows information for all servers assigned to the cluster. You can click column headings in the table to sort the information in the table. You can also click Customize this view to select the columns to display in the table.

- To configure a cluster, see [“Configuring a Cluster” on page 41-1](#).
- To assign a server to a cluster, see [“Assigning Servers to a Cluster” on page 41-4](#).



Cluster

The Clusters page contains a table that displays key information about each WebLogic Server cluster that has been configured for the current domain. On this page, you can:

Click Customize this view to change the cluster attributes that the table displays.

Click the name of a cluster to modify its configuration.

Click Configure a new cluster to create a new cluster.

Click the Clone icon in the table to clone an existing cluster.

- For an introduction to WebLogic Server clusters, information about cluster features, and guidelines for planning and configuring clusters, see *Using WebLogic Server Clusters*.
- For an a overview of the cluster implementation process, see "[Cluster Implementation Procedures](#)" in *Using WebLogic Server Clusters*.



Cluster --> Monitoring

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The Cluster→Monitoring page displays how many Managed Servers are configured for the current cluster, and how many of the configured Managed Servers are currently running. For each Managed Server configured for the cluster, the monitoring table displays the current state of the server instance, and metrics about activity since the server instance was started.

Tasks

[“Assigning Servers to a Cluster” on page 41-4](#)

[“Monitoring a Cluster” on page 41-4](#)

Related Topics

- [Using WebLogic Server Clusters](#)
- [“Monitoring a WebLogic Server Domain”](#) in *Configuring and Managing WebLogic Server*
- [“Understanding Server State”](#) in *Configuring and Managing WebLogic Server*

Attributes

Table 40-1

Attribute Label	Description	Value Constraints
Number of Servers configured for this cluster	The number of servers configured for this Cluster, including inactive servers. <i>MBean:</i> weblogic.management.configuration.ClusterMBean <i>Attribute:</i> ConfiguredServerCount	
Number of Servers currently participating in this cluster	The number of servers currently active in this Cluster. <i>MBean:</i> weblogic.management.configuration.ClusterMBean <i>Attribute:</i> ActualServerCount	
Name	Name of the Managed Server.	
State	Current state of the Managed Server <i>MBean:</i> weblogic.management.runtime.ServerRuntimeMBean <i>Attribute:</i> HealthState	
Servers	Number of active Managed Servers in the cluster. <i>MBean:</i> weblogic.management.ClusterRuntimeMBean <i>Attribute:</i> AliveServersCount	
Resend Requests	The number of state change messages that the current Managed Server resent because a receiving Managed Server missed a message. <i>MBean:</i> weblogic.management.ClusterRuntimeMBean <i>Attribute:</i> ResendRequestsCount	

Table 40-1

Attribute Label	Description	Value Constraints
Fragments Received	Total number of multicast message fragments received by the current Managed Server from other Managed Servers in the cluster. <i>MBean:</i> weblogic.management.ClusterRuntimeMBean <i>Attribute:</i> FragmentsReceivedCount	
Lost Multicast Messages	Total number of incoming multicast messages lost by the current Managed Server. <i>MBean:</i> weblogic.management.ClusterRuntimeMBean <i>Attribute:</i> MulticastMessagesLostCount	
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. <i>MBean:</i> weblogic.management.ClusterRuntimeMBean <i>Attribute:</i> SecondaryDistributionNames	
Primaries	The number of object that the local server hosts. <i>MBean:</i> weblogic.management.ClusterRuntimeMBean <i>Attribute:</i> PrimaryCount	
Machine	Machine upon which the Managed Server runs.	

Table 40-1

Attribute Label	Description	Value Constraints
Known Servers	The names of the active Managed Servers in the cluster. <i>MBean:</i> weblogic.management.ClusterRuntimeMBean <i>Attribute:</i> ServerNames	
Foreign Fragments Dropped	The number of dropped multicast message fragments that originated in foreign domains/cluster that use the same multicast address. <i>MBean:</i> weblogic.management.ClusterRuntimeMBean <i>Attribute:</i> ForeignFragmentsDroppedCount	

1 Clusters

[“Attributes and Console Screen Reference for Clusters” on page 43-1]

A cluster is multiple WebLogic Server instances running simultaneously and working together to provide increased scalability and reliability. A cluster appears to clients to be a single WebLogic Server instance. The server instances that constitute a cluster can run on the same machine, or be located on different machines. You can increase a cluster's capacity by adding additional server instances to it.

For more information, see [Using WebLogic Server Clusters](#).

The following topics describe the cluster configuration and monitoring tasks you can perform with the Administration Console.

Tasks

Configuring a Cluster

There are multiple techniques and tools you can use to create and maintain a cluster configuration. For a list, see "[Methods of Configuring Clusters](#)" in *Using WebLogic Server Clusters*.

If you are configuring a new cluster in a new domain, BEA recommends that you use the using the Configuration Wizard. For more information, see "[Example: Creating a Domain with Administration Server and Clustered Managed Servers](#)" in *Configuring and Managing WebLogic Server*.

Follow these steps to configure a cluster using the Administration Console:

1. In the left pane, click the Clusters node.
2. In the right pane, click Configure a New Cluster.
3. Enter values for:
 - Name—assign a unique name to the cluster. Each configurable resource in your WebLogic Server environment should have a unique name.
 - 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.
 - Default Load Algorithm
 - WebLogic Plug-In Enabled
 - Service Age Threshold
 - Client Cert Proxy Enabled
4. Click Create to create the cluster.
5. Click the Multicast tab.
6. If necessary, edit the default values for:
 - Multicast Address (use a value between 224.0.0.1 and 239.255.255.255)
 - Multicast Port number
 - Multicast Send Delay
 - Multicast TTL
 - Multicast Buffer Size
7. Click Apply.
8. In the right pane, click the Servers tab.
9. In the Available column, select one or more servers to assign to the cluster.
10. Click the right arrow button.
11. Click Apply.

Cloning a Cluster

Clone a cluster to configure a new cluster based on the configuration of an existing cluster. Edit the attributes that must be unique for each cluster in a domain, such as Cluster Name and Cluster Address.

1. In the left pane, click Clusters.
2. Click the Clone icon in the row of the cluster you want to clone.
3. Enter values for:
 - Name—Make sure the name is unique with respect to other configurable resources in the domain.
 - Cluster Address
 - Default Load Algorithm
 - WebLogic Plug-In Enabled
 - Service Age Threshold
 - Client Cert Proxy Enabled
4. Click Clone to create the new cluster.
5. Click the Multicast tab.
6. As appropriate, edit the values for:
 - Multicast Address (use a value between 224.0.0.1 and 239.255.255.255)
 - Multicast Port number
 - Multicast Send Delay
 - Multicast TTL
 - Multicast Buffer Size
7. Click Apply.
8. In the right pane, click the Servers tab.
9. In the Available column, select one or more servers to assign to the cluster.
10. Click the right arrow button.

11. Click Apply.

Deleting a Cluster

1. In the left pane, click Clusters.
2. Click the Delete icon in the row of the cluster you want to delete.
3. At the confirmation question, click Yes.
4. Click Continue.

Assigning Servers to a Cluster

Follow these steps to assign server instances to a cluster.

1. In the left pane, expand Clusters.
2. Click a cluster name that you want to add servers to.
3. In the right pane, click the Servers tab.
4. In the Available column, select one or more servers to assign to the cluster.
5. Click the right arrow button.
6. Click Apply.

Monitoring a Cluster

The Cluster-->Monitoring page displays current status of the cluster, and statistics on the activity of each Managed server in the cluster.

1. In the left pane, expand Clusters.
2. Click a cluster name that you want to monitor.
3. In the right pane, click the Monitoring tab.

4. The Monitoring page displays the number of servers configured for the cluster, the number of servers currently participating in the cluster, and in tabular form, key statistics on each Managed Server's recent activity.

Adding a Note to a Cluster Configuration

1. In the left pane, expand Clusters.
2. Click the name of a cluster to which you want to add a note.
3. Click the Notes tab. Type the note in the Notes field.
4. Click Apply to save your changes.

Specifying a Server's Cluster Replication Group

A cluster is a set of WebLogic Server instances running simultaneously and working together to provide increased scalability and reliability. For more information about cluster replication groups, refer to "[Using Replication Groups](#)."

To specify a server's cluster replication group:

1. In the left pane of the Administration Console, expand the Server node. Then select a server.
2. Click the Configuration tab. Then click the Cluster tab.
3. In the right pane, in the Replication Group field, enter the name of a list of clustered instances for storing session state replicas.
4. In the Preferred Secondary Group field, enter the name of a secondary list of clustered instances for storing session state replicas.
5. Click Apply to save your changes.
6. Restart the server.

Specifying a Server's Cluster Weight

A server's cluster weight determines the proportion of the load the server will bear relative to other servers in a cluster. If all servers have the same weight, they will each bear an equal proportion of the load. If one server has weight 50 and all other servers have weight 100, the 50-weight server will bear half as much as any other server. This algorithm makes it possible to apply the advantages of the round-robin algorithm to clusters that are not homogeneous. For more information, refer to "[Weight-Based Load Balancing](#)."

To specify a server's cluster weight:

1. In the left pane of the Administration Console, expand the Server node. Then select a server.
2. Click the Configuration tab. Then click the Cluster tab.
3. In the right pane, in the Cluster Weight field, enter a value in the range 1–100.
4. Click Apply to save your changes.
5. Restart the server.

Specifying the Address of a Server's NIC Card for Cluster Communication

To specify the address of a server's NIC card for communication in a cluster:

1. In the left pane of the Administration Console, expand the Server node. Then select a server.
2. Click the Configuration tab. Then click the Cluster tab.
3. In the right pane, in the Interface Access field, enter the interface address to handle multicast traffic.
4. Click Apply to save your changes.
5. Restart the server.

Start all Managed Servers

Use this command to start clustered Managed Servers with Node Manager.

Note: To start a Managed Server from the Administration Console using Node Manager, a Node Manager process must be running on each machine that hosts Managed Servers in the cluster. For more information, refer to [“Starting Node Manager”](#) in *Configuring and Managing WebLogic Server*.

To start all of the Managed Servers in a cluster:

1. Start the Administration Server for the domain.
2. In the Administration Console, expand the Clusters node in the left panel.
3. Click the cluster whose members you want to start.
4. Click the **Start all Managed Servers...** link in the right panel.
5. When the Administration Console prompts you to confirm the command, click Yes.
6. As Node Manager starts each Managed Server, the status of the operation is displayed in the Transition Activity table. When an operation is complete for a Managed Server, TASK COMPLETED appears in the Status column.

Resume all Managed Servers

Use this command to resume cluster members in standby state with Node Manager. For information on placing a server in the standby state, refer to [“Starting a Managed Server in the STANDBY State”](#) on page 497-12.

Notes: To control a Managed Server from the Administration Console using Node Manager, a Node Manager process must be running on each machine that hosts Managed Servers in the cluster. For more information, refer to [“Starting Node Manager”](#) in *Configuring and Managing WebLogic Server*.

To resume clustered Managed Servers:

1. Start the Administration Server for the domain.

2. In the Administration Console, expand the Clusters node in the left panel.
3. Click the cluster whose members you want to resume.
4. Click the **Resume all Managed Servers...** link in the right panel.
5. When the Administration Console prompts you to confirm the command, click Yes.
6. As Node Manager resumes each Managed Server, the status of the operation is displayed in the Transition Activity table. When an operation is complete for a Managed Server, TASK COMPLETED appears in the Status column.

Graceful Shutdown of all Servers

Use this command to gracefully shutdown the Managed Servers in a cluster with Node Manager. During a graceful shutdown, WebLogic Server subsystems complete in-flight work and suspend themselves in a specific sequence and in a synchronized fashion. For details, see [“Graceful Shutdown”](#) in *Configuring and Managing WebLogic Server*.

Note: To control a Managed Server from the Administration Console using Node Manager, a Node Manager process must be running on each machine that hosts Managed Server in the cluster. For more information, refer to [“Starting Node Manager”](#) in *Configuring and Managing WebLogic Server*.

To gracefully shutdown clustered Managed Servers:

1. Start the Administration Server for the domain.
2. In the Administration Console, expand the Clusters node in the left panel.
3. Click the cluster whose members you want to shutdown.
4. Click the **Graceful shutdown of all servers...** link in the right panel.
5. When the Administration Console prompts you to confirm the command, click Yes.
6. The right panel displays shutdown settings.
 - If you want to drop all sessions, select Ignore Sessions During Shutdown.

- If you want to set a timeout for the graceful shutdown to complete, enter the timeout period in seconds in the Graceful Shutdown Timeout box.

Click Apply.

7. As the shutdown process proceeds, status is displayed in the Transition Activity table. When an operation is complete for a Managed Server, TASK COMPLETED appears in the Status column.

Force Shutdown of all Servers

Use this command to force shutdown the Managed Servers in a cluster with Node Manager.

Note: To control a Managed Server from the Administration Console using Node Manager, a Node Manager process must be running on each machine that hosts Managed Server in the cluster. For more information, refer to [“Starting Node Manager”](#) in *Configuring and Managing WebLogic Server*.

To force shutdown clustered Managed Servers:

1. Start the Administration Server for the domain.
2. In the Administration Console, expand the Clusters node in the left panel.
3. Click the cluster whose members you want to shutdown.
4. Click the **Force shutdown of all servers...** link in the right panel.
5. When the Administration Console prompts you to confirm the command, click Yes.
6. As the shutdown process proceeds, status is displayed in the Transition Activity table. When an operation is complete for a Managed Server, TASK COMPLETED appears in the Status column.

Start/Stop a Server

Use the Start/Stop button in the Transition Activity table to start a Managed Server that is shutdown, or to gracefully shutdown a Managed Server that is running.

Note: To control a Managed Server from the Administration Console using Node Manager, a Node Manager process must be running on each machine that hosts Managed Server in the cluster. For more information, refer to [“Starting Node Manager”](#) in *Configuring and Managing WebLogic Server*.

To start or stop a Managed Server in the current cluster:

1. Start the Administration Server for the domain.
2. In the Administration Console, expand the Clusters node in the left panel.
3. Click the cluster to which the Managed Server belongs.
4. The Transition Activity table lists the Managed Server in the cluster.
5. Click the **Force shutdown of all servers...** link in the right panel.
6. When the Administration Console prompts you to confirm the command, click Yes.
7. The right panel displays shutdown settings.
 - If you want to drop all sessions, select Ignore Sessions During Shutdown.
 - If you want to set a timeout for the graceful shutdown to complete, enter the timeout period in seconds in the Graceful Shutdown Timeout box.

Click Apply.

8. As the shutdown process proceeds, status is displayed in the Transition Activity table. When an operation is complete for a Managed Server, TASK COMPLETED appears in the Status column.

Cluster --> Protocols --> HTTP

[Related Topics](#) [Attributes](#)

Overview

This page allows you to set HTTP attributes for the default web server for each managed server in a cluster. If you set these attributes, they are applied to the default web server for each managed server in the cluster.

HTTP attributes defined for a specific `WebServerMBean` take precedence over the values set on this page. Note also that the values set on this page apply only to the default webserver—they do not apply to Virtual Hosts. Virtual Host attributes are set in `VirtualHostMBean`.

Related Topics

For information about VirtualHosts, see the Administration Guide, “[Overview of WebLogic Server HTTP Services](#)”.

Attributes

Table 42-1

Attribute Label	Description	Value Constraints
Frontend Host	<p>The HTTP frontendHost is set when the Host information coming from the URL may be inaccurate due to the presence of a firewall or proxy. If this parameter is set, the HOST header is ignored and this value is always used.</p> <p><i>MBean:</i> weblogic.management.configuration.ClusterMBean</p> <p><i>Attribute:</i> frontendHost</p>	Null
Frontend HTTP Port	<p>The frontend HTTP Port is set when the Port information coming from the URL may be inaccurate due to the presence of a firewall or proxy. If this parameter is set, the HOST header is ignored and this value is always used.</p> <p><i>MBean:</i> weblogic.management.configuration.ClusterMBean</p> <p><i>Attribute:</i> frontendHTTPPort</p>	0
Frontend HTTPS Port	<p>The frontend HTTPS Port is set when the Port information coming from the URL may be inaccurate due to the presence of a firewall or proxy. If this parameter is set, the HOST header is ignored and this value is always used.</p> <p><i>MBean:</i> weblogic.management.configuration.ClusterMBean</p> <p><i>Attribute:</i> frontendHTTPSPort</p>	0

Attributes and Console Screen Reference for Clusters

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“Active Clusters” on page 38-1](#)

[“Cluster” on page 39-1](#)

[“Cluster --> Configuration --> General” on page 28-1](#)

[“Cluster --> Configuration --> Multicast” on page 29-1](#)

[“Cluster --> Configuration --> Servers” on page 30-1](#)

[“Clusters --> Control” on page 31-1](#)

[“Clusters --> Deployments --> Applications” on page 32-1](#)

[“Cluster --> Deployments --> EJB Modules” on page 35-1](#)

[“Clusters --> Deployments --> Web Modules” on page 36-1](#)

[“Clusters --> Deployments --> Connector” on page 34-1](#)

[“Clusters --> Deployments --> Classes” on page 33-1](#)

[“Cluster --> Monitoring” on page 40-1](#)

[“Cluster --> Notes” on page 37-1](#)

[“Cluster --> Protocols --> HTTP” on page 42-1](#)



Connector Component --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

In the Connector Component --> Configuration --> General tab, you configure a new Connector component (.rar file) to be deployed to WebLogic Server.

A Connector component (also called a resource adapter) is a system-level software driver used by an application server such as WebLogic Server to connect to an EIS. A resource adapter serves as the “J2EE connector.” The WebLogic J2EE Connector Architecture supports resource adapters developed by Enterprise Information Systems (EISes) vendors and third-party application developers that can be deployed in any application server supporting the Sun Microsystems J2EE Platform Specification, Version 1.3. Resource adapters contain the Java, and if necessary, the native components required to interact with the EIS.

Tasks

“Deploying New Connector (Resource Adapter) Modules” on page 57-2

“Viewing Deployed Connectors” on page 57-3

“Stopping a Deployed Connectors” on page 57-4

“Adding or Editing Connector Deployment Notes” on page 57-4

“Deleting a Connector” on page 57-4

“Monitoring Connectors” on page 57-5

“Viewing and Editing Run-Time Deployment Descriptors” on page 57-5

Related Topics

“Connectors” on page 57-1

See the ["Configuration"](#) chapter in *Programming WebLogic Server J2EE Connectors*.

Attributes

Attribute Label	Description	Value Constraints
Name	<i>Attribute:</i> Name	
Parent.Name	<i>Attribute:</i> Parent.Name	
Deployment Order	<i>Attribute:</i> DeploymentOrder	
Application.Staging Mode	<i>Attribute:</i> Application.StagingMode	

Connector Component --> Configuration --> Descriptor

[Tasks](#) [Related Topics](#)

Overview

This release of WebLogic Server has deprecated the Administration Console Deployment Descriptor Editor. A new Descriptor tab in the Administration Console has replaced it. Using the Descriptor tab, you can view, modify, and persist deployment descriptor elements to the descriptor file within the resource adapter in the same manner that they were persisted using the Deployment Descriptor Editor.

These descriptor element changes take place dynamically at runtime without requiring redeployment of the application. The descriptor elements contained in the Descriptor tab are limited to only those descriptor elements that may be dynamically changed at runtime. These include the following `weblogic-ra.xml` elements.

Note: These elements are maintained in the `weblogic-ra.xml` deployment descriptor file. (For more information, refer to "[weblogic-ra.xml](#)" in *Programming WebLogic Server J2EE Connectors*.) If you alter the value of any of these elements and select Apply, then the Connector deployment descriptor file is updated and deployed to all of the appropriate server machines.

- `initial-capacity`
- `max-capacity`
- `capacity-increment`
- `shrinking-enabled`
- `shrink-frequency-seconds`
- `highest-num-unavailable`
- `highest-num-waiters`

-
- check-on-create-enabled
 - check-on-reserve-enabled
 - check-on-release-enabled
 - inactive-connection-timeout-seconds
 - connection-reserve-timeout-seconds
 - test-frequency-seconds
 - connection-creation-retry-frequency-seconds

Tasks

“Deploying New Connector (Resource Adapter) Modules” on page 57-2

“Viewing Deployed Connectors” on page 57-3

“Stopping a Deployed Connectors” on page 57-4

“Adding or Editing Connector Deployment Notes” on page 57-4

“Deleting a Connector” on page 57-4

“Monitoring Connectors” on page 57-5

“Viewing and Editing Run-Time Deployment Descriptors” on page 57-5

Related Topics

“weblogic-ra.xml” in *Programming WebLogic Server J2EE Connectors*

ConnectorComponent --> Deploy

[Tasks](#) [Related Topics](#)

Overview

In the Connector Component > Deploy tab, view the deployment status of each Connector module. You may also choose to stop, deploy, or redeploy the Connector using the buttons shown beneath the Deployment Status by Target table. (To configure additional deployment targets for this Connector, click the Targets tab.)

Deployment of a resource adapter is similar to deployment of Web Applications, EJBs, and Enterprise Applications. As with these deployment units, you can deploy a resource adapter in an exploded directory format or as an archive file.

Note that WebLogic Server will deploy all components it finds in and below the specified directory.

Tasks

“Deploying New Connector (Resource Adapter) Modules” on page 57-2

“Viewing Deployed Connectors” on page 57-3

“Stopping a Deployed Connectors” on page 57-4

“Adding or Editing Connector Deployment Notes” on page 57-4

“Deleting a Connector” on page 57-4

“Monitoring Connectors” on page 57-5

“Viewing and Editing Run-Time Deployment Descriptors” on page 57-5

Related Topics

“Connectors” on page 57-1

See [*WebLogic Server Deployment and Packaging*](#).

ConnectorComponent --> Monitoring

[Tasks](#) [Related Topics](#)

Overview

In the Connector Component > Monitoring tab, you monitor Connectors modules that are currently active.

Tasks

“Deploying New Connector (Resource Adapter) Modules” on page 57-2

“Viewing Deployed Connectors” on page 57-3

“Stopping a Deployed Connectors” on page 57-4

“Adding or Editing Connector Deployment Notes” on page 57-4

“Deleting a Connector” on page 57-4

“Monitoring Connectors” on page 57-5

“Viewing and Editing Run-Time Deployment Descriptors” on page 57-5

Related Topics

See “Connectors” on page 57-1.

See [Programming WebLogic Server J2EE Connectors](#).



Connector Component --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

In the Connector Component > Notes tab, you enter any additional information that describes the configuration of this Connector module.

Tasks

“Deploying New Connector (Resource Adapter) Modules” on page 57-2

“Viewing Deployed Connectors” on page 57-3

“Stopping a Deployed Connectors” on page 57-4

“Adding or Editing Connector Deployment Notes” on page 57-4

“Deleting a Connector” on page 57-4

“Monitoring Connectors” on page 57-5

“Viewing and Editing Run-Time Deployment Descriptors” on page 57-5

Related Topics

See “Connectors” on page 57-1.

See [Programming WebLogic Server J2EE Connectors](#).

Attributes

Table 48-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.ConnectorComponentMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

Connector Module Deployment Assistant -->

Step 2 - Select Targets

[Tasks](#) [Related Topics](#)

Overview

This page displays the available servers and clusters to which you can deploy the Connector module.

To deploy to individual servers, select one or more server instances from the Independent Servers list and click Continue.

To deploy to a cluster of servers, select the name of the cluster from the Clusters list. By default, the Administration Console deploys a Connector module to all server instances in the cluster (the All servers in the cluster option). If you want to deploy to only a subset of the servers in a cluster, select Parts of the cluster, and then select the individual server instances to which you want to deploy the module.

Click Continue to review your choices and deploy the Resource Adaptor.

Tasks

[“Deploying New Connector \(Resource Adapter\) Modules” on page 57-2](#)

[“Viewing Deployed Connectors” on page 57-3](#)

[“Stopping a Deployed Connectors” on page 57-4](#)

[“Adding or Editing Connector Deployment Notes” on page 57-4](#)

[“Deleting a Connector” on page 57-4](#)

[“Monitoring Connectors” on page 57-5](#)

[“Viewing and Editing Run-Time Deployment Descriptors” on page 57-5](#)

Related Topics

["Configuration"](#) in *Programming WebLogic Server J2EE Connectors*

["weblogic-ra.xml"](#) in *Programming WebLogic Server J2EE Connectors*

Connector Component --> Targets

[Tasks](#) [Related Topics](#)

Overview

In the Connector Component > Targets tab, you select the independent servers, clusters, or virtual hosts on which you would like to deploy the Connector module at server startup time. You can reconfigure deployment targets later if you wish. To deploy or undeploy this Connector module immediately without restarting your server(s), click the Deploy tab.

Tasks

“Deploying New Connector (Resource Adapter) Modules” on page 57-2

“Viewing Deployed Connectors” on page 57-3

“Stopping a Deployed Connectors” on page 57-4

“Adding or Editing Connector Deployment Notes” on page 57-4

“Deleting a Connector” on page 57-4

“Monitoring Connectors” on page 57-5

“Viewing and Editing Run-Time Deployment Descriptors” on page 57-5

Related Topics

See “Connectors” on page 57-1.

See [Programming WebLogic Server J2EE Connectors](#).



Connector Component Deployment Assistant --> Step 3 - Review Choices and Deploy

[Tasks](#) [Related Topics](#)

Overview

This page displays a list of the targeted clusters and servers for the Connector module. Review the entries under the Deployment Targets heading. If you need to change a target, click your browser's Back button.

The Source accessibility header displays the selected staging mode for deploying the Connector module source files:

- **Copy this application onto every target for me**—This option is selected by default if you targeted the Connector module to a cluster or to multiple server instances. This corresponds to “stage” mode where the Administration Server copies the Connector files to each targeted server; and the target servers then deploy the Connector using their copy of the files.
- **I will make the application accessible from the following location**—This option is selected by default if you targeted the Connector module to a single server instance. This corresponds to the “nostage” mode where the server deploys a connector from a single directory; all targeted servers must be able to access the directory to deploy the Connector. Select this option if you are deploying to a cluster that resides on a single physical machine.

In the Identity header, the Name field specifies a unique name to refer to this Connector module deployment in the Administration Console. Accept the default name or enter a new name to describe the Connector.

Click Deploy to accept the values on this page and deploy the Connector module to all specified server instances.

Tasks

“Deploying New Connector (Resource Adapter) Modules” on page 57-2

“Viewing Deployed Connectors” on page 57-3

“Stopping a Deployed Connectors” on page 57-4

“Adding or Editing Connector Deployment Notes” on page 57-4

“Deleting a Connector” on page 57-4

“Monitoring Connectors” on page 57-5

“Viewing and Editing Run-Time Deployment Descriptors” on page 57-5

Related Topics

“Connectors” on page 57-1

See the “[Configuration](#)” chapter in *Programming WebLogic Server J2EE Connectors*.

Connector Module Deployment Assistant -->

Step 1 - Start

[Tasks](#) [Related Topics](#)

Overview

The Connector Module Deployment Assistant helps you to deploy a resource adaptor on one or more servers in the domain. You can select either an archived Connector (.rar file), or a Connector in exploded .rar format.

Use the links in the Location field to browse directories on the Administration Server machine and locate the Connector module to deploy. If the Connector does not reside on the Administration Server machine, first use the upload link to upload the Connector's .rar file. This places the Connector in the Administration Server's configured upload directory, and automatically opens that directory in the Location field.

When the assistant detects a .rar file or exploded .rar directory in the current directory, it lists the archive or directory name as a selection beneath the Location field. Select the name of the archive or directory you want to deploy.

If your domain contains multiple WebLogic Server instances, click Target Module to deploy the Connector to a single server, or to multiple server instances or clusters.

In a single server domain, click Continue to automatically target the Connector to the available server instance.

Tasks

[“Deploying New Connector \(Resource Adapter\) Modules” on page 57-2](#)

[“Viewing Deployed Connectors” on page 57-3](#)

[“Stopping a Deployed Connectors” on page 57-4](#)

[“Adding or Editing Connector Deployment Notes” on page 57-4](#)

[“Deleting a Connector” on page 57-4](#)

[“Monitoring Connectors” on page 57-5](#)

[“Viewing and Editing Run-Time Deployment Descriptors” on page 57-5](#)

Related Topics

[“Configuration” in *Programming WebLogic Server J2EE Connectors*](#)

[“weblogic-ra.xml” in *Programming WebLogic Server J2EE Connectors*](#)

Resource Connectors

Use the Resource Connectors page to deploy a new Connector component (.rar file) to servers in this domain. A Connector component (also called a resource adapter) is a system-level software driver used by an application server such as WebLogic Server to connect to an EIS. A resource adapter serves as the “J2EE connector.” The WebLogic J2EE Connector Architecture supports resource adapters developed by Enterprise Information Systems (EISes) vendors and third-party application developers that can be deployed in any application server supporting the Sun Microsystems J2EE Platform Specification, Version 1.3. Resource adapters contain the Java, and if necessary, the native components required to interact with the EIS.

- For more information, see:
- [“Connectors” on page 57-1](#)
- ["Configuration" in *Programming WebLogic Server J2EE Connectors*.](#)
- ["weblogic-ra.xml" in *Programming WebLogic Server J2EE Connectors*.](#)



Connector Connection Pool Idle/Leaked Connections

[Tasks](#) [Related Topics](#)

Overview

A Connection Leak Profiles column in the Console allows you to view profile information pertaining to leaked connections. This column is not to be confused with the Leaked Connections Detected column, which simply displays the number of leaked connections.

A key difference between these two columns is the Connection Leak Profiles column is controlled by use of the `connection-profiling-enabled` setting in the `weblogic-ra.xml` file. By default, this setting is `false`, so normally the Connection Leak Profiles column will be zero (disabled). However, the Leaked Connections Detected column is always enabled and will always display the number of leaked connections.

A Connection Idle Profiles column in the Console allows you to view profile information pertaining to idle connections. This column is not to be confused with the Idle Connections Detected column, which simply displays the number of idle connections.

A key difference between these two columns is the Connection Idle Profiles column is controlled by use of the `connection-profiling-enabled` setting in the `weblogic-ra.xml` file. By default, this setting is `false`, so normally the Connection Idle Profiles column will be zero (disabled). However, the Idle Connections Detected column is always enabled and will always display the number of idle connections.

Tasks

“Deploying New Connector (Resource Adapter) Modules” on page 1-2

“Viewing Deployed Connectors” on page 1-3

“Stopping a Deployed Connectors” on page 1-4

“Adding or Editing Connector Deployment Notes” on page 1-4

“Deleting a Connector” on page 1-4

“Monitoring Connectors” on page 1-5

“Viewing and Editing Run-Time Deployment Descriptors” on page 1-5

Related Topics

“Connectors” on page 1-1

See the WebLogic Server Connector ["Connection Management"](#) documentation.

ConnectorConnectionPoolRuntime

In addition to the connection management requirements stated in the J2EE Connector Specification, Version 1.0 Final Release, BEA WebLogic Server provides optional settings and services to configure and automatically maintain the size of the connection pool.

- For more information, see the "[Connection Management](#)" chapter in *Programming WebLogic Server J2EE Connectors*.



ConnectorConnectionRuntime

In addition to the connection management requirements stated in the J2EE Connector Specification, Version 1.0 Final Release, BEA WebLogic Server provides optional settings and services to configure and automatically maintain the size of the connection pool.

- For more information, see the "[Connection Management](#)" chapter in *Programming WebLogic Server J2EE Connectors*.



1 Connectors

[“Attributes and Console Screen Reference for Connectors” on page 58-1]

Overview

BEA WebLogic Server continues to build upon the implementation of the Sun Microsystems J2EE Platform Specification, Version 1.3. The J2EE Connector Architecture adds simplified Enterprise Information System (EIS) integration to the J2EE platform. The goal is to leverage the strengths of the J2EE platform—including component models, and transaction and security infrastructures—to address the challenges of EIS integration.

The J2EE Connector Architecture provides a Java solution to the problem of connectivity between the multitude of application servers and EISes. Using the J2EE Connector Architecture, EIS vendors no longer must customize their product for each application server nor does an application server (such as BEA WebLogic Server) that conforms to the J2EE Connector Architecture require custom code in order to support connectivity to a new EIS.

The J2EE Connector Architecture enables an EIS vendor to provide a standard resource adapter (also referred to as a connector) for its EIS; the resource adapter plugs into an application server such as WebLogic Server and provides the underlying infrastructure for the integration between an EIS and the application server.

An application server vendor (BEA WebLogic Server) extends its system only once to support the J2EE Connector Architecture and is then assured of connectivity to multiple EISes. Likewise, an EIS vendor provides one standard resource adapter that can plug in to any application server that supports the J2EE Connector Architecture.

Tasks

For more information on application deployment, refer to “Deploying Applications and Modules” on page 62-1.

Deploying New Connector (Resource Adapter) Modules

To deploy a new Connector (Resource Adapter) module using the WebLogic Server Administration Console:

1. Start the WebLogic Server Administration Console, and select the Domain in which you will be working.
2. In the left pane of the Console, open the Deployments folder.
3. In the left pane of the Console, right-click Connector Modules, and select Deploy a New ConnectorComponent. This initiates the Connector Module Deployment Assistant in the right panel.
4. Use links in the Location field to select the .rar file or exploded .rar directory you want to deploy. If the files do not reside on the Administration Server, use the upload link to upload them to the Administration Server machine.
5. When you have selected the archive file or directory to deploy, click Target Module.
6. If your domain contains multiple WebLogic Server instances, the Deployment Assistant displays a list of server to which you can deploy the Enterprise Application.

Select the individual servers to which you will deploy the new application. If you want to deploy to all server instances in a cluster, select the name of the cluster and ensure that the All servers in the cluster option is chosen. Click Continue when you have selected all servers and clusters.

7. Review the selected deployment targets for the Connector Module. If you need to change a target server, use your browser's Back button.
8. Choose one of the available options from the Source accessibility header:

- Copy this application onto every target for me—This option is selected by default if you targeted the module to a cluster or to multiple server instances. This corresponds to “Stage” mode where the Administration Server copies the module files to each targeted server; and the target servers then deploy the module using their copy of the files.
 - I will make the application accessible from the following location—This option is selected by default if you targeted the module to a single server instance. This corresponds to the “Nostage” mode where the server deploys a module from a single directory; all targeted servers must be able to access the directory to deploy the module.
9. Enter a unique name for the Connector module in the Name field.
 10. Click Deploy. The Console displays the Deploy panel, which lists deployment status and deployment activities for the Enterprise Application.
 11. Click Deploy. The Console will display the Deploy panel, which lists deployment status and deployment activities for the Connector module.
 12. Using the available tabs, enter the following information as necessary:
 - Configuration—Define the general configuration of this Connector module.
 - Targets—Select the independent servers, clusters, or virtual hosts on which you would like to deploy the Connector module at server startup time.
 - Deploy—View the deployment status of this Connector module, and stop, deploy, or redeploy the Connector module.
 - Monitoring—Use this page to monitor the Connector that are currently active.
 - Testing—Test your deployment to see if it was successful.
 - Notes—Include any additional information that describes the configuration of this Connector module.

Viewing Deployed Connectors

To view a deployed Connector in the Administration Console:

1. In the left pane of the Console, expand Deployments and click on the Connector Modules folder name.

2. View a list of deployed Connectors in the table displayed in the right side of the Console.

Stopping a Deployed Connectors

To stop a deployed Connector module from the WebLogic Server Administration Console:

1. In the left pane of the Console, expand Deployments and click on the Connector Modules folder name.
2. Click the Connectors option.
3. In the displayed table, click the name of the Connector you want to stop.
4. Select the Deploy tab.
5. Click Stop or Stop All.

Adding or Editing Connector Deployment Notes

To add or edit Connector deployment notes:

1. In the left pane of the Console, expand the Deployments folder and click on the Connector Modules folder name.
2. In the table showing deployed Connectors, click on the .rar file for which you will add notes.
3. Select the Notes tab.
4. Add or edit the optional information in the Notes scroll box.
5. Click Apply.

Deleting a Connector

To delete a Connector from the domain:

1. In the left pane of the Console, expand Deployments and click the Connector Modules folder name.
2. A table is displayed in the right pane of the Console showing all the deployed Connectors. Select the Configuration tab if it is not displayed.
3. Click the Garbage Can icon to the right of the file you want to delete.
4. Click Yes to confirm your decision.
5. Click Continue to return to the previous screen.

Monitoring Connectors

To monitor active Connectors in the Console:

1. In the left pane of the Console, expand the Deployments folder and right-click on Connector Modules.
2. Select Monitor All Connector Modules to display the Monitoring tab.
3. In the right pane, select the Monitoring tab.
4. View statistics about all of the Connector modules that are deployed on specific server targets. You can customize the presented information by clicking the Customize this view... link.

Viewing and Editing Run-Time Deployment Descriptors

This release of WebLogic Server deprecates the Administration Console Deployment Descriptor Editor and replaces it with a Descriptor tab. Using the Descriptor tab, you can view, modify, and persist certain deployment descriptor elements to the descriptor file within the resource adapter in the same manner that they were persisted using the Deployment Descriptor Editor.

These descriptor elements are initiated dynamically at runtime without requiring the resource adapter to be redeployed. The descriptor elements contained in the Descriptor tab are limited to only those descriptor elements that may be dynamically changed at runtime. These include the following `weblogic-ra.xml` elements:

- initial-capacity
- max-capacity
- capacity-increment
- shrinking-enabled
- shrink-frequency-seconds
- highest-num-unavailable
- highest-num-waiters
- check-on-create-enabled
- check-on-reserve-enabled
- check-on-release-enabled
- inactive-connection-timeout-seconds
- connection-reserve-timeout-seconds
- test-frequency-seconds
- connection-creation-retry-frequency-seconds

The Administration Console allows you to modify these deployment descriptor elements for Connectors that are deployed as exploded archive files (you cannot edit these for applications packaged as .RAR archives). If you alter the value of any of these elements and select Apply, then the Connector deployment descriptor file is updated and deployed to all of the appropriate server machines.

To view and edit descriptor information in the Administration Console:

1. In the left pane of the Console, expand Deployments and click Connector Modules.
2. Click the name of the Connector Module whose descriptor information you want to modify.
3. In the right pane, select Configuration followed by Descriptor.
4. Define the configuration of the application deployment descriptor file that is associated with this Connector module by clicking on the link to the file and changing the provided attribute values as needed.

5. Click Apply to save your changes.

Attributes and Console Screen Reference for Connectors

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“Connector Component --> Configuration --> General” on page 44-1](#)

[“Connector Component --> Configuration --> Descriptor” on page 45-1](#)

[“ConnectorComponent --> Deploy” on page 46-1](#)

[“ConnectorComponent --> Monitoring” on page 47-1](#)

[“Connector Component --> Notes” on page 48-1](#)

[“Connector Component --> Targets” on page 50-1](#)

[“Resource Connectors” on page 53-1](#)

[“ConnectorConnectionPoolRuntime” on page 55-1](#)

[“ConnectorConnectionRuntime” on page 56-1](#)

[“Connector Connection Pool Idle/Leaked Connections” on page 54-1](#)

Deployment Assistant:

[“Connector Module Deployment Assistant --> Step 1 - Start” on page 52-1](#)

[“Connector Component Deployment Assistant --> Step 3 - Review Choices and Deploy” on page 51-1](#)

[“ConnectorComponent --> Deploy” on page 46-1](#)

[“Connector Module Deployment Assistant --> Step 2 - Select Targets” on page 49-1](#)



Install or Update an Application

[Tasks](#) [Related Topics](#)

Overview

Use this page to upload an archive file (a JAR, WAR, RAR, or EAR) from the local machine onto the Administration Server machine. The uploaded file will be placed in the configured staging directory for the Administration Server.

Tasks

[“Setting the Server Staging Mode” on page 62-5](#)

Related Topics

[Packaging and Deploying Applications](#)



Deployment Order

[Tasks](#) [Related Topics](#)

Overview

This page lists all of the applications and modules configured for deployment in this domain, in the order in which they would be deployed at server startup time. The deployment order of modules is determined by their Load Order attribute. Click the Change button next to a displayed module to change it's deployment order.

Tasks

[“Changing the Order of Deployment” on page 62-6](#)

[“Deploying New Applications and Modules” on page 62-6](#)

[“Removing an Application or Module from the Domain” on page 62-9](#)

Related Topics

[Packaging and Deploying Applications](#)



Change Deployment Order

[Tasks](#) [Related Topics](#)

Overview

This page displays the name and staging mode assigned to the application or module, and allows you to change the application's Load Order. Enter a new value in the Load Order field to change the order in which this application or module is deployed at startup.

Modules with a lower Load Order value are deployed before those with a higher value. Modules with the same Load Order value are deployed in alphabetical order using the deployment name.

Tasks

[“Changing the Order of Deployment” on page 62-6](#)

Related Topics

[Packaging and Deploying Applications](#)



1 Deploying Applications and Modules

[“Attributes and Console Screen Reference for Deployment” on page 63-1]

In the Deployment area of the Administration Console, you deploy new applications and modules—J2EE Applications, EJB modules, Web Applications, Connectors, and Startup or Shutdown classes—to servers and clusters in the domain. Deploying a new application or module involves choosing the files to deploy, selecting target servers or clusters to deploy the files, and (if necessary) copying the files to the Administration Server and each target server.

After you have initially setup and deployed an application or module to one or more servers, you can later stop, deploy, or redeploy the application or module without reconfiguring or re-copying its files. The Deployment area of the Administration Console also enables you to do the following tasks:

- Change the order of deployment for applications and modules.
- Change the targeted servers and clusters for applications and modules.
- Remove an application or module from the domains.

Tasks

Preparing Applications and Modules for Deployment

WebLogic Server can deploy applications and modules that are packaged according to the J2EE specifications for Enterprise Applications, Web applications, Web Services, EJB modules, and resource adapters. For each module type, the J2EE specifications and J2EE deployment descriptors define the files required and their location in the directory structure. Modules may also include WebLogic-specific deployment descriptors and, possibly, *container* classes generated with the WebLogic EJB, RMI, or JSP compilers. See [Packaging and Deploying Applications](#) for more information about preparing J2EE application or module files for deployment to WebLogic Server.

Startup and shutdown classes can also be deployed to WebLogic Server instances, but they are not packaged according to the J2EE specifications. You simply identify the class to execute at startup or shutdown.

Archives and Exploded Archives

WebLogic Server enables you to deploy an application or module either as an archive file (`.jar` file), or as an exploded archive file that contains maintains the same directory structure as the `.jar`. Applications and modules archived as `.jar` files must use the correct file extension for the module type:

- EJBs are packaged as `.jar` files.
- Web Applications are packaged as `.war` files.
- Resource Adapters are packaged as `.rar` files.
- Enterprise Applications are packaged as `.ear` files.
- Web Services are similar to Web Applications, but they can be archived either as `.war` or `.ear` files.

An exploded archive directory contains the same files and directories as a `jar` archive. However, the files and directories reside directly in your file system and are not packaged into a single archive file using the `jar` utility.

Exploded archives are frequently used in a development environment, because they allow you to easily recompile or change parts of the deployment without regenerating a `jar` archive. To target an exploded archive, you select the top-level directory of the application or module, rather than the `jar` archive.

Summary of Application and Module Types

The following table summarizes the application and module types you can deploy from the Administration Console, including the key distinguishing deployment files.

Table 62-1 Summary of Deployable Modules

Application or Module	Archive Extension	Key J2EE Deployment Descriptor
Enterprise Application	.ear	META-INF/application.xml
Enterprise JavaBean Module	.jar	META-INF/ejb-jar.xml
Web Application	.war	WEB-INF/web.xml
Web Service	.ear or .war	WEB-INF/web-services.xml
Connector Module	.rar	META-INF/ra.xml
Startup or Shutdown Class	n/a	No deployment descriptor (Class file only)

Selecting a Deployment Staging Mode

The deployment staging mode determines how a module's archive files are made available to server instances that must deploy the module. WebLogic Server provides three different options for staging archive files:

- **Stage mode**—The Administration Server copies the archive files from their source location to a location on each of the targeted Managed Servers that deploy the archive. For example, if you deploy a J2EE Application to three

servers in a cluster, the Administration Server copies the application archive files to each of the three servers. Each server then deploys the J2EE Application using its local copy of the archive files.

Stage mode is the default mode when deploying to more than one WebLogic Server instance.

- **Nostage mode**—The Administration Server does not copy the archive files from their source location. Instead, each targeted server must access the archive files from a single source directory for deployment. For example, if you deploy a J2EE Application to three servers in a cluster, each server must be able to access the same application archive files (from a shared or network-mounted directory) to deploy the application.

Nostage mode is the default mode when deploying only to the Administration Server (for example, in a single-server domain). You can also select nostage mode if you run a cluster of server instances on the same machine.

- **External_stage mode**—External_stage mode is similar to stage mode, in that the deployment files must reside locally to each targeted server. However, the Administration Server does not automatically copy the deployment files to targeted servers in external_stage mode; instead, you must manually copy the files, or use a third-party application to copy the files for you.

In general, if you are deploying to a standalone development server, or if all target servers reside on the same machine, select nostage mode. In either case, all of the target servers can access the same set of files for deployment.

If you are deploying to multiple, remote servers in a production environment, use stage mode. Stage mode ensures that each server has a local copy of the deployment files on hand, even if a network outage makes the Administration Server unreachable. If you do not want the Administration Server to copy the files for you, use external_stage mode instead and ensure that the files are copied before deployment.

Setting the Application Staging Mode

When you deploy an application or module using the Administration Console, the staging mode is configured for the application or module itself. This staging mode overrides any mode configured for the target server(s).

To set the staging mode for an application or module, simply follow the instructions under “Deploying New Applications and Modules” on page 62-6. Note that the Administration Console does not enable you to set the `external_stage` mode at the application level. If you wish to deploy an application or module with in external stage mode, use the `weblogic.Deployer` utility. See [Packaging and Deploying Applications](#) for more information.

Setting the Server Staging Mode

The server staging mode specifies the default deployment mode for a server if none is specified at deployment time. For example, the server staging mode is used if you deploy an application or module using `weblogic.Deployer` and you do not specify a staging mode.

To set the server staging mode:

1. Expand the Servers node in the left pane.
2. Select the name of the server instance that you want to configure.
3. Select the Configuration->Deployment tab in the right pane to display the current staging mode.
4. Select stage, nostage, or external_stage from the Staging Mode menu. These modes correspond to the staging modes described in “Selecting a Deployment Staging Mode” on page 62-3, and apply only to the selected server instance.
5. Enter a path in the Staging Directory Name attribute to store staged deployment files. The path is relative to the root directory of the selected server.
6. If you are configuring the staging mode for the Administration Server, also specify an Upload Directory Name, relative to the server’s root directory. This is the directory where the Administration Server stores uploaded files for deployment to servers and clusters in the domain.
7. Click Apply to change the staging mode and directory.

Deploying New Applications and Modules

The Administration Console provides Deployment Assistants to help you deploy each type of application or module. After you have deployed a new application or module, you can redeploy, stop, and later deploy using Administration Console controls.

To deploy a new application or module using a Deployment Assistant:

1. Expand the Deployments node in the left pane to display the different deployment types.
2. In the left pane, select the type of application or module that you want to deploy. The available deployment types are:
 - Applications—Enterprise Applications or Web Services packaged as `.ear` files or directories
 - EJB Modules—Enterprise JavaBeans
 - Web Application Modules—Web Applications or Web Services packaged as `.war` files or directories
 - Connector Modules—Resource adaptors
 - Startup & Shutdown—Startup classes and Shutdown Classes
3. In the right pane, select the Configure a new *module* link, where *module* is the type of application or module you want to deploy. This initiates the Deployment Assistant for the module.
4. Follow the instructions provided by the deployment assistant to select the deployment files, target the files to individual servers, and deploy the application or module. If you need more help with a particular Deployment Assistant step, click the help icon on the corresponding page for more information.

Changing the Order of Deployment

By default, WebLogic Server deploys Enterprise Applications, EJBs, Web Applications, and Web Services deployed immediately after its subsystems initialize at boot time. Startup classes are deployed and run after application modules have been deployed.

The actual deployment order of modules is determined by their Load Order attribute. By default, new applications and modules are configured with a Load Order value of 100. Modules with a lower Load Order value are deployed before those with a higher value during startup. Modules with the same Load Order value are deployed in alphabetical order using the deployment name.

Follow these steps to view or change the deployment order of modules deployed to the WebLogic Server domain:

1. Select the Deployments node in the left pane. The right pane displays all modules configured for deployment in the domain, listed in their current deployment order.
2. Select the Change button next to a module name to display the Change Deployment Order page.
3. Enter a new value in the Load Order field, and click Apply to apply your changes. The again displays the complete list of modules configured for deployment in the domain.

To change the deployment order of a Startup class, follow the instructions in “Configure a Startup or Shutdown Class” on page 525-2.

Changing the Target Servers for a Deployment

After you have deployed a new application or module to one or more servers, you can add or remove servers from the target list to change the scope of the deployment. To change the list of target servers for a configured application or module:

1. Expand the Deployments node in the left pane to display the different deployment types.
2. In the left pane, select the type of application or module that you want to reconfigure.
3. In the right pane, select the name of the deployment you want to reconfigure.
4. Click the Targets tab in the right pane to display the list of servers and clusters currently assigned to the deployment.
5. Check or uncheck boxes next to individual server names to add or remove the deployment from those servers.

6. Click Apply to apply your changes.

Deploying, Redeploying, and Stopping Applications

After you have deployed a new application or module to one or more servers, you can use the Administration Console to redeploy, stop, or deploy the module files on those servers. To deploy, redeploy, or stop an application or module:

1. Expand the Deployments node in the left pane to display the different deployment types.
2. In the left pane, select the type of application or module that you want to deploy, redeploy, or stop.
3. In the right pane, select the name of the deployment you want to deploy, redeploy, or stop.
4. Click the Deploy tab. The Deploy tab shows the current status of each module contained in the selected deployment. For example, if you selected a J2EE application, the Deploy tab shows the status and controls for each EJB Module and Web Application contained in the J2EE Application deployment.

The available controls (Deploy, Redeploy, or Stop) depend on the current status of the module:

- **Deploy** makes the module available for client use. This control is available when the module is inactive on the specified server.
 - **Redeploy** updates the deployment with the available module files. This control is available when the module is active on the specified server.
 - **Stop** makes the module unavailable to clients. This control is available when the module is active on the specified server.
5. To deploy, redeploy, or stop a single module, click the corresponding control next to the module entry in the table. To deploy, redeploy, or stop all available modules, click the Deploy All, Redeploy All, or Stop All button at the bottom of the table.

Removing an Application or Module from the Domain

A deployed application or module remains available in the domain to redeploy, stop, or deploy, until you explicitly remove it. If you no longer want to deploy an application to servers in the domain, use the following steps to remove it from the domain:

1. Expand the Deployments node in the left pane to display the different deployment types.
2. In the left pane, select the type of application or module that you want to deploy, redeploy, or stop.
3. In the left pane, right-click the name of the deployment you want to remove, and select Delete *module_name...* from the menu.
4. In the right pane, select Yes to remove the application or module.

If you later want to redeploy the removed application or module, follow the instructions in “Deploying New Applications and Modules” on page 62-6 to identify the deployment files and deploy it to servers in the domain.

Attributes and Console Screen Reference for Deployment

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“Install or Update an Application” on page 59-1](#)

[“Deployment Order” on page 60-1](#)

[“Change Deployment Order” on page 61-1](#)

[“Display Deployment Descriptor” on page 4-1](#)



Domain --> Control

[Tasks](#) [Related Topics](#) [Operations](#) [Status](#)

Overview

The Domain—Control tab changes the state of Managed Servers in the domain.

Some of the operations require the Node Manager and the domain-wide administration port.

Tasks

“Starting All Managed Servers in a Domain” on page 497-11

“Starting a Managed Server in the STANDBY State” on page 497-12

“Resuming a Server” on page 497-24

“Shutting Down All Managed Servers in a Domain” on page 497-29

Related Topics

[Overview of Node Manager](#)

[“Enabling the Domain-Wide Administration Port” on page 74-1](#)

[Overview of the Server Lifecycle](#)

Operations

The following table describes the applicability and requirements of each operation:

Operation	Description	Requirements/Restrictions
Start all Managed Servers	Starts all Managed Servers that have been configured for the Node Manager. For each server that has not been configured for the Node Manager, the operation generates an exception. By default, a server instance starts in the <code>RUNNING</code> state, but the Startup Mode setting can change the default behavior. The Startup Mode setting is located on the Servers—Configuration—General tab, under Advanced Options.	Requires the Node Manager. Only available for Managed Servers.
Resume all Managed Servers	Moves all Managed Servers from the <code>STANDBY</code> state to <code>RUNNING</code> .	Requires the administration port to be enabled.
Graceful shutdown all Managed Servers	Gracefully stops all Managed Servers in the domain. New requests are rejected but in-work requests are completed before the server stops.	
Force shutdown of all Managed Servers	Immediately stops all Managed Servers in the domain. In-work requests are dropped, no new requests are accepted, and the server immediately stops.	

Status

Table 64-1

Table Column	Description
Name	The name of a server in the domain. The status table contains one row for each server in the domain.

Table 64-1

Table Column	Description
State	The current lifecycle state of the server. For a description of lifecycle states, refer to Understanding Server State .
Transition Activity	Shows the status of the most recent transition in lifecycle state. For details on the transition, click the link in the Status tab.
Controls	Provides a button that navigates to the server's Control →Start/Stop tab. From the Start/Stop tab, you can start or stop the individual server.



Domain --> Configuration --> Applications

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines auto deployment attributes for all applications in the domain.

Tasks

Related Topics

- “Enterprise Applications” on page 26-1
- See [Developing WebLogic Server Applications](#).

Attributes

Table 65-1

Attribute Label	Description	Value Constraints
Auto Update Interval	The auto-update interface for the application manager.	<i>Default:</i> 3000 <i>Dynamic:</i> yes



Domain --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use the Domain --> Configuration --> General tab to configure attributes for the active domain, including the domain-wide administration port.

The domain-wide administration port enables you to start a WebLogic Server instance in *STANDBY* state. It also allows you to separate administration traffic from application traffic in your domain. Because all servers in the domain must enable or disable the administration port at once, you configure the default administration port settings at the domain level.

Note: Before enabling the domain-wide administration port, you must ensure the SSL and keystore configuration is correct. If one or more servers does not have the correct SSL configuration, you will be unable to start or administer those servers after you enable the administration port. See [“Enabling the Domain-Wide Administration Port” on page 74-1](#) for more information.

Tasks

[Archiving a Domain’s Configuration File](#)

Related Topics

[Overview of WebLogic Server Management](#)

[Configuring Network Resources](#)

["Application Deployment Topics"](#) in *Using WebLogic Server Clusters*

Attributes

Table 66-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.DomainMBean</code></p> <p><i>Attribute:</i> <code>Name</code></p>	
Enable Administration Port	<p>Indicates whether or not the administration port should be enabled for the domain. This will force all the servers in a domain to have the same view of setting up the server's administration port.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.DomainMBean</code></p> <p><i>Attribute:</i> <code>AdministrationPortEnabled</code></p>	<p><i>Default:</i> <code>false</code></p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ <code>true</code>■ <code>false</code>
Administration Port	<p>The common secure administration port for the domain. The administration port uses SSL so all the servers require to have SSL if the administration port is enabled in the domain. If administration port is enabled then each server in the same domain should setup a administration port either using the domain's administration port or overriding it by using the server's administration port The managed server will require to use</p> <p><code>-Dweblogic.management.server=https://admin_server:administration_port</code> to connect to the admin server</p> <p><i>MBean:</i> <code>weblogic.management.configuration.DomainMBean</code></p> <p><i>Attribute:</i> <code>AdministrationPort</code></p>	<p><i>Minimum:</i> <code>1</code></p> <p><i>Maximum:</i> <code>65534</code></p> <p><i>Default:</i> <code>9002</code></p>

Table 66-1

Attribute Label	Description	Value Constraints
Production Mode	<p>Set this attribute to run the servers in the domain in production mode. If not set, servers run in development mode. The operating mode affects subsystem behaviors, such as whether or not the Application Poller is running (ie, only in Development Mode). It also influences what default attribute values will be established.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.DomainMBean</code></p> <p><i>Attribute:</i> <code>ProductionModeEnabled</code></p>	<p><i>Valid values:</i></p> <ul style="list-style-type: none">■ <code>true</code>■ <code>false</code>
Enable Auditing of Administration Changes	<p>Causes the Administration Server to emit log messages when a user changes the configuration or invokes management operations on any resource within a domain.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.DomainMBean</code></p> <p><i>Attribute:</i> <code>AdministrationMBeanAuditingEnabled</code></p>	<p><i>Default:</i> <code>false</code></p> <p><i>Configurable:</i> <code>no</code></p>

Table 66-1

Attribute Label	Description	Value Constraints
Enable Cluster Constraints	<p>When this option is enabled, WebLogic Server enforces a strict two-phase deployment policy for all Managed Servers in a domain—ensuring that application deployment to a cluster succeeds only if all Managed Servers in the cluster are reachable and can deploy the application. If any server instance in the cluster is unreachable, the application is not deployed to any of the Managed Servers.</p> <p>If you do not enable this option, WebLogic Server allows deployment to a partial cluster. When the unreachable Managed Server becomes available, deployment to that server instance will be initiated. Until the deployment process is completed, the Managed Server may experience failures related to missing or out-of-date classes.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.DomainMBean</code></p> <p><i>Attribute:</i> <code>EnforceClusterConstraints</code></p>	<i>Default:</i> not enabled

Advanced Attributes

Table 66-2

Attribute Label	Description	Value Constraints
Console Enabled	Indicates whether the WLS console should be auto-deployed for this domain. <i>MBean:</i> weblogic.management.configuration.DomainMBean <i>Attribute:</i> ConsoleEnabled	<i>Default:</i> true <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false
Console Context Path	Specifies the context path for the WLS console. <i>MBean:</i> weblogic.management.configuration.DomainMBean <i>Attribute:</i> ConsoleContextPath	<i>Default:</i> "console"
Archive Configuration Count	Determines how many versions of the domain's configuration file the Administration Server archives. For more information, refer to " Archiving a Domain's Configuration File ."	<i>Default:</i> 5



Domain --> Configuration --> JTA

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

On the Domain—Configuration—JTA tab in the Administration Console, you can configure Java Transaction API (JTA) attributes to suit your environment. The Administration Console provides default values for all JTA configuration attributes. If you specify an invalid value for any configuration attribute, the WebLogic Server does not boot when you restart it.

Configuration settings for JTA are applicable at the *domain* level. This means that configuration attribute settings apply to all servers within a domain. Monitoring and logging tasks for JTA are performed at the *server* level.

Once you configure WebLogic JTA and any transaction participants, the system can perform transactions using the JTA API and the WebLogic JTA extensions.

Tasks

[“Configuring JTA” on page 237-2](#)

[“Setting the Transaction Log File Location \(Prefix\)” on page 237-11](#)

[“Monitoring a Server Instance from the Administration Console” on page 495-21](#)

Related Topics

Programming WebLogic JTA

Attributes

Table 67-1

Attribute Label	Description	Value Constraints
Timeout Seconds	<p>The transaction timeout in seconds. If the transaction is still in the "active" state after this time (counting from begin()), it is automatically rolled back. Once the transaction moves on to the prepared state, however, this timeout parameter does not apply; the transaction is retried until all the resources are committed.</p> <p><i>MBean:</i> weblogic.management.configuration.JTAMBean</p> <p><i>Attribute:</i> TimeoutSeconds</p>	<p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 30</p> <p><i>Dynamic:</i> yes</p>
Abandon Timeout Seconds	<p>The transaction abandon timeout in seconds. During the second phase of the two-phase commit process, the transaction manager will continue to try to complete the transaction until all resource managers indicate that the transaction is completed. Using the AbandonTimeoutSeconds attribute, you can set the maximum time that a transaction manager will persist in attempting to complete a transaction during the second phase of the transaction. After the abandon transaction timer expires, no further attempt is made to resolve the transaction. If the transaction is in a prepared state before being abandoned, the transaction manager will roll back the transaction to release any locks held on behalf of the abandoned transaction.</p> <p><i>MBean:</i> weblogic.management.configuration.JTAMBean</p> <p><i>Attribute:</i> AbandonTimeoutSeconds</p>	<p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 86400</p> <p><i>Dynamic:</i> yes</p>

Table 67-1

Attribute Label	Description	Value Constraints
Before Completion Iteration Limit	<p>The maximum number of cycles the transaction manager will perform the beforeCompletion synchronization callback. Nothing prevents a Synchronization object from registering another during beforeCompletion, even those whose beforeCompletions have already been called. For example, an EJB can call another in its ejbStore() method. To accomodate this, the transaction manager calls all Synchronization objects, then repeates the cycle if new ones have been registered. This count sets a limit to the number of cycles that can happen.</p> <p><i>MBean:</i> weblogic.management.configuration.JTAMBean</p> <p><i>Attribute:</i> BeforeCompletionIterationLimit</p>	<p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 10</p> <p><i>Dynamic:</i> yes</p>
Max Transactions	<p>The maximum number of simultaneous in-progress transactions allowed on a server.</p> <p><i>MBean:</i> weblogic.management.configuration.JTAMBean</p> <p><i>Attribute:</i> MaxTransactions</p>	<p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 10000</p> <p><i>Dynamic:</i> yes</p>
Max Unique Name Statistics	<p>The maximum number of unique transaction names for which statistics will be maintained. A transaction name typically represents a category of business transactions (such as "funds-transfer")</p> <p><i>MBean:</i> weblogic.management.configuration.JTAMBean</p> <p><i>Attribute:</i> MaxUniqueNameStatistics</p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 1000</p> <p><i>Dynamic:</i> yes</p>

Table 67-1

Attribute Label	Description	Value Constraints
Checkpoint Interval Seconds	<p>Interval at which the transaction manager creates a new transaction log file and checks all old transaction log files to see if they are ready to be deleted. Default is 300 seconds (5 minutes); minimum is 10 seconds; maximum is 1800 seconds (30 minutes).</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JTAMBean</code></p> <p><i>Attribute:</i> <code>CheckpointIntervalSeconds</code></p>	<p><i>Minimum:</i> 10</p> <p><i>Maximum:</i> 1800</p> <p><i>Default:</i> 300</p> <p><i>Dynamic:</i> yes</p>
Forget Heuristics	<p>Whether or not the transaction manager automatically performs an XA Resource <code>forget()</code> operation for all resources reporting a heuristic decision. The default is true; a forget is issued as soon as the transaction learns of a heuristic outcome. Set it to false only if you know what to do with the resource when it reports a heuristic decision.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JTAMBean</code></p> <p><i>Attribute:</i> <code>ForgetHeuristics</code></p>	<p><i>Default:</i> true</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false <p><i>Dynamic:</i> yes</p>

Domain --> Configuration --> Logging

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The Domain—Configuration—Logging tab configures the location, file-rotation criteria, and number of files that the Administration Server uses to store domain log messages.

The domain's log contains messages that individual servers within the domain have forwarded. By default, only the log messages of an `ERROR` or higher severity are forwarded from the local servers to the domain log. To change the default, you must create a domain log filter.

Tasks

“Viewing the Domain Log” on page 253-12

“Rotating Log Files” on page 253-16

“Specifying the Messages That a Server Forwards to the Domain Log” on page 81-1

“Viewing Server Logs” on page 253-9

Related Topics

“Overview of WebLogic Server Log Messages and Log Files” on page 253-2

Attributes

Table 68-1

Attribute Label	Description	Value Constraints
File Name	<p>The name of the file that stores current log messages for the domain log. If the pathname is not absolute, the path is assumed to be relative to the domain directory.</p> <p>To include a time or date stamp in the file name when the log file is rotated, add <code>java.text.SimpleDateFormat</code> variables. Surround each variable with percentage (%) characters.</p> <p>For example,</p> <pre>wl-domain_%yyyy%_%MM%_%dd%_%h h%_mm%.log</pre> <p>If you do not include a time and date stamp, the rotated log files are numbered in order of creation <i>filenamennnnn</i>, where <i>filename</i> is the name configured for the log file.</p> <p><i>MBean</i>: <code>weblogic.management.configuration.LogMBean</code></p> <p><i>Attribute</i>: <code>FileName</code></p>	<p><i>Default</i>: <code>wl-domain.log</code></p> <p><i>Configurable</i>: yes</p>

Table 68-1

Attribute Label	Description	Value Constraints
Rotation Type	<p>Criteria for moving old log messages to a separate file:</p> <ul style="list-style-type: none"> ■ 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 <code>FileMinSize</code>, the server renames the file as <code>FileName.n</code>. ■ By Time. At each time interval that you specify in <code>TimeSpan</code>, the server renames the file as <code>FileName.n</code>. <p>After the server renames a file, subsequent messages accumulate in a new file with the name that you specified in <code>FileName</code>.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.LogMBean</code></p> <p><i>Attribute:</i> <code>RotationType</code></p>	<p><i>Default:</i> none</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ <code>bySize</code> ■ <code>byTime</code> ■ none <p><i>Dynamic:</i> yes</p>
Minimum File Size	<p>The size (1 - 65535 kilobytes) that triggers the server to move log messages to a separate file. After the log file reaches the specified minimum size, the next time the server checks the file size, it will rename the current log file as <code>FileName.n</code> and create a new one to store subsequent messages. (This field is relevant only if you set Rotation Type to By Size.)</p> <p><i>MBean:</i> <code>weblogic.management.configuration.LogMBean</code></p> <p><i>Attribute:</i> <code>FileMinSize</code></p>	<p><i>Units:</i> kilobytes</p> <p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 65535</p> <p><i>Default:</i> 500</p> <p><i>Configurable:</i> yes</p> <p><i>Dynamic:</i> yes</p>

Table 68-1

Attribute Label	Description	Value Constraints
Rotation Time	<p>Determines the start time (hour and minute) for a time-based rotation sequence.</p> <p>At the time that this value specifies, the server renames the current log file as <i>FileName.n</i>. Thereafter, the server renames the log file at an interval that you specify in <i>FileTimeSpan</i>.</p> <p>Use the following format: <i>hh:mm</i>, where <i>hh</i> is the hour in a 24-hour format and <i>mm</i> is the minute.</p> <p>If the time that you specify has already past, then the server starts its file rotation immediately.</p> <p><i>MBean</i>: <code>weblogic.management.configuration.LogMBean</code></p> <p><i>Attribute</i>: <code>RotationTime</code></p>	<p><i>Default</i>: 00:00</p> <p><i>Configurable</i>: yes</p> <p><i>Dynamic</i>: yes</p>
File Time Span	<p>The interval (in hours) at which the server saves old log messages to another file. This value is relevant only you use the time-based rotation type.</p> <p><i>MBean</i>: <code>weblogic.management.configuration.LogMBean</code></p> <p><i>Attribute</i>: <code>FileTimeSpan</code></p>	<p><i>Units</i>: hours</p> <p><i>Minimum</i>: 1</p> <p><i>Default</i>: 24</p> <p><i>Dynamic</i>: yes</p>

Table 68-1

Attribute Label	Description	Value Constraints
Limit Number of Retained Log Files	<p>Limits the number of files that a server creates to store old messages to the maximum number specified in <code>FileCount</code>. After the server reaches this limit, it overwrites the oldest file.</p> <p>If you do not enable this option, the server creates new files indefinitely. You must clean up these files as you require.</p> <p>This value is relevant only if you specify a file rotation type of <code>SIZE</code> or <code>TIME</code>.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.LogMBean</code></p> <p><i>Attribute:</i> <code>NumberOfFilesLimited</code></p>	<p><i>Default:</i> <code>false</code></p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ <code>true</code>■ <code>false</code> <p><i>Dynamic:</i> <code>yes</code></p>
Log Files to Retain	<p>The maximum number of log files that the server creates when it rotates the log. Only valid if <code>isNumberOfFilesLimited</code> is <code>true</code> and <code>setRotationType</code> is either <code>Size</code> or <code>Time</code>.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.LogMBean</code></p> <p><i>Attribute:</i> <code>FileCount</code></p>	<p><i>Default:</i> <code>7</code></p> <p><i>Dynamic:</i> <code>yes</code></p>



Domain --> Configuration --> SNMP

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The WebLogic Administration Server has the ability to function as a Simple Network Management Protocol (SNMP) agent. The WebLogic SNMP agent runs as a service which responds to requests from SNMP managers and sends SNMP trap notifications to SNMP managers. Use this tab to administer SNMP for a domain.

Tasks

“Enabling and Configuring the WebLogic SNMP Agent” on page 516-3

Related Topics

[WebLogic SNMP Management Guide](#)

[WebLogic SNMP MIB Reference](#)

Attributes

Table 69-1

Attribute Label	Description	Value Constraints
Enabled	The state of the SNMP service on an administration server. Select to enable the service. <i>MBean:</i> weblogic.management.configuration.SNMPAgentMBean <i>Attribute:</i> Enabled	<i>Default:</i> false <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false
SNMP Port	The port number on which the WebLogic SNMP agent listens for requests from SNMP managers. Most SNMP managers can ping SNMP agents and some SNMP managers can request the status of specific attributes. <i>MBean:</i> weblogic.management.configuration.SNMPAgentMBean <i>Attribute:</i> SNMPPort	<i>Minimum:</i> 1 <i>Maximum:</i> 65535 <i>Default:</i> 161
MIB Data Refresh Interval	The minimum amount of time all MIB values are cached before the agent attempts to refresh them. <i>MBean:</i> weblogic.management.configuration.SNMPAgentMBean <i>Attribute:</i> MibDataRefreshInterval	<i>Units:</i> seconds <i>Minimum:</i> 30 <i>Maximum:</i> 65535 <i>Default:</i> 120

Table 69-1

Attribute Label	Description	Value Constraints
Server Status Check Interval Factor	<p>Defines a multiplier used to calculate the interval at which the server status is checked.</p> <p>This value is multiplied by the MIB Data Refresh Interval to determine the interval at which server status is checked:</p> $\text{interval} = n * \text{MibDataRefreshInterval}$ <p>The most frequent interval at which server status can be checked is the interval defined for MIB Data Refresh Interval. If you want that interval to be the same as the MIB Data Refresh Interval, enter 1 in the Server Status Check Interval Factor field.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.SNMPAgentMBean</code></p> <p><i>Attribute:</i> <code>ServerStatusCheckIntervalFactor</code> or</p>	<p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 65535</p> <p><i>Default:</i> 1</p>

Table 69-1

Attribute Label	Description	Value Constraints
Community Prefix	<p>The SNMP community (password) that SNMP managers must specify when sending requests to the WebLogic SNMP agent.</p> <p>For more information about the community prefix, refer to "Using Community Names to Specify Target Servers in Management Requests."</p> <p>The default value is <code>public</code>. However, to secure access to the values of the WebLogic attributes, it is recommended that you set Community Prefix to some value other than <code>public</code>.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.SNMPAgentMBean</code></p> <p><i>Attribute:</i> <code>CommunityPrefix</code></p>	<p><i>Default:</i> "public"</p>
Debug Level	<p>Determines whether internal messages are generated indicating what the agent code is doing.</p> <p>The Debug level. Valid values are:</p> <ul style="list-style-type: none">■ 0-NoDebug messages■ 1-Fatal■ 2-Critical■ 3-Non-Critical <p><i>MBean:</i> <code>weblogic.management.configuration.SNMPAgentMBean</code></p> <p><i>Attribute:</i> <code>DebugLevel</code></p>	<p><i>Default:</i> 0</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ 0■ 1■ 2■ 3
Trap Version	<p>Specifies whether this WebLogic Server domain generates traps that conform to SNMPv1 or SNMPv2.</p> <p>Choose the protocol that your SNMP managers support.</p>	

Table 69-1

Attribute Label	Description	Value Constraints
Send Automatic Traps Enabled	<p>Determines whether the WebLogic SNMP agent sends automatically generated traps to SNMP managers.</p> <p>By default, the WebLogic SNMP agent sends traps when servers start and shutdown. For a complete list of events that generate automatic traps, refer to "Automatically Generated WebLogic SNMP Traps."</p>	
Targeted Trap Destinations	<p>WebLogic Server uses a trap destination to specify the SNMP management station and the community name used by the SNMP agent to send trap notifications.</p> <p>Select which trap destination(s) should be used in this WebLogic Server domain from the list of available trap destinations.</p> <p>For information on creating Trap Destinations, refer to "Creating a Trap Destination" on page 516-4.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPAgentMBean</p> <p><i>Attribute:</i> TargetedTrapDestinations</p>	



JCOM --> General

[Related Topics](#)

Overview

Use this page to define the Java classes in your `CLASSPATH` to which COM clients will have access. You may enter the entire `CLASSPATH` or a subset of classes and click Define Policy.

Note: BEA Systems strongly recommends against exposing your entire set of Java classes to COM clients.

Related Topics

[Configuring Access Control](#)



Domain --> Monitor --> Clusters

[Related Topics](#)

Overview

This tab displays key attributes and the current state of a cluster. Click:

- Cluster Name to display more information about the state of the cluster.
- Cluster Status to display status of the Managed Servers in the cluster.

Related Topics

["Monitoring Servers"](#)

["Monitoring Clusters"](#)

["Understanding Server State"](#)



Domain --> Monitor --> Servers

[Related Topics](#)

Overview

This tab displays key attributes and the current state of a server instance. Click Server Name to display the Server-->Configuration-->General tab for the server instance.

Related Topics

["Monitoring Servers"](#)

["Monitoring Clusters"](#)

["Understanding Server State"](#)



Domain --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab stores optional notes to describe a configured WebLogic Server domain.

Tasks

Enter freeform text notes to describe the domain's function or configuration.

Related Topics

[Overview of WebLogic Server Management](#)

Attributes

Table 73-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.DomainMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes



1 Domain

[“Attributes and Console Screen Reference for Domains” on page 76-1]

A *domain* is an inter-related set of WebLogic Server resources that are managed as a unit. A domain includes one or more WebLogic Servers, and may include one or more WebLogic Server clusters.

A domain is a self-contained administrative unit. If an application is deployed in a domain, components of that application cannot be deployed on servers that are not a part of that domain. When a cluster is configured in a domain, all of its servers must be a part of that domain as well.

For each domain, you can configure a subset of Weblogic Server resources that apply to all servers and clusters that reside in the domain. These attributes are configured in the Domain node of the Administration Console.

Tasks

Enabling the Domain-Wide Administration Port

WebLogic Server provides the option to enable an SSL administration port for use with all servers in the domain. The administration port is optional, but it provides two capabilities:

- It enables you to start a server in `STANDBY` state.
- It enables you to separate administration traffic from application traffic in your domain.

The administration port accepts only secure, SSL traffic, and all connections via the port require authentication by a server administrator. Because of these features, enabling the administration port imposes the following restrictions on your domain:

- The Administration Server and all Managed Servers in your domain must be configured with support for the SSL protocol.
- All servers in the domain, including the Administration Server, enable or disable the administration port at the same time.
- After enabling the administration port, you must establish an SSL connection to the Administration Server in order to start any Managed Server in the domain. This applies whether you start Managed Servers manually, at the command line, or using Node Manager.
- After enabling the administration port, all Administration Console traffic *must* connect via the administration port.

To enable the administration port for your domain:

1. First ensure that all servers in the domain are properly configured to use SSL. See [“Configuring Two-Way SSL” on page 428-46](#) for more information.
2. Select the name of the active domain in the left pane to display the domain’s configuration attributes.
3. Click the Configuration tab in the right pane.
4. Click the General tab in the right pane.
5. Select the Enable Domain Wide Administration Port attribute to enable the SSL administration port for this domain.
6. In the Administration Port box, enter the SSL port number that server instances in the domain use as the administration port. You can override an individual server instance’s administration port assignment on the Advanced Options portion of the Configuration->General tab for the server instance.
7. Click Apply to apply your changes.
8. Restart all server instances in the domain, including the Administration Server and all Managed Servers. The Administration Port will not function until all server instances in the domain are restarted.

9. To start Managed Servers after enabling the administration port, you must establish an SSL connection to the domain's Administration Server. You can do this by specifying the following option at the command line when started the Managed Server:

```
-Dweblogic.management.server=https://host:admin_port
```

In this argument, `host` refers to the address of the Administration Server and `admin_port` is the administration port that the Administration Server uses. Note that you must specify the `https://` prefix, rather than `http://`.

Converting the weblogic.properties File

If you have used a version of Weblogic Server earlier than 6.0, you can convert your `weblogic.properties` files into a configuration file (`config.xml`) for a WebLogic Server 8.x domain. The resources defined in a single XML configuration file comprise a single domain.

1. Locate the root directory for the WebLogic Server installation you want to convert. 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.
2. Click the Convert `weblogic.properties` link in the right pane of the Administration Console.
3. 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.
4. 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.
5. Select the Cluster and Server specific `weblogic.properties` directories.
6. Click Convert.
7. Continue to convert your `weblogic.properties` files until you have created the domains needed for your application.
8. See the notes in this section about the following subjects
 - “SSL Security Files” on page 75-1
 - “Servlets” on page 75-2
 - “EJB JAR files and Web App WAR files” on page 75-3

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=ca.pem
weblogic.security.clientRootCA=SecureServerCA.pem
```

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

```
<Server Name=....

  <SSL
ServerCertificateFileName="<NewDomain>/myserver/democert.pem"
ServerKeyFileName="<NewDomain>/myserver/demokey.pem"
ServerCertificateChainFileName="<NewDomain>/myserver/ca.pem"
TrustedCAFileName="<NewDomain>/myserver/SecureServerCA.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 <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 directories:

- <NewDomain>/applications/DefaultWebApp_myserver/WEB-INF directory and will be made as the default webapp, unless there is already a default web app declared in the properties file.
- <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(7.x)` 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 `HelloWorldServlet.class` should be copied into the corresponding `WEB-INF/classes/weblogic/hello` directory. The current directory signifies the directory from which the Weblogic Server(7.x) 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(7.x) 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.



Attributes and Console Screen Reference for Domains

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

“Domain --> Configuration --> Applications” on page 65-1

“Domain --> Configuration --> General” on page 66-1

“Domain --> Configuration --> JTA” on page 67-1

“Domain --> Configuration --> Logging” on page 68-1

“Domain --> Configuration --> SNMP” on page 69-1

“Domain --> Control” on page 64-1

“Domain --> Monitor --> Clusters” on page 71-1

“Domain --> Monitor --> Servers” on page 72-1

“Domain --> Notes” on page 73-1

“JCOM --> General” on page 70-1



Domain Log Filter --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The Domain Log Filter—Configuration tab defines a domain log filter, which modifies the set of messages that one or more servers send to the domain log. By default, all messages of severity `ERROR` or higher are sent.

Note: HTTP requests, JDBC messages, JTA transactions, and messages of severity `DEBUG` are never sent to the domain log, even if you use a filter.

If you are creating a new filter, you must enter information on the Configuration tab and click **Apply** before you can access other tabs. Once you create a filter, you cannot change its name. Instead, you must create a new filter under a different name.

After you create a filter, you activate it by targeting it to one or more servers.

Tasks

“Specifying the Messages That a Server Forwards to the Domain Log” on page 81-1

“Cloning a Domain Log Filter” on page 81-3

“Deleting a Domain Log Filter” on page 81-4

“Specifying Which Messages a Server Sends to Standard Out” on page 253-20

“Viewing the Domain Log” on page 253-12

Related Topics

“Local Log Files and Domain Log Files” on page 253-7

Attributes

Table 77-1

Attribute Label	Description	Value Constraints
Name	<p>An alpha-numeric value that identifies the filter. This name attribute is for your identification purposes only.</p> <p>After you have created a domain log filter, you cannot change its name. Instead, clone the filter and provide a new name for the clone. For more information, refer to “Cloning a Domain Log Filter” on page 81-3.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.DomainLogFilterMBean</code></p> <p><i>Attribute:</i> Name</p>	

Table 77-1

Attribute Label	Description	Value Constraints
Severity Level	<p>The minimum severity of a message that this filter forwards to the domain log. All messages with the specified severity and higher will be sent to the domain log. The ascending order of severities is as follows:</p> <ul style="list-style-type: none"> ■ INFO (64). Used for reporting normal operations. ■ WARNING (32). A suspicious operation or configuration has occurred but it may not have an impact on normal operation. ■ ERROR (16). A user error has occurred. The system or application is able to handle the error with no interruption, and limited degradation, of service. ■ NOTICE (8). A warning message. A suspicious operation or configuration has occurred which may not affect the normal operation of the server. ■ CRITICAL (4). 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 (2). A particular 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 (1). The server is in an unusable state. This severity indicates a severe system failure or panic. <p><i>MBean:</i> weblogic.management.configuration.DomainLogFilterMBean</p> <p><i>Attribute:</i> SeverityLevel</p>	<p><i>Default:</i> weblogic.logging.Severities.WARNING</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ 64 ■ 32 ■ 16 ■ 8 ■ 4 ■ 2 ■ 1

Table 77-1

Attribute Label	Description	Value Constraints
User Ids	<p>The user IDs for which associated messages are sent to the domain log.</p> <p>If no IDs are specified, messages from all user IDs can be sent to the domain log.</p> <p>Every message includes the user ID under which the associated event was executed.</p> <p>To execute some pieces of internal code, WebLogic Server authenticates the ID of the user who initiates the execution and then runs the code under a special Kernel Identity user ID.</p> <p>J2EE modules such as EJBs that are deployed onto a server instance report the user ID that the module passes to the server.</p> <p><i>MBean:</i> weblogic.management.configuration.DomainLogFilterMBean</p> <p><i>Attribute:</i> UserIds</p>	<i>Dynamic:</i> yes
SubSystem Names	<p>A list of subsystems whose messages are selected by this log filter. If none are specified, messages from all subsystems are selected.</p> <p><i>MBean:</i> weblogic.management.configuration.DomainLogFilterMBean</p> <p><i>Attribute:</i> SubSystemNames</p>	

Domain Log Filter --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab stores optional configuration notes for a WebLogic Server domain log filter.

Tasks

Enter text notes to describe the domain's function or configuration.

Related Topics

[“Overview of WebLogic Server Log Messages and Log Files” on page 253-2](#)

Attributes

Table 78-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.DomainLogFilterMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes



Domain Log Filters --> Target

[Tasks](#) [Related Topics](#)

Overview

This tab shows all of the servers that are in the domain. Use it to assign or unassign this domain log filter for one or more servers.

Each server instance can use only one domain log filter. If you select a server that is already using another domain log filter, when you click the Apply button, the Administration Console will replace the older filter assignment with this newer one.

To determine whether a server instance is already using a filter, look on the server's Logging—Domain—Use Log Filter list.

You cannot assign domain log filters to clusters. Instead, you must assign the filters servers individually.

Tasks

[“Specifying the Messages That a Server Forwards to the Domain Log” on page 1-1](#)

Related Topics

[“Overview of WebLogic Server Log Messages and Log Files” on page 1-2](#)



Domain Log Filters

A domain log filter specifies which messages a server sends to the domain log.

To create a new domain log filter, click the [Configure a new Domain Log Filter](#) link. For more information, see “[Specifying the Messages That a Server Forwards to the Domain Log](#)” on page 81-1.

For information on changing the information that this tab displays, refer to “[Customizing Table Views](#)” on page 6-15.



1 Domain Log Filters

[“Attributes and Console Screen Reference for Domain Log Filters” on page 82-1]

In addition to writing messages to its local log file, each WebLogic Server instance broadcasts its log messages. The Administration Server listens for a subset of these messages and stores them in a central *domain log*. For more information about how a server instance broadcasts log messages, refer to "[Overview of WebLogic Logging Services](#)."

By default, the Administration Server listens for only the most important log messages (as determined by Message Severity) that a server instance broadcasts. To change the default, you must create a domain log filter. You can create one domain log filter and assign this filter to all server instances in a domain, or you can create separate domain log filters for each server instance.

The following tasks manage domain log filters:

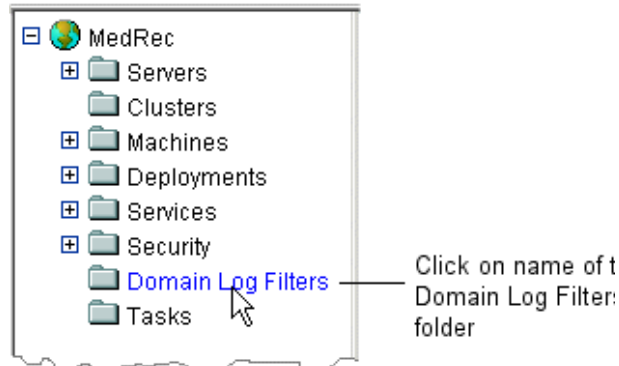
- “Specifying the Messages That a Server Forwards to the Domain Log” on page 81-1
- “Cloning a Domain Log Filter” on page 81-3
- “Deleting a Domain Log Filter” on page 81-4

Specifying the Messages That a Server Forwards to the Domain Log

To specify the messages that a server forwards to the domain log:

1. In the left pane of the Administration Console, click on the name of the Domain Log Filters folder. (See Figure 81-1.)

Figure 81-1 Click on the Domain Log Filters Node



The Domain Log Filters table displays all the log filters in the domain.

2. In the right pane, click the Configure a New Domain Log Filter text link.
3. On the Create a new DomainLogFilter page, in the Name field, enter a value to identify the filter.
4. Select a severity level from the Severity Level drop-down list.

Only messages of the selected severity or higher are forwarded to the domain log. You cannot forward `DEBUG` messages to the domain log. For information about message severity, refer to “Message Severity” on page 253-5.
5. Enter the User IDs to associate with this log filter, if any. If no User IDs are chosen, the filter sends messages that are generated by any user.

All messages that the WebLogic logging services generate within a WebLogic Server JVM include the User ID under which the associated event was executed. J2EE modules such as EJBs that are deployed onto a server instance report the user ID that the module passes to the server. To execute some pieces of internal code, WebLogic Server authenticates the ID of the user who initiates the execution and then runs the code under a special Kernel Identity user ID.

Log messages that are generated within a client JVM client do not include this field.

6. Select the Subsystems for which you want messages forwarded to the domain log. If no subsystems are chosen, the filter sends messages from all subsystems.
7. Click Create.

-
8. To assign the filter to one or more servers, select the Targets tab.
 9. On the Targets tab, select the servers that you want to use this filter.

Note: Each server instance can use only one domain log filter. If you select a server that is already using another domain log filter, when you click the Apply button, the Administration Console will replace the older filter assignment with this newer one.

To determine whether a server instance is already using a filter, look on the server's Logging—Domain—Use Log Filter list.

10. Click Apply.


Cloning a Domain Log Filter

If you want to create a domain log filter that is similar to an existing domain log filter, you can clone the existing filter and then modify it.

To clone an existing domain log filter:

1. In the left pane of the Administration Console, click on the name of the Domain Log Filters folder. (See Figure 81-1.)

The Domain Log Filters table displays all the log filters in the domain.


2. In the Domain Log Filters table, click the Clone icon  in the row of the domain log filter you want to clone.
3. On the Clone page, change the attribute values for the new domain log filter as necessary, using the instructions in "[Specifying the Messages That a Server Forwards to the Domain Log](#)."
4. Click Clone to save your changes.

To use this filter, you must assign it to one or more servers as describe in step 8. in "[Specifying the Messages That a Server Forwards to the Domain Log](#)".

Deleting a Domain Log Filter

1. In the left pane of the Administration Console, click on the name of the Domain Log Filters folder. (See Figure 81-1.)

The Domain Log Filters table displays all the log filters in the domain.

2. Click the Delete icon  in the row of the domain log filter you want to delete.
3. Click Yes to delete the domain log filter.

Attributes and Console Screen Reference for Domain Log Filters

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“Domain Log Filters” on page 80-1](#)

[“Domain Log Filter --> Configuration” on page 77-1](#)

[“Domain Log Filters --> Target” on page 79-1](#)

[“Domain Log Filter --> Notes” on page 78-1](#)



EJB --> Configuration --> Descriptors

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The top portion of this page lists the Enterprise JavaBeans (EJBs) in this module, *if this module has been deployed as exploded*. To display and update selected deployment elements for an exploded EJB, click the EJB name. An archived module's deployment descriptor elements cannot be displayed or updated.

Tasks

[Configuring Deployment Descriptor Values](#)

Related Topics

["Edit Deployment Descriptors"](#) in *Programming WebLogic Server EJBs*.

Attributes

The following table summarizes, by EJB type, the WebLogic-specific deployment descriptor elements whose values you can update for an exploded EJB module:

EJB Type	Deployment Descriptor Elements
Entity	<ul style="list-style-type: none">■ Max Beans in Cache■ Max Beans in Free Pool■ Key Cache Size■ Transaction Timeout Seconds■ Idle Timeout Seconds (read-only entity beans only)
Message-Driven	<ul style="list-style-type: none">■ Max Beans in Free Pool■ Message Selector■ JMSPolling Interval Seconds■ Transaction Timeout Seconds
Stateless	<ul style="list-style-type: none">■ Max Beans in Free Pool■ Transaction Timeout Seconds
Stateful	<ul style="list-style-type: none">■ Max Beans in Free Pool■ Idle Timeout Seconds■ Transaction Timeout Seconds

For element descriptions, see "[weblogic-ejb-jar.xml Deployment Descriptor Reference](#)" in Programming WebLogic EJBs.

EJB --> Configuration --> Compiler options

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

You can configure EJB compiler options in the Configuration tab of the Administration Console. The `weblogic.appc` and `weblogic.ejbcc` utilities generate and compile container classes if you want to compile the `.ear` or `.jar` file for deployment into the EJB container. Although you can configure options for this utility at the command line, you may find it more convenient to configure the more commonly used options in the Administration Console.

Tasks

“Configuring Compiler Options” on page 103-5

Related Topics

[*WebLogic Server Deployment and Packaging Guide*](#)

["webLogic.appc"](#) in *Programming WebLogic Enterprise JavaBeans*

Attributes

Table 84-1

Attribute Label	Description	Value Constraints
Java Compiler	<p>The path to the Java compiler to use to compile EJB's (e.g. "sj" or "javac"). Note: the default for this attribute must be null. If no JavaCompiler is specified on this specific EJBComponent, the default will be pulled in the following order from - EJBContainerMBean - Server.JavaCompiler.</p> <p><i>MBean:</i> weblogic.management.configuration.EJBComponentMBean</p> <p><i>Attribute:</i> JavaCompiler</p>	<i>Default:</i> null
Extra RMI Compiler Options	<p>The extra options passed to the RMI compiler during server-side generation. Note: the default for this attribute must be null. If no extra RMI compiler options are specified on the EJBComponent, the default will be pulled from the server MBean.</p> <p><i>MBean:</i> weblogic.management.configuration.EJBComponentMBean</p> <p><i>Attribute:</i> ExtraRmicOptions</p>	<i>Default:</i> null
Extra EJB Compiler Options	<p>The extra options passed to the EJB compiler during server-side generation. Note: the default for this attribute must be null. If no extra EJB compiler options are specified on the EJBComponent, the default will be pulled from Server.ExtraEJBOptions.</p> <p><i>MBean:</i> weblogic.management.configuration.EJBComponentMBean</p> <p><i>Attribute:</i> ExtraEJBOptions</p>	<i>Default:</i> null

Table 84-1

Attribute Label	Description	Value Constraints
Keep Generated Source Files	<p>Return true if the EJB compiler should keep its generated source files, false if it should delete them after compiling them.</p> <p><i>MBean:</i> weblogic.management.configuration.EJBComponentMBean</p> <p><i>Attribute:</i> KeepGenerated</p>	<p><i>Default:</i> true</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false
Force Generation	<p>Return true if the EJB compiler should force regeneration of wrapper classes, false if it should regenerate the files only if it determines it needs to do so.</p> <p><i>MBean:</i> weblogic.management.configuration.EJBComponentMBean</p> <p><i>Attribute:</i> ForceGeneration</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false



EJB --> Configuration --> General

[Tasks](#) [Attributes](#)

Overview

This page displays an EJB's name, path, staging mode and load order.

Tasks

“Displaying and Configuring General Information” on page 103-3

Attributes

Table 85-1

Attribute Label	Description	Value Constraints
Name	The name of this EJB. WebLogic Server uses an MBean to implement and persist the configuration. <i>MBean:</i> weblogic.management.configuration.EJBComponentMBean <i>Attribute:</i> Name	
Path	The pathname to the .jar. <i>MBean:</i> weblogic.management.configuration.EJBComponentMBean <i>Attribute:</i> Path	
Load Order	Specifies the order in which a standalone EJB is loaded at server startup. EJBs with the lowest values are loaded first. EJB ordering is only supported for EJBs deployed with the 2 phase protocol. <i>MBean:</i> weblogic.management.configuration.ApplicationMBean <i>Attribute:</i> LoadOrder	<i>Default:</i> 100
Staging Mode	Indicates whether this standalone EJB is being staged. Staging involves distributing the application files from the administration server to the targeted managed servers staging directory. This attribute is used to override the managed server's StagingMode attribute. <i>MBean:</i> weblogic.management.configuration.ApplicationMBean <i>Attribute:</i> StagingMode	<i>Default:</i> null <i>Valid values:</i> <ul style="list-style-type: none">■ nostage■ stage■ external_stage





EJB --> Deploy

Tasks

Overview

This page displays an EJB's deployment status, and allows you to stop or redeploy the EJB.

To configure additional deployment targets for this EJB module before redeploying, click the Targets tab.

Tasks

“Viewing an EJB Module's Deployment Status” on page 103-6

“Stopping or Redeploying an EJB Module” on page 103-6

“Setting an EJB Module's Target Server and/or Cluster” on page 103-7



EJB --> Monitoring

Choose the associated tab to monitor stateful session, stateless session, entity or message-driven EJBs.



Deployments-->EJB-->Monitoring--> Stateless EJBs

[Run-Time Statistics for Stateless Session EJBs](#) [Raw Monitoring Data for Stateless Session EJBs](#) [Tasks](#) [Related Topics](#)

Overview

This page displays run-time statistics for stateless session EJBs in the currently selected archive. You can display statistics for one or more server instances on which the EJB is deployed. The statistics are defined in [“Run-Time Statistics for Stateless Session EJBs”](#).

You can tailor this console page to display some or all of the underlying counts from which the statistics are calculated, by clicking [Customize this View](#), and selecting the desired data from the list of available columns. The available data is defined in [“Raw Monitoring Data for Stateless Session EJBs”](#).

The statistics and underlying counts reflect activity since the bean was last deployed.

Run-Time Statistics for Stateless Session EJBs

The following table defines run-time statistics displayed for stateless session EJBs.

Table 88-1 Stateless Session EJB Run-Time Statistics

Statistic	Description
Pool Miss Ratio	<p>The pool miss ratio is a ratio of the number of times a request was made to get a bean from the pool when no beans were available, to the total number of requests for a bean made to the pool. The consequence of a pool miss is different for different types of beans.</p> <p>A pool miss for a stateless session bean will cause the requesting thread to wait for a bean to become available in the pool. The maximum time a thread will wait is equal to the transaction timeout value for the bean.</p> <p>Entity beans and message-driven beans will never wait for an instance to become available. Instead, a pool miss will cause the pool to create a new bean instance to service the request. Pool misses come at a cost since the executing thread will either have to wait for a bean to become available or have to wait for a new bean to be created. As such, it is best to try to keep your pool miss ratio to a minimum.</p> <p>For information on what to tune in response to the pool miss ratio statistic, see “Pool Miss Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>
Destroyed Bean Ratio	<p>The destroyed bean ratio is a ratio of the number of beans destroyed to the total number of requests for a bean. The EJB specification mandates that the EJB container destroys a bean when non-application exceptions are thrown from the bean during execution. Destroying beans comes at a cost, however, because destroyed beans will likely have to be replaced with new bean instances. As a result, you should keep your destroyed bean ratio to a minimum.</p> <p>For information on what to tune in response to the destroyed bean ratio statistic, see “Destroyed Bean Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>
Pool Timeout Ratio	<p>The pool timeout ratio is a ratio of requests that have timed out waiting for a bean from the pool to the total number of requests made. This ratio is only valid for stateless session beans because it is the only type of bean that will wait for a bean to become available.</p> <p>Other types of beans will automatically create a new instance to service a request rather than waiting. For best performance, the pool timeout ratio should be as small as possible.</p> <p>For information on what to tune in response to the pool timeout ratio statistic, see “Pool Timeout Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>

Statistic	Description
Transaction Rollback Ratio	<p>The transaction rollback ratio is the ratio of transactions that have rolled back to the number of total transactions involving the EJB. 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. It may also be useful in gauging 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.</p> <p>For information on what to tune in response to the transaction rollback ratio statistic, see “Transaction Rollback Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>
Transaction Timeout Ratio	<p>The transaction timeout ratio is the ratio of transactions that have timed out to the total number of transactions involving an EJB. Timeouts can be especially concerning because they are a signal of inefficiency.</p> <p>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 vein. The transaction timeout ratio is a good indicator of a problem with an application.</p> <p>For information on what to tune in response to the transaction timeout ratio statistic, see “Transaction Timeout Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>

Raw Monitoring Data for Stateless Session EJBs

This section defines the underlying run-time counts from which the statistics in “Run-Time Statistics for Stateless Session EJBs” are calculated. To display the counts on the EJB --> Monitoring --> Stateless EJBs tab page, select Customize this View and choose from the list of available data.

Table 88-2 Stateless Session EJB Raw Monitoring Data

Run-Time Count	Description
Access Total Count	Displays the total number of times an attempt was made to get an instance from the free pool. This information is useful for giving context to the other free pool counts.
Beans In Use Current Count	Displays the number of bean instances 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.
Destroyed Total Count	
EJB Name	
Miss Total Count	Displays the total number of times a failed attempt was made to get an instance from the free pool. An Attempt to get a bean from the pool will fail if there are no available instances in the pool. This information is useful for calculating the pool miss ratio.
Pool Timeout Total Count	Displays the total number of Threads that have timed out waiting for an available bean instance from the free pool. This information is useful for calculating the pool timeout ratio.
Pool Waiter Total Count	
Pooled Beans Current Count	Displays the current number of available bean instances in 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.
Server	
Transactions Committed Total Count	
Transactions Rolled Back Total Count	

Run-Time Count	Description
Transactions Timed out Total Count	
Waiter Current Count	Displays the number of threads currently waiting for an available bean instance from the free pool. This information may be useful, for example, for investigating the cause of poor application performance at a particular time.

Tasks

“Monitoring EJBs” on page 103-8

Related Topics

[“Tuning WebLogic Server EJBs”](#) in *WebLogic Server Performance and Tuning*



Deployments-->EJB --> Monitoring --> Stateful Session EJBs

[Run-Time Statistics for Stateful Session EJBs](#) [Raw Monitoring Data for Stateful Session EJBs](#) [Tasks](#) [Related Topics](#)

Overview

This page displays run-time statistics for stateful session EJBs in the currently selected archive. You can display statistics for one or more server instances on which the EJB is deployed. The statistics are defined in “[Run-Time Statistics for Stateful Session EJBs](#)”.

You can tailor this console page to display some or all of the underlying counts from which the statistics are calculated, by clicking [Customize this View](#), and selecting the desired data from the list of available columns. The available data is defined in “[Raw Monitoring Data for Stateful Session EJBs](#)”.

The statistics and underlying counts reflect activity since the bean was last deployed.

Run-Time Statistics for Stateful Session EJBs

The following table defines the run-time statistics displayed for stateful session EJBs.

Table 89-1 Stateful Session EJB Run-Time Statistics

Statistic	Description
Cache Miss Ratio	<p>The cache miss ratio is a ratio of the number of times a container cannot find a bean in the cache (cache miss) to the number of times it attempts to find a bean in the cache (cache access). In general, the lower your cache miss ratio, the better your EJB will perform.</p> <p>The amount of time saved by getting a bean from the cache depends on the cost of the bean's <code>ejbActivate</code> method as well as the bean's cache-between-transactions setting. When a cache miss occurs, a bean must be obtained from the free pool and its <code>ejbActivate</code> method must be called. The more expensive it is to invoke <code>ejbActivate</code>, the more the cache miss will hurt performance. If the EJB is configured with cache-between-transactions set to true, the cache miss will also force the EJB container to make an extra call to the database to load the bean.</p> <p>For information on what to tune in response to the cache miss ratio statistic, see “Cache Miss Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>
Lock Manager Waiter Ratio	<p>This is the ratio of the number of times a thread had to wait to obtain a lock on a bean to the total amount of lock requests issued. For best performance, you want the lock waiter ratio to be as low as possible.</p> <p>For information on what to tune in response to the lock waiter ratio statistic, see “Lock Waiter Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>
Lock Manager Timeout Ratio	<p>This is the ratio of timeouts to accesses for the lock manager. Timeouts are very detrimental to performance and therefore, you should strive to keep your lock timeout ratio to an absolute minimum. Timeouts hurt performance on several levels. First, each thread waiting for a lock is one less thread that the server can be using to service other requests. Second, a lock timeout will result in an exception that will roll back the current transaction, erasing any work already done in the transaction and causing the current request to fail.</p> <p>For information on what to tune in response to the lock timeout ratio statistic, see “Lock Timeout Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>

Statistic	Description
Transaction Rollback Ratio	<p>The transaction rollback ratio is the ratio of transactions that have rolled back to the number of total transactions involving the EJB. 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. It may also be useful in gauging 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.</p> <p>For information on what to tune in response to the transaction rollback ratio statistic, see “Transaction Rollback Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>
Transaction Timeout Ratio	<p>The transaction timeout ratio is the ratio of transactions that have timed out to the total number of transactions involving an EJB. Timeouts can be especially concerning because they are a signal of inefficiency.</p> <p>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 vein. The transaction timeout ratio is a good indicator of a problem with an application.</p> <p>For information on what to tune in response to the transaction timeout ratio statistic, see “Transaction Timeout Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>

Raw Monitoring Data for Stateful Session EJBs

This section defines the underlying run-time counts from which the statistics in “Run-Time Statistics for Stateful Session EJBs” are calculated. To display the counts on the EJB --> Monitoring -->Stateful Session EJBs tab page, select Customize this View and choose from the list of available data.

Table 89-2 Stateful Session EJB Raw Monitoring Data.

Run-Time Count	Description
Activation Count	
Beans In Use Current Count	Displays the number of bean instances 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.
Cache Access Count	Displays 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 Miss Count	Displays the total number of times an attempt to access a bean from the cache failed. This information is useful for determining the effectiveness of the EJB cache.
Cached Beans Current Count	
Destroyed Total Count	
EJB Name	
Lock Mgr Access Count	Displays 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.
Lock Mgr Entries Current Count	Displays the current number of lock entries in the lock manager. This information can be helpful in detecting stale lock entries.
Lock Mgr Timeout Total Count	Displays the total number of threads that have timed out waiting for a lock on a bean. This information is useful for calculating the lock timeout ratio.
Lock Mgr Waiter Total Count	Displays the total number of threads that have waited for a lock on a bean. This information is useful for calculating the lock waiter ratio.
Passivation Count	
Server	

Run-Time Count	Description
----------------	-------------

Transactions Committed Total Count
--

Transactions Rolled Back Total Count
--

Transactions Timed out Total Count
--

Tasks

“Monitoring EJBs” on page 103-8

Related Topics

“[Tuning WebLogic Server EJBs](#)” in *WebLogic Server Performance and Tuning*.



Deployments-->EJB --> Monitoring--> Message Driven EJBs

[Run-time Statistics for MDBs](#) [Raw Monitoring Data for Message-Driven EJBs](#) [Tasks](#) [Related Topics](#)

Overview

This page displays run-time statistics for MDBs in the currently selected archive. You can display statistics for one or more server instances on which the EJB is deployed. The statistics are defined in “Message Driven EJB Run-Time Statistics”.

You can tailor this console page to display some or all of the underlying counts from which the statistics are calculated, by clicking [Customize this View](#), and selecting the desired data from the list of available columns. The available data is defined in “Raw Monitoring Data for Message-Driven EJBs”.

The statistics and underlying counts reflect activity since the bean was last deployed.

Run-time Statistics for MDBs

The following table defines the run-time statistics displayed for entity EJBs.

Table 90-1 Message Driven EJB Run-Time Statistics

Pool Miss Ratio	<p>The pool miss ratio is a ratio of the number of times a request was made to get a bean from the pool when no beans were available, to the total number of requests for a bean made to the pool. The consequence of a pool miss is different for different types of beans.</p> <p>A pool miss for a stateless session bean will cause the requesting thread to wait for a bean to become available in the pool. The maximum time a thread will wait is equal to the transaction timeout value for the bean.</p> <p>Entity beans and message-driven beans will never wait for an instance to become available. Instead, a pool miss will cause the pool to create a new bean instance to service the request. Pool misses come at a cost since the executing thread will either have to wait for a bean to become available or have to wait for a new bean to be created. As such, it is best to try to keep your pool miss ratio to a minimum.</p> <p>For information on what to tune in response to the pool miss ratio statistic, see “Pool Miss Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>
Destroyed Bean Ratio	<p>The destroyed bean ratio is a ratio of the number of beans destroyed to the total number of requests for a bean. The EJB specification mandates that the EJB container destroys a bean when non-application exceptions are thrown from the bean during execution. Destroying beans comes at a cost, however, because destroyed beans will likely have to be replaced with new bean instances. As a result, you should keep your destroyed bean ratio to a minimum.</p> <p>For information on what to tune in response to the destroyed bean ratio statistic, see “Destroyed Bean Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>
JMS Connection Alive	<p>Reports whether the EJB container has successfully connected to the JMS destination source and that therefore the message-driven bean is receiving messages. If this field’s value displays as <code>false</code>, check the server log for possible reasons for connection failure.</p>

Transaction Rollback Ratio	<p>The transaction rollback ratio is the ratio of transactions that have rolled back to the number of total transactions involving the EJB. 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. It may also be useful in gauging 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.</p> <p>For information on what to tune in response to the transaction rollback ratio statistic, see “Transaction Rollback Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>
Transaction Timeout Ratio	<p>The transaction timeout ratio is the ratio of transactions that have timed out to the total number of transactions involving an EJB. Timeouts can be especially concerning because they are a signal of inefficiency.</p> <p>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 vein. The transaction timeout ratio is a good indicator of a problem with an application.</p> <p>For information on what to tune in response to the transaction timeout ratio statistic, see “Transaction Timeout Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>

Raw Monitoring Data for Message-Driven EJBs

This section defines the underlying run-time counts from which the statistics in “Run-time Statistics for MDBs” are calculated. To display the counts on the EJB --> Monitoring --> Message Driven EJBs tab page, select “Customize this View” and choose from the list of available data.

Table 90-2 Message-Driven EJB Raw Monitoring Data

Run-Time Count	Description
Access Total Count	Displays the total number of times an attempt was made to get an instance from the free pool. This information is useful for giving context to the other free pool counts.
Beans In Use Current Count	Displays the number of bean instances 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.
Destroyed Total Count	
EJB Name	
Miss Total Count	Displays the total number of times a failed attempt was made to get an instance from the free pool. An Attempt to get a bean from the pool will fail if there are no available instances in the pool. This information is useful for calculating the pool miss ratio.
Pool Timeout Total Count	Displays the total number of Threads that have timed out waiting for an available bean instance from the free pool. This information is useful for calculating the pool timeout ratio.
Pool Waiter Total Count	
Pooled Beans Current Count	Displays the current number of available bean instances in 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.
Server	
Transactions Committed Total Count	
Transactions Rolled Back Total Count	

Run-Time Count	Description
Transactions Timed out Total Count	
Waiter Current Count	Displays the number of threads currently waiting for an available bean instance from the free pool. This information may be useful, for example, for investigating the cause of poor application performance at a particular time.

Tasks

[“Monitoring EJBs” on page 103-8](#)

Related Topics

[“Tuning WebLogic Server EJBs”](#) in *WebLogic Server Performance and Tuning*.



Deployments-->EJB --> Monitoring --> Entity EJBs

[Run-Time Statistics for Entity EJBs](#) [Raw Monitoring Data for Entity EJBs](#) [Tasks](#) [Related Topics](#)

Overview

This page displays run-time statistics for entity EJBs in the currently selected archive. You can display statistics for one or more server instances on which the EJB is deployed. The statistics are defined in [“Run-Time Statistics for Entity EJBs”](#).

You can tailor this console page to display some or all of the underlying counts from which the statistics are calculated, by clicking [Customize this View](#), and selecting the desired data from the list of available columns. The available data is defined in [“Raw Monitoring Data for Entity EJBs”](#).

The statistics and underlying counts reflect activity since the bean was last deployed.

Run-Time Statistics for Entity EJBs

The following table defines the run-time statistics displayed for entity EJBs.

Table 91-1 Entity EJB Run-Time Statistics

Statistic	Description
Cache Miss Ratio	<p>The cache miss ratio is a ratio of the number of times a container cannot find a bean in the cache (cache miss) to the number of times it attempts to find a bean in the cache (cache access). In general, the lower your cache miss ratio, the better your EJB will perform.</p> <p>The amount of time saved by getting a bean from the cache depends on the cost of the bean's <code>ejbActivate</code> method as well as the bean's cache-between-transactions setting. When a cache miss occurs, a bean must be obtained from the free pool and its <code>ejbActivate</code> method must be called. The more expensive it is to invoke <code>ejbActivate</code>, the more the cache miss will hurt performance. If the EJB is configured with cache-between-transactions set to true, the cache miss will also force the EJB container to make an extra call to the database to load the bean.</p> <p>For information on what to tune in response to the cache miss ratio statistic, see “Cache Miss Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>
Lock Manager Waiter Ratio	<p>This is the ratio of the number of times a thread had to wait to obtain a lock on a bean to the total amount of lock requests issued. For best performance, you want the lock waiter ratio to be as low as possible.</p> <p>For information on what to tune in response to the lock waiter ratio statistic, see “Lock Waiter Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>
Lock Manager Timeout Ratio	<p>This is the ratio of timeouts to accesses for the lock manager. Timeouts are very detrimental to performance and therefore, you should strive to keep your lock timeout ratio to an absolute minimum. Timeouts hurt performance on several levels. First, each thread waiting for a lock is one less thread that the server can be using to service other requests. Second, a lock timeout will result in an exception that will roll back the current transaction, erasing any work already done in the transaction and causing the current request to fail.</p> <p>For information on what to tune in response to the lock timeout ratio statistic, see “Lock Timeout Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>

Statistic	Description
Pool Miss Ratio	<p>The pool miss ratio is a ratio of the number of times a request was made to get a bean from the pool when no beans were available, to the total number of requests for a bean made to the pool. The consequence of a pool miss is different for different types of beans.</p> <p>A pool miss for a stateless session bean will cause the requesting thread to wait for a bean to become available in the pool. The maximum time a thread will wait is equal to the transaction timeout value for the bean.</p> <p>Entity beans and message-driven beans will never wait for an instance to become available. Instead, a pool miss will cause the pool to create a new bean instance to service the request. Pool misses come at a cost since the executing thread will either have to wait for a bean to become available or have to wait for a new bean to be created. As such, it is best to try to keep your pool miss ratio to a minimum.</p> <p>For information on what to tune in response to the pool miss ratio statistic, see “Pool Miss Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>
Pool Timeout Ratio	<p>The pool timeout ratio is a ratio of requests that have timed out waiting for a bean from the pool to the total number of requests made. This ratio is only valid for stateless session beans because it is the only type of bean that will wait for a bean to become available.</p> <p>Other types of beans will automatically create a new instance to service a request rather than waiting. For best performance, the pool timeout ratio should be as small as possible.</p> <p>For information on what to tune in response to the pool timeout ratio statistic, see “Pool Timeout Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>
Destroyed Bean Ratio	<p>The destroyed bean ratio is a ratio of the number of beans destroyed to the total number of requests for a bean. The EJB specification mandates that the EJB container destroys a bean when non-application exceptions are thrown from the bean during execution. Destroying beans comes at a cost, however, because destroyed beans will likely have to be replaced with new bean instances. As a result, you should keep your destroyed bean ratio to a minimum.</p> <p>For information on what to tune in response to the destroyed bean ratio statistic, see “Destroyed Bean Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>

Statistic	Description
JMS Connection Alive	Reports whether the EJB container has successfully connected to the JMS destination source and that therefore the message-driven bean is receiving messages. If this field's value displays as <code>false</code> , check the server log for possible reasons for connection failure.
Transaction Rollback Ratio	<p>The transaction rollback ratio is the ratio of transactions that have rolled back to the number of total transactions involving the EJB. 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. It may also be useful in gauging 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.</p> <p>For information on what to tune in response to the transaction rollback ratio statistic, see “Transaction Rollback Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>
Transaction Timeout Ratio	<p>The transaction timeout ratio is the ratio of transactions that have timed out to the total number of transactions involving an EJB. Timeouts can be especially concerning because they are a signal of inefficiency.</p> <p>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 vein. The transaction timeout ratio is a good indicator of a problem with an application.</p> <p>For information on what to tune in response to the transaction timeout ratio statistic, see “Transaction Timeout Ratio” in the <i>WebLogic Server Performance and Tuning Guide</i>.</p>

Raw Monitoring Data for Entity EJBs

This section defines the underlying run-time counts from which the statistics in “Run-Time Statistics for Entity EJBs” are calculated. To display the counts on the EJB --> Monitoring --> Entity EJBs tab page, select Customize this View and choose from the list of available data.

Table 91-2 Entity EJB Raw Monitoring Data

Run-Time Count	Description
Access Total Count	Displays the total number of times an attempt was made to get an instance from the free pool. This information is useful for giving context to the other free pool counts.
Activation Count	
Beans In Use Current Count	Displays the number of bean instances 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.
Cache Access Count	Displays 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 Miss Count	Displays the total number of times an attempt to access a bean from the cache failed. This information is useful for determining the effectiveness of the EJB cache.
Cached Beans Current Count	
Destroyed Total Count	
EJB Name	
Lock Mgr Access Count	
Lock Mgr Entries Current Count	
Lock Mgr Timeout Total Count	
Lock Mgr Waiter Total Count	
Miss Total Count	Displays the total number of times a failed attempt was made to get an instance from the free pool. An Attempt to get a bean from the pool will fail if there are no available instances in the pool. This information is useful for calculating the pool miss ratio.
Passivation Count	

Run-Time Count	Description
Pool Timeout Total Count	Displays the total number of Threads that have timed out waiting for an available bean instance from the free pool. This information is useful for calculating the pool timeout ratio.
Pool Waiter Total Count	
Pooled Beans Current Count	Displays the current number of available bean instances in 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.
Server	
Transactions Committed Total Count	
Transactions Rolled Back Total Count	
Transactions Timed out Total Count	
Waiter Current Count	Displays the number of threads currently waiting for an available bean instance from the free pool. This information may be useful, for example, for investigating the cause of poor application performance at a particular time.

Tasks

“Monitoring EJBs” on page 103-8

Related Topics

“[Tuning WebLogic Server EJBs](#)” in *WebLogic Server Performance and Tuning*.

EJB Module Deployment Assistant --> Step 3

- Review Choices and Deploy

[Tasks](#) [Related Topics](#)

Overview

This page displays a list of the targeted clusters and servers for the EJB module. Review the entries under the Deployment Targets heading. If you need to change a target, click your browser's Back button.

The Source accessibility header displays the selected staging mode for deploying the EJB source files:

- **Copy this application onto every target for me**—This option is selected by default if you targeted the EJB to a cluster or to multiple server instances. This corresponds to “stage” mode where the Administration Server copies the EJB files to each targeted server; and the target servers then deploy the EJB using their copy of the files.
- **I will make the application accessible from the following location**—This option is selected by default if you targeted the EJB to a single server instance. This corresponds to the “nostage” mode where the server deploys an EJB from a single directory; all targeted servers must be able to access the directory to deploy the EJB. Select this option if you are deploying to a cluster that resides on a single physical machine.

In the Identity header, the Name field specifies a unique name to refer to this EJB module in the Administration Console. Accept the default name or enter a new name to describe the EJB.

Click Deploy to accept the values on this page and deploy the EJB to all specified server instances.

Tasks

[“Deploying a New EJB Module” on page 103-2](#)

[“Configuring an EJB Module” on page 103-3](#)

[“Stopping or Redeploying an EJB Module” on page 103-7](#)

Related Topics

[Packaging and Deploying WebLogic Server Applications](#)

EJB Module Deployment Assistant --> Step 1

- Select Archive

[Tasks](#) [Related Topics](#)

Overview

The EJB Module Deployment Assistant helps you deploy a new EJB to one or more servers in the domain. You can select either an archived EJB module (.jar file), or an EJB in exploded .jar format.

Use the links in the Location field to browse directories on the Administration Server machine and locate the EJB to deploy. If the EJB does not reside on the Administration Server machine, first use the upload link to upload the EJB .jar file. This places the archive in the Administration Server's configured upload directory, and automatically opens that directory in the Location field.

When the assistant detects an .jar file or exploded .jar directory in the current directory, it lists the archive or directory name as a selection beneath the Location field. Select the name of the archive or directory you want to deploy.

If your domain contains multiple WebLogic Server instances, click Target Module to deploy the EJB to a single server, or to multiple server instances or clusters.

In a single server domain, click Continue to automatically target the EJB to the available server instance.

Tasks

[“Deploying a New EJB Module” on page 103-2](#)

[“Stopping or Redeploying an EJB Module” on page 103-7](#)

[“Setting an EJB Module's Target Server and/or Cluster” on page 103-7](#)

Related Topics

[Programming WebLogic Enterprise JavaBeans](#)

EJB --> Targets

[Tasks](#) [Attributes](#)

Overview

Use this page to select the independent servers or clusters on which you would like to deploy this EJB module at server startup time. You can reconfigure deployment targets later if you wish. To deploy or undeploy this EJB module immediately without restarting your server(s), click the Deploy tab.

Tasks

“Setting an EJB Module’s Target Server and/or Cluster” on page 103-7

Attributes

Attribute Label	Description	Value Constraints
Targets	The servers and/or clusters in the current domain to which this EJB module is targeted and on which it can be deployed. <i>MBean:</i> weblogic.management.configuration.EJBComponentMBean <i>Attribute:</i> Targets	<i>Configurable:</i> yes <i>Dynamic:</i> yes <i>Readable:</i> yes <i>Writable:</i> yes



EJB Module Deployment Assistant --> Step 2

- Select Targets

[Tasks](#) [Related Topics](#)

Overview

This page displays the available servers and clusters to which you can deploy an EJB module.

To deploy to individual servers, select one or more server instances from the Independent Servers list and click Continue.

To deploy to a cluster of servers, select the name of the cluster from the Clusters list. By default, the assistant deploys an EJB to all server instances in the cluster (the All servers in the cluster option). If you want to deploy only to a subset of the servers in a cluster, select Parts of the cluster, and then select the individual server instances to which you want to deploy the EJB.

Click Continue to review your choices and deploy the EJB module.

Tasks

[“Deploying a New EJB Module” on page 103-2](#)

[“Stopping or Redeploying an EJB Module” on page 103-7](#)

[“Setting an EJB Module’s Target Server and/or Cluster” on page 103-7](#)

Related Topics

[Programming WebLogic Enterprise JavaBeans](#)



EJB --> Testing

Tasks

Overview

You can test a remote EJB to see whether it can be found via its JNDI name via the Testing tab.

Tasks

“Testing an EJB Module” on page 103-10



EJB --> Notes

[Tasks](#) [Attributes](#)

Overview

You can provide or edit optional information about your application by entering information the Notes.

Tasks

“Adding or Editing EJB Module Deployment Notes” on page 103-10

Attributes

Table 97-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.EJBComponentMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes



EJB Modules

The EJB Deployments page for an application displays a table of the EJBs, packaged as .jars or exploded .jar directories, that are deployed for an application.

To deploy a new EJB that is packaged as a .jar or exploded .jar directory, select the "Deploy a New EJB" link.

- For more information about deployment, configuration, and monitoring tasks for EJB that are packaged as a .jar or exploded .jar directory, see [Tasks](#).
- For information about deployment, configuration, and monitoring tasks for EJBs that are packaged as an enterprise application in an .ear or exploded .ear directory, see [“Applications” on page 16-1](#)



EJB --> Configuration --> Descriptors --> Stateless Session EJBs

[Tasks](#) [Related Topics](#)

Overview

Via the Administration Console, you can tune stateless session EJBs by dynamically updating the values of the following deployment descriptor elements:

- Max Beans in Free Pool
- Transaction Timeout Seconds

Tasks

“Configuring Deployment Descriptor Values” on page 103-4

Related Topics

[Tuning WebLogic Server EJBs](#)



EJB --> Configuration --> Descriptors --> Stateful Session EJBs

[Tasks](#) [Related Topics](#)

Overview

Via the Administration Console, you can tune stateful session EJBs by dynamically updating the values of the following deployment descriptor elements:

- Max Beans in Free Pool
- Idle Timeout Seconds
- Transaction Timeout Seconds
- Allow Remove During Transaction— specifies that the remove method on stateful session bean can be invoked within a transaction context.

Note: A stateful session bean that implements the `synchronization` interface and has Allow Remove During Transaction enabled should not call `remove` before the transaction ends—it it does, the container will not invoke the synchronization callbacks.

Tasks

“Configuring Deployment Descriptor Values” on page 103-4

Related Topics

[Tuning WebLogic Server EJBs](#)



EJB --> Configuration --> Descriptors --> Message-Driven EJBs

[Tasks](#) [Related Topics](#)

Overview

Via the Administration Console, you can tune message-driven EJBs by dynamically updating the values of the following deployment descriptor elements:

- Max Beans in Free Pool
- Message Selector
- JMSPolling Interval Seconds
- Transaction Timeout Seconds

Tasks

“Configuring Deployment Descriptor Values” on page 103-4

Related Topics

[Tuning WebLogic Server EJBs](#)



EJB --> Configuration --> Descriptors --> Entity EJBs

[Tasks](#) [Related Topics](#)

Overview

Via the Administration Console, you can tune entity EJBs by dynamically updating the values of the following deployment descriptor elements:

- Max Beans in Free Pool
- Transaction Timeout Seconds
- Max Beans in Cache
- Key Cache Size

Tasks

“Configuring Deployment Descriptor Values” on page 103-4

Related Topics

[Tuning WebLogic Server EJBs](#)



1 EJB

[\[“Attributes and Console Screen Reference for EJB” on page 104-1\]](#)

Enterprise JavaBeans (EJB) are reusable Java components that implement business logic and enable you to develop component-based distributed business applications.

EJBs reside in an EJB container that provides a standard set of services, including persistence, security, transactions, and concurrency. Enterprise JavaBeans are the standard for defining server-side components. WebLogic Server’s implementation of the Enterprise JavaBeans component architecture is based on Sun Microsystems EJB specification.

Using the Administration Console, you can configure, target, deploy, stop, redeploy, display information about and monitor the performance of EJBs.

For information about types of EJBs, design considerations, and development topics, see [Programming WebLogic Server Enterprise Java Beans](#).

Tasks

These topics provide instructions for performing command tasks for EJBs using the Administration Console:

- [Deploying a New EJB Module](#)
- [Configuring an EJB Module](#)
- [Displaying and Configuring General Information](#)
- [Configuring Deployment Descriptor Values](#)

- [Configuring Compiler Options](#)
- [Viewing an EJB Module's Deployment Status](#)
- [Stopping or Redeploying an EJB](#)
- [Setting an EJB Module's Target Server and/or Cluster](#)
- [Monitoring EJBs](#)
- [Testing an EJB](#)
- [Adding or Editing EJB Deployment Notes](#)

Deploying a New EJB Module

Follow these steps to deploy an EJB that is packaged in a .jar file or an exploded .jar directory.

1. In the left pane of the Console, select the Deployments-->EJB Modules node.
EJBs that are packaged in a .jar file or an exploded .jar directory are listed.

Note: EJB modules that packaged in an enterprise application are listed under Deployment --> Applications, under the node for the application name. To deploy an EJB that is packaged in an enterprise application, see [“Deploying a New Enterprise Application or Web Service”](#).

2. In the right pane of the Console, select the “Deploy an EJB module” option.
3. On the “Select the archive for this application” page, browse to the archive file, or the exploded component directory that you want to deploy, select the radio button next to the item and click Target Module.

Note: If the item you want to deploy is not available on the administration server, select “upload your files” to upload it.

4. On the “Select targets for this module” page, select the servers and/or clusters to which you want to deploy the EJB module, and click Continue.
5. On the “Review your choices and deploy” page, specify how servers access the EJB's source files. There are two options:

- Copy the EJB onto every target.
 - Enter the location from which all targets access the EJB's source files.
6. On the “Review your choices and deploy” page, enter a name by which to identify the EJB and click Deploy.

The Deploy tab displays the deployment status for the EJB module.

For an overview of deploying from the Administration Console, see [“Deploying Applications and Modules”](#).

Configuring an EJB Module

After you have deployed or tried to deploy an EJB that is packaged as a .jar or an exploded.jar directory, the EJB name appears in the left pane of the console. Use the Deployments-->EJB-->Configuration tab to configure the EJB module.

The Deployments-->EJB-->Configuration tab contains the following pages:

- **General**—This page displays the name, path, staging mode, and load order for an EJB.

See “Displaying and Configuring General Information” on page 103-4.

- **Descriptors**—This page allows you to:

- View the name of the EJB module and its type.
- Modify values of tuning-related deployment descriptor elements for the EJB module.
- Display a read-only version of the EJB deployment descriptors:
ejb-jar.xml, weblogic-ejb-jar.xml and
weblogic-cmp-rdbms-jar.xml.

See “Configuring Deployment Descriptor Values” on page 103-4.

- **Compiler options**—This page allows you to configure the Java compiler and associated options for compiling the EJB.

See “Configuring Compiler Options” on page 103-5.

Displaying and Configuring General Information

Follow these steps to configure or display information for a deployed EJB module:

1. In the left pane of the Console, select the Deployments-->EJB Modules node.

EJBs that are packaged as a .jar or a exploded .jar directory are listed.

Note: EJB modules that packaged in an enterprise application are listed under Deployment --> Applications, under the node for the application name.

2. Select the desired EJB module, and select the Configuration--> General tab.

The right pane of Console displays:

- Name—the name for the deployment.
- Path—the path to the archive or application directory.
- Load Order—the order in which the EJB module is loaded at startup. To set or change the load order, enter a numerical value and click Apply.
- Staging Mode

For attribute descriptions, see [EJB --> Configuration --> General](#).

Configuring Deployment Descriptor Values

The Configuration-->Descriptors tab under the Deployments-->EJB Modules node allows you to:

- Display a read-only version of the all EJB deployment descriptors:
ejb-jar.xml, weblogic-ejb-jar.xml and weblogic-cmp-rdbms-jar.xml
for a deployed EJB module.
- Configure selected deployment descriptor element values for an EJB that is deployed as an exploded application.

Follow these steps to configure selected deployment descriptor values for an EJB that is deployed as an exploded application:

1. In the left pane of the Console, select the Deployments-->EJB Modules node.

Standalone EJB modules are listed. (EJB modules that packaged in an enterprise application are listed under Deployment --> Applications, under the sub-node for the application name.

2. Select the desired EJB module, and select the Configuration-->Descriptors tab.
For exploded EJBs, the EJB name and its type are displayed in the right pane.
3. Click the EJB module name in the right pane.
Selected deployment descriptor elements for the EJB module are listed.
4. Enter new values for the elements you want change, and click Apply.
5. To view a read-only deployment descriptor file, click its name.

For attribute descriptions, see [EJB --> Configuration --> Descriptors](#).

Configuring Compiler Options

Follow these steps to configure compiler options for a deployed EJB module:

1. In the left pane of the Console, select the Deployments-->EJB Modules node.
EJBs that are packaged as a .jar or an exploded .jar directory are listed.

Note: EJB modules that packaged in an enterprise application are listed under Deployment --> Applications, under the sub-node for the application name.

2. Select the Configuration--> Compiler options tab.
3. Enter the Java compiler to be used for the EJB module.
4. Enter RMI compiler options.
5. Enter EJB compiler options.

Note: You can specify RMI and EJB compiler options at the server level, on the Server --> Configuration --> General page. The EJB-level setting takes precedence over the server-level setting.

6. Check the box to enable or disable the ability to keep generated source files.
7. Check the box to enable or disable forced regeneration of wrapper classes.

8. Click Apply.

For attribute descriptions, see [EJB --> Configuration --> Compiler options](#).

Viewing an EJB Module's Deployment Status

To view an EJB's deployment status:

1. Select the Deployments --> EJB Modules node in the left pane of the Console.

After you expand it, the Deployments --> EJB Modules node lists the EJBs that are packaged in a .jar or an exploded .jar directory that you have deployed, or tried to deploy.

Note: EJB modules that are packaged in an enterprise application .ear are listed under Deployment --> Applications, under the node for the application.

2. Click the Deploy tab.

The EJB module's deployment status is displayed in the right pane.

- Active—The module is deployed.
- Inactive—The module is not deployed, either because the deployment attempt failed, or because the module was stopped.
- Failed—The deployment attempt failed.

Stopping or Redeploying an EJB

To stop a deployed EJB, or redeploy a stopped EJB:

1. Select the Deployments --> EJB Modules node in the left pane of the Console.

After you expand it, the Deployments --> EJB Modules node lists the EJBs that are packaged in a .jar or an exploded .jar directory that you have deployed, or tried to deploy.

Note: EJBs that are packaged in an enterprise application .ear are listed under Deployment --> Applications, under the node for the application.

2. Select the desired EJB, and click the Deploy tab.

The EJB Deployments page displays the deployment status of the EJB:.

- Active—The module is deployed.
- Inactive—The module is not deployed, either because the deployment attempt failed, or because the module was stopped.
- Failed—The deployment attempt failed.

3. Click Stop to stop a deployed EJB module, or click Redeploy to redeploy a stopped EJB module.

Setting an EJB Module's Target Server and/or Cluster

To target an EJB to one or more servers or clusters:

1. Select the Deployments --> EJB Modules node in the left pane of the Console.

After you expand it, the Deployments --> EJB Modules node lists the EJBs that are packaged in a .jar or an exploded .jar directory that you have deployed, or tried to deploy.

Note: EJB modules that are packaged in an enterprise application .ear are listed under Deployment --> Applications, under the node for the application.

2. Select the desired EJB, and click the Targets tab.

Servers and clusters in the domain are displayed.

3. Select the individual server(s) and/or cluster(s) to which you wish to deploy the EJB and click Apply.

By default, the Administration Console deploys an EJB to all server instances in the cluster (the “All servers in the cluster” option). To deploy the EJB to selected servers in a cluster, select “Part of the cluster”, and then select the individual server instances to which you want to deploy the EJB.

Monitoring EJBs

To monitor a deployed EJB:

1. Select the Deployments --> EJB Modules node in the left pane of the Console.

After you expand it, the Deployments --> EJB Modules node lists the EJBs that are packaged in a .jar or an exploded .jar directory that you have deployed, or tried to deploy.

Note: EJB modules that are packaged in an enterprise applications .ear are listed under Deployment --> Applications, under the node for the application.

2. Select the desired EJB, and click the Monitoring tab.

A table listing statistics about the EJB displays in the right pane. For descriptions of the data, see:

- [Deployments-->EJB-->Monitoring--> Stateless EJBs](#)
- [Deployments-->EJB --> Monitoring --> Stateful Session EJBs](#)
- [Deployments-->EJB --> Monitoring --> Entity EJBs](#)
- [Deployments-->EJB --> Monitoring--> Message Driven EJBs](#)

3. You can monitor the EJB on a single server, or on all servers to which it is deployed. Select the deployment you wish to monitor from the Select Servers drop-down list.

Testing an EJB

To test an EJB:

1. Select the Deployments --> EJB Modules node in the left pane of the Console.

After you expand it, the Deployments --> EJB Modules node lists the EJBs that are packaged in a .jar or an exploded .jar directory that you have deployed, or tried to deploy.

Note: EJB modules that are packaged in an enterprise application .ear, are listed under Deployments --> Applications, under the node for the application.

2. Click the Testing tab.

If the test succeeds, the following message is displayed:

The *EJB_name* has been tested successfully with a JNDI name of *JNDI_name*.

If the test fails, the following message is displayed:

The EJB *EJB_name* has not been tested successfully. There was a problem determining the JNDI Name of the given bean.

Adding or Editing EJB Deployment Notes

To add or edit EJB deployment notes:

1. Select the Deployments --> EJB Modules node in the left pane of the Console.

After you expand it, the Deployments --> EJB Modules node lists the EJBs that are packaged in a .jar or an exploded .jar directory that you have deployed, or tried to deploy.

Note: EJB modules that are packaged in an enterprise application .ear are listed under Deployment --> Applications, under the node for the application.

2. Click the Notes tab.
3. Add or edit the optional information in the Notes scroll box.
4. Click Apply.

Attributes and Console Screen Reference for EJB

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

“EJB Modules” on page 98-1

“EJB Module Deployment Assistant --> Step 1 - Select Archive” on page 93-1

“EJB Module Deployment Assistant --> Step 2 - Select Targets” on page 95-1

“EJB Module Deployment Assistant --> Step 3 - Review Choices and Deploy” on page 92-1

“EJB --> Configuration --> General” on page 85-1

“EJB --> Configuration --> Descriptors” on page 83-1

“EJB --> Configuration --> Descriptors --> Stateful Session EJBs” on page 100-1

“EJB --> Configuration --> Descriptors --> Stateless Session EJBs” on page 99-1

“EJB --> Configuration --> Descriptors --> Entity EJBs” on page 102-1

“EJB --> Configuration --> Descriptors --> Message-Driven EJBs” on page 101-1

“EJB --> Configuration --> Compiler options” on page 84-1

“EJB --> Testing” on page 96-1

“EJB --> Targets” on page 94-1

“EJB --> Deploy” on page 86-1

“EJB --> Monitoring” on page 87-1

“Deployments-->EJB --> Monitoring --> Stateful Session EJBs” on page 89-1

“Deployments-->EJB-->Monitoring--> Stateless EJBs” on page 88-1

“Deployments-->EJB --> Monitoring--> Message Driven EJBs” on page 90-1

“Deployments-->EJB --> Monitoring --> Entity EJBs” on page 91-1

“EJB --> Notes” on page 97-1

“Server --> Deployments --> EJB Modules” on page 475-1

FileT3 (Deprecated)--> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to configure FileT3 file systems.

Tasks

“Create a File System” on page 109-1

Related Topics

[Using WebLogic File Services](#)

Attributes

Table 105-1

Attribute Label	Description	Value Constraints
Name	The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.	

Table 105-1

Attribute Label	Description	Value Constraints
Path	<p>The path used to locate a T3 file service on a server host.</p> <p>For example: To map the file system name <code>users</code> to the path on the server host <code>/usr/local/tmp</code>, specify the value of the Name attribute as <code>users</code> and specify the value of the Path attribute as <code>/usr/local/tmp</code>.</p>	

FileT3 (Deprecated) --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab stores optional configuration notes for the WebLogic Server File service.

Tasks

Enter free form text notes to describe the function or configuration.

Related Topics

[Using WebLogic File Services](#)

Attributes

Table 106-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration.	<i>Dynamic:</i> yes



Domain --> FileT3 --> Target and Deploy

[Tasks](#) [Related Topics](#)

Overview

On the Domain →FileT3 →Target and Deploy tab, you deploy the selected file on servers and clusters in the domain.

Tasks

“Assign Servers for a FileT3 File System” on page 109-2

“Assign Clusters for a FileT3 File System” on page 109-3

Related Topics

[Using WebLogic File Services](#)



FileT3 (Deprecated)

The WebLogic File service allows you to provide high-speed, client-side access to native operating system files on the server. Use the client API to extend the capabilities of `java.io.InputStream` and `java.io.OutputStream`.

- For more information, see [“Create a File System” on page 109-1](#)
- For more information, see [“Modify a File System Path NAME” on page 109-2](#)
- For more information, see [“Assign Servers for a FileT3 File System” on page 109-2](#)
- For more information, see [“Assign Clusters for a FileT3 File System” on page 109-3](#)



1 FileT3 (Deprecated)

[“Attributes and Console Screen Reference for FileT3” on page 110-1]

The WebLogic File service allows you to provide high-speed, client-side access to native operating system files on the server. Use the client API to extend the capabilities of `java.io.InputStream` and `java.io.OutputStream`.

Tasks

Create a File System

1. Click the Services node.
2. Click the FileT3 node.
3. Click the Configure a new FileT3 text link. A dialog displays in the right pane showing the tabs associated with configuring a file service.
4. Enter the name of your file system in the Name field.
5. Enter the path to the file system on the server host in the Path field.
6. Click the Create button in the lower right corner to create a file service instance with the name you specified in the Name field. The new instance is added under the FileT3 node in the left pane.
7. Select servers. See [Assign Servers for a FileT3 File System](#).
8. For Clustered environments, see [Assign Clusters for a FileT3 File System](#).

9. Restart the Administration Server so that your changes can take effect.

Modify a File System Path NAME

1. Click the Services node.
2. Click the FileT3 node.
3. Click the Name of the file system you want to modify.
4. Modify the value in the Path attribute field.
5. Click Apply.
6. Restart the Administration Server so that your changes can take effect.

Assign Servers for a FileT3 File System

1. Click the Services node.
2. Click the FileT3 node.
3. Click the Name of the file system you want to modify. A dialog displays in the right pane showing the tabs associated with configuring a file system.
4. Click the Targets tab.
5. Click the Servers tab.
6. To assign a FileT3 file system to the selected server, select a server in the list of Available servers and click the right arrow to move the server to the Chosen list. You can select multiple servers by Shift-clicking or Ctrl-clicking servers in the list.
7. To remove a FileT3 from a server, select a server in the list of Chosen servers and click the left arrow to move the server to the Available list. You can select multiple servers by Shift-clicking or Ctrl-clicking servers in the list.
8. Click Apply.
9. Restart the Administration Server so that your changes can take effect.

Assign Clusters for a FileT3 File System

1. Click the Services node.
2. Click the FileT3 node.
3. Click the Name of the file system you want to modify. A dialog displays in the right pane showing the tabs associated with configuring a file system.
4. Click the Targets tab.
5. Click the Clusters tab.
6. To assign a FileT3 file system to the selected cluster, select a cluster in the list of Available clusters and click the right arrow to move the cluster to the Chosen list. You can select multiple clusters by Shift-clicking or Ctrl-clicking clusters in the list.
7. To remove a FileT3 from a cluster, select a cluster in the list of Chosen clusters and click the left arrow to move the cluster to the Available list. You can select multiple clusters by Shift-clicking or Ctrl-clicking clusters in the list.
8. Click Apply.
9. Restart the Administration Server so that your changes can take effect.

Attributes and Console Screen Reference for FileT3

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“FileT3 \(Deprecated\)--> Configuration” on page 105-1](#)

[“FileT3 \(Deprecated\) --> Notes” on page 106-1](#)

[“FileT3 \(Deprecated\)” on page 108-1](#)

[“Domain --> FileT3 --> Target and Deploy” on page 107-1](#)



1 JDBC

[“Attributes and Console Screen Reference for JDBC” on page 112-1]

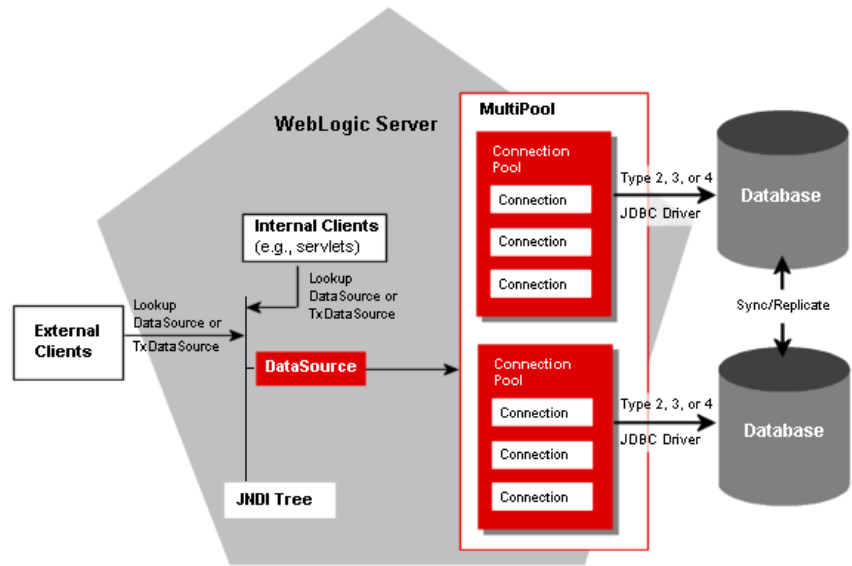
In the JDBC area of the Administration Console, you can configure and monitor JDBC components, including connection pools, MultiPools, and data sources. The following sections provide guidelines for configuring and managing database connectivity through the JDBC components for both local and distributed transactions:

- “Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1
- “Overview of JDBC Administration” on page 111-6

Creating and Deploying JDBC Components— Connection Pools, MultiPools, and Data Sources

Using the Administration Console, you create the JDBC components—connection pools, data sources, and MultiPools—by specifying attributes and database properties and then deploy them to servers and clusters by selecting targets.

Figure 111-1 JDBC Components in WebLogic Server



First you create the connection pools (and optionally, MultiPools) and deploy them to servers and clusters, then you create the data sources which are bound to the JNDI tree, and deploy the data sources to the same servers and clusters. When you create a data source object, you specify a connection pool or MultiPool as one of the data source attributes. This associates that data source with one specific connection pool or MultiPool ("pool").

When you target (deploy) data sources and connection pools, you should target the data source and its related connection pool to the same servers and clusters. Some common scenarios are as follows:

- In a single server configuration, assign each data source and its associated connection pool or MultiPool to the server.
- In a configuration with an administration server and managed servers, target the data source and associated connection pool to the servers on which they will be used.
- In a cluster, assign the data source and the associated connection pool or MultiPool to the cluster, not to individual servers in the cluster.

You can target data source/connection pool combinations to more than one server or cluster, but they must be targeted in combination. For example, you cannot target a data source to Managed Server A if its associated connection pool is assigned only to Managed Server B.

Related Information

- “JDBC Connection Pools” on page 127-1
- “Creating and Configuring a JDBC Connection Pool” on page 127-4
- “JDBC DataSources” on page 141-1
- “Creating and Configuring a JDBC Data Source” on page 141-6
- “JDBC MultiPools” on page 149-1
- “Creating and Configuring a JDBC MultiPool” on page 149-2

JDBC Objects in a Cluster

WebLogic Server allows you to cluster JDBC objects, including data sources, connection pools, and MultiPools, to improve the availability of cluster-hosted applications. Each JDBC object you configure for your cluster must exist on *each* managed server in the cluster—when you configure the JDBC objects, target them to the cluster.

For information about JDBC objects in a clustered environment, see “[JDBC Connections](#)” in *Using WebLogic Server Clusters* at [{DOCRROOT}/cluster/overview.html#JDBC](#).

Application Scoped JDBC Objects

See “Application-Scoped JDBC Data Sources and Connection Pools” on page 127-9.

JDBC Configuration Guidelines

To set up JDBC connectivity, you configure connection pools, data sources (always recommended, but optional in some cases), and MultiPools (optional) by defining attributes in the Administration Console or, for dynamic connection pools, in application code or at the command line.

There are three types of transaction scenarios:

- Local transactions—non-distributed transactions
- Distributed transactions using an XA Driver—distributed transactions with multiple participants that use two-phase commit
- Distributed transactions using a non-XA Driver—transactions with a single resource manager and single database instance that emulate two-phase commit

You configure data sources, connection pools, and MultiPools according to the way transactions are handled in your system. The following table summarizes how to configure these objects for use in the three transaction scenarios:

Table 111-2 Summary of JDBC Configuration Guidelines

Description/ Object	Local Transactions	Distributed Transactions XA Driver	Distributed Transactions Non-XA Driver
JDBC driver	<ul style="list-style-type: none"> ■ WebLogic jDriver for Oracle . ■ Compliant third-party drivers. 	<ul style="list-style-type: none"> ■ WebLogic jDriver for Oracle/XA. ■ XA-compliant third-party drivers. 	<ul style="list-style-type: none"> ■ WebLogic jDriver for Oracle ■ Compliant third-party drivers.
Data Source	Data source object recommended. (If there is no data source, use the JDBC API.)	Requires data source with Honor Global Transactions selected (Tx Data Source in the <code>config.xml</code> file).	Requires data source with Honor Global Transactions selected (Tx Data Source in the <code>config.xml</code> file). Select Emulate Two-Phase Commit for non-XA Driver (set <code>enable two-phase commit=true</code>) See “Configuring Non-XA JDBC Drivers for Distributed Transactions” on page 127-27 and “Emulating Two-Phase Commit” on page 141-3
Connection Pool	Requires data source object when configuring in the Administration Console.	Requires data source with Honor Global Transactions selected (Tx Data Source in the <code>config.xml</code> file).	Requires data source with Honor Global Transactions selected (Tx Data Source in the <code>config.xml</code> file).

Table 111-2 Summary of JDBC Configuration Guidelines

Description/ Object	Local Transactions	Distributed Transactions XA Driver	Distributed Transactions Non-XA Driver
MultiPool	Connection pool and data source required.	Not Supported	Not Supported

Note: For distributed transactions, use an XA-compliant driver, such as the *WebLogic jDriver for Oracle/XA*, which is the XA compliant version of the WebLogic jDriver for Oracle.

Overview of JDBC Administration

The Administration Console provides an interface to the tools that allow you to configure and manage WebLogic Server features, including JDBC (Java database connectivity). For most JDBC administrative functions, which include creating, managing and monitoring connectivity, systems administrators use the Administration Console or the command-line interface. Application developers may want to use the JDBC API or the WebLogic Management API.

Frequently performed tasks to set and manage connectivity include:

- Defining the attributes that govern JDBC connectivity between WebLogic Server and your database management system
- Monitoring established connectivity
- Tuning JDBC Objects

About the Administration Console

Your primary way to set and manage JDBC connectivity is through the Administration Console. In addition to setting connectivity, the Administration Console allows you to manage and monitor established connectivity.

You can also create connection pools, data sources, and MultiPools with the administration command line or in application code. For details, see [Commands for Managing JDBC Connection Pools](#) in the *WebLogic Server Command Reference* and see [Programming WebLogic JDBC](#).

Related Information

The JDBC drivers, used in local and distributed transactions, interface with many WebLogic Server components and information appears in several documents. For example, information about JDBC drivers is included in the documentation sets for JDBC, JTA, and WebLogic jDrivers.

The following sections list Related Topics resources for JDBC, JTA, and Administration:

Administration and Management

- For a complete list of the JDBC attributes used in the configuration file (`config.xml`), see the [WebLogic Server Configuration Reference Guide](#)
- For information about using the command-line interface, see [Commands for Managing JDBC Connection Pools](#) in the *WebLogic Server Command Reference*
- For information about administering a WebLogic Server domain using the JDBC API and the JMX API for configuring JDBC objects in WebLogic Server, see [Programming WebLogic JDBC](#)

JDBC and WebLogic jDrivers

The following documentation is written primarily for application developers. Systems Administrators may want to read the introductory material as a supplement to the material in this document.

- For information about using third-party JDBC drivers, see "[Using Third-Party Drivers with WebLogic Server](#)" in *Programming WebLogic JDBC*.
- For information about using the WebLogic jDriver for Oracle, see [Configuring and Using WebLogic jDriver for Oracle](#).

Transactions (JTA)

- For information about managing JTA, see “JTA” on page 237-1.
- For information about using third-party drivers in distributed transactions, see "Using Third-Party JDBC XA Drivers with WebLogic Server" in *Programming WebLogic JTA* at [{DOCR00T}/jta/thirdpartytx.html](#).

The following documentation is written primarily for application developers. Systems Administrators may want to read the following as supplements to the material in this section.

- For information about distributed transactions, see *Programming WebLogic JTA* at [{DOCR00T}/jta/index.html](#).
- For information about using the WebLogic jDriver for Oracle/XA, see "Using WebLogic jDriver for Oracle/XA in Distributed Transactions" in *Using WebLogic jDriver for Oracle* at [{DOCR00T}/oracle/trxjdbcx.html](#).

Attributes and Console Screen Reference for JDBC

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

“Server --> Logging --> JDBC” on page 447-1

“Server --> Services --> JDBC” on page 453-1

See Also:

“Attributes and Console Screen Reference for JDBC Connection Pools” on page 128-1

“Attributes and Console Screen Reference for JDBC Data Sources” on page 142-1

“Attributes and Console Screen Reference for JDBC MultiPools” on page 150-1



JDBC Connection Pool --> Configuration --> Connections

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

In the JDBC Connection Pool—~~Configuration~~—Connections tab, you specify the number of connections in the connection pool and details for each connection in the connection pool.

A connection pool contains a group of JDBC connections that are created when the connection pool is registered, usually when starting up WebLogic Server. Your application borrows a connection from the connection pool, uses it, then returns it to the connection pool by closing it.

Tasks

[“Creating and Configuring a JDBC Connection Pool” on page 127-4](#)

[“Deploying a JDBC Connection Pool to One or More Servers or Clusters” on page 127-7](#)

[“Cloning a JDBC Connection Pool” on page 127-6](#)

[“Monitoring Connections in a JDBC Connection Pool” on page 127-36](#)

Related Topics

[“Increasing Performance with the Statement Cache” on page 127-43](#)

[“Automatically Recovering Leaked Connections” on page 127-38](#)

“Initializing Database Connections with SQL Code” on page 127-40

“Connection Testing Options” on page 127-40

[“JDBC” on page 111-1](#)

[Programming WebLogic JDBC](#)

[Third-Party Driver Configuration and Performance Requirements](#) in *Programming WebLogic JTA*

Attributes

Table 113-1

Attribute Label	Description	Value Constraints
Initial Capacity	The number of physical database connections to create when creating the connection pool. <i>MBean:</i> weblogic.management.configuration.JDBCConnectionPoolMBean <i>Attribute:</i> InitialCapacity	<i>Minimum:</i> 0 <i>Maximum:</i> 2147483647 <i>Default:</i> 1 <i>Dynamic:</i> yes
Maximum Capacity	Maximum number of physical database connections that this connection pool can contain. Different JDBC Drivers and database servers may limit the number of possible physical connections. <i>MBean:</i> weblogic.management.configuration.JDBCConnectionPoolMBean <i>Attribute:</i> MaxCapacity	<i>Minimum:</i> 1 <i>Maximum:</i> 2147483647 <i>Default:</i> 15 <i>Dynamic:</i> yes

Table 113-1

Attribute Label	Description	Value Constraints
Capacity Increment	<p>Increment by which the connection pool capacity is expanded. When there are no more available physical connections to service requests, the connection pool will create this number of additional physical database connections and add them to the connection pool. The connection pool will ensure that it does not exceed the maximum number of physical connections as set by <code>MaxCapacity</code>.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCConnectionPoolMBean</code></p> <p><i>Attribute:</i> <code>CapacityIncrement</code></p>	<p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 1</p> <p><i>Dynamic:</i> yes</p>
Statement Cache Type	<p>The algorithm used to maintain the statement cache:</p> <ul style="list-style-type: none">■ LRU - After the <code>statementCacheSize</code> is met, the Least Recently Used statement is removed when a new statement is used.■ Fixed - The first <code>statementCacheSize</code> number of statements is stored and stay fixed in the cache. No new statements are cached unless the cache is manually cleared. <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCConnectionPoolMBean</code></p> <p><i>Attribute:</i> <code>StatementCacheType</code></p>	<p><i>Default:</i> LRU</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ LRU■ FIXED <p><i>Dynamic:</i> no</p>

Table 113-1

Attribute Label	Description	Value Constraints
Statement Cache Size	<p>The number of Prepared and Callable Statements stored in the cache for further use. WebLogic Server can reuse statements in the cache without reloading them, which can increase server performance. Setting the size of the statement cache to 0 turns it off. Each connection in the pool has its own cache of statements.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCConnectionPoolMBean</code></p> <p><i>Attribute:</i> <code>StatementCacheSize</code></p>	<p><i>Default:</i> 10</p> <p><i>Dynamic:</i> yes</p>

Advanced Attributes

Table 113-2

Attribute Label	Description	Value Constraints
Login Delay	<p>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.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCConnectionPoolMBean</code></p> <p><i>Attribute:</i> <code>LoginDelaySeconds</code></p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 0</p> <p><i>Dynamic:</i> no</p>
Remove Infected Connections Enabled	<p>Controls whether a connection will be removed from the connection pool and recreated after an application uses the underlying vendor connection object.</p> <p>When set to <code>true</code>, the physical connection is not returned to the connection pool after the application closes the logical connection. Instead, the physical connection is closed and recreated.</p> <p>When set to <code>false</code>, when you close the logical connection, the physical connection is returned to the connection pool. If you use this setting, you must make sure that the database connection is suitable for reuse by other applications.</p> <p>Enabling this attribute will have an impact on performance as it will essentially disable the pooling of connections (as connections will be removed from the pool and replaced with new connections).</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCConnectionPoolMBean</code></p> <p><i>Attribute:</i> <code>RemoveInfectedConnectionsEnabled</code></p>	<p><i>Default:</i> <code>true</code></p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ <code>true</code>■ <code>false</code> <p><i>Dynamic:</i> no</p>

Table 113-2

Attribute Label	Description	Value Constraints
Allow Shrinking	<p>Indicates whether or not the pool can shrink back to its <code>InitialCapacity</code> when it is detected that connections created during increased traffic are not being used.</p> <p>When shrinking, the number of connections is reduced to the greater of either the initial capacity or the current number of connections in use.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCConnectionPoolMBean</code></p> <p><i>Attribute:</i> <code>ShrinkingEnabled</code></p>	<p><i>Default:</i> <code>true</code></p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ <code>true</code>■ <code>false</code> <p><i>Dynamic:</i> <code>yes</code></p>
Shrink Frequency	<p>Number of seconds to wait before shrinking a connection pool that has incrementally increased to meet demand. <code>ShrinkingEnabled</code> must be set to <code>true</code> for a connection pool to shrink.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCConnectionPoolMBean</code></p> <p><i>Attribute:</i> <code>ShrinkFrequencySeconds</code></p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 900</p> <p><i>Dynamic:</i> <code>yes</code></p>

Table 113-2

Attribute Label	Description	Value Constraints
Enable Connection Leak Profiling	<p>Specifies that JDBC Connection leak profiling information is gathered.</p> <p>A Connection leak occurs when a connection from the pool is not closed explicitly by calling <code>close()</code> on that connection.</p> <p>When connection leak profiling is active, the connection pool will store the stack trace at the time the Connection object is allocated from the connection pool and given to the client. When a connection leak is detected (when the Connection object is garbage collected), this stack trace is reported.</p> <p>This option is required to view leaked connections from the connection pool (right-click the connection pool name and select View Leaked Connections).</p> <p>This feature uses extra resources and will likely slow down connection pool operations, so it is not recommended for production use.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCConnectionPoolMBean</code></p> <p><i>Attribute:</i> <code>ConnLeakProfilingEnabled</code></p>	<p><i>Default:</i> false</p> <p><i>Dynamic:</i> yes</p>

Table 113-2

Attribute Label	Description	Value Constraints
Test Frequency	<p>The number of seconds between database connection tests. After every TestFrequencySeconds interval, unused database connections are tested using TestTableName. Connections that do not pass the test will be closed and reopened to re-establish a valid physical database connection. If the test fails again, the connection is closed.</p> <p>If TestTableName is not set, the test will not be performed.</p> <p>If set to 0 (the default), connections are not tested.</p> <p><i>MBean:</i> weblogic.management.configuration.JDBCConnectionPoolMBean</p> <p><i>Attribute:</i> TestFrequencySeconds</p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 0</p> <p><i>Dynamic:</i> yes</p>
Test Reserved Connections	<p>When selected (set to true), WebLogic Server tests a connection before 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).</p> <p>The attribute TestTableName must be enabled for TestConnectionsOnReserve to be effective.</p> <p>This attribute is required for connection pools used within a MultiPool that uses the High Availability algorithm.</p> <p><i>MBean:</i> weblogic.management.configuration.JDBCConnectionPoolMBean</p> <p><i>Attribute:</i> TestConnectionsOnReserve</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ true ■ false <p><i>Dynamic:</i> yes</p>

Table 113-2

Attribute Label	Description	Value Constraints
Test Created Connections	<p>When selected (set to true), WebLogic Server tests a connection after creating it and before adding it to the list of connections available to the client. The test adds a small delay in creating the connection, but ensures that the client receives a working connection (assuming that the DBMS is available and accessible). The attribute TestTableName must be set for TestConnectionsOnCreate to be effective.</p> <p><i>MBean:</i> weblogic.management.configuration.JDBCConnectionPoolMBean</p> <p><i>Attribute:</i> TestConnectionsOnCreate</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false <p><i>Dynamic:</i> yes</p>
Test Released Connections	<p>When selected (set to true), 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. The attribute TestTableName must be enabled for TestConnectionsOnRelease to be effective.</p> <p><i>MBean:</i> weblogic.management.configuration.JDBCConnectionPoolMBean</p> <p><i>Attribute:</i> TestConnectionsOnRelease</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false <p><i>Dynamic:</i> yes</p>

Table 113-2

Attribute Label	Description	Value Constraints
Connection Reserve Timeout	<p>The number of seconds after which a call to reserve a connection from the pool will timeout. When set to -1, a call will never timeout.</p> <p><i>MBean:</i> weblogic.management.configuration.JDBCConnectionPoolMBean</p> <p><i>Attribute:</i> ConnectionReserveTimeoutSeconds</p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 10</p> <p><i>Dynamic:</i> yes</p>

Table 113-2

Attribute Label	Description	Value Constraints
Connection Creation Retry Frequency	<p>The number of seconds between attempts to create database connections when the connection pool is created. If you do not set this value, connection pool creation fails if the database is unavailable. If set and if the database is unavailable when the connection pool is created, WebLogic Server will attempt to create connections in the pool again after the number of seconds you specify, and will continue to attempt to create the connections until it succeeds. When set to 0 (the default), this feature is disabled.</p> <p>Note: Do not enable connection creation retries for connection pools included in a High Availability MultiPool. Connection requests to the MultiPool will fail (not fail-over) when a connection pool in the list is dead and the number of connection requests equals the number of connections in the first connection pool, even if connections are available in subsequent connection pools in the MultiPool.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCConnectionPoolMBean</code></p> <p><i>Attribute:</i> <code>ConnectionCreationRetryFrequencySeconds</code></p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 0</p> <p><i>Dynamic:</i> yes</p>

Table 113-2

Attribute Label	Description	Value Constraints
Inactive Connection Timeout	<p>The number of seconds of inactivity after which reserved connections will forcibly be released back into the pool. When set to 0 (the default), this feature is disabled.</p> <p><i>MBean:</i> weblogic.management.configuration.JDBCConnectionPoolMBean</p> <p><i>Attribute:</i> InactiveConnectionTimeoutSeconds</p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 0</p> <p><i>Dynamic:</i> yes</p>
Maximum Waiting for Connection	<p>The maximum number of connection requests that can concurrently block waiting to reserve a connection from the pool.</p> <p><i>MBean:</i> weblogic.management.configuration.JDBCConnectionPoolMBean</p> <p><i>Attribute:</i> HighestNumWaiters</p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 2147483647</p> <p><i>Dynamic:</i> yes</p>
Maximum connections made unavailable	<p>The maximum number of connections in the pool that can be made unavailable (to an application) for purposes like refreshing the connection, etc. Note that in cases like the backend system being unavailable, this specified value could be exceeded due to factors outside the pool's control. When set to 0 (the default), this feature is disabled.</p> <p><i>MBean:</i> weblogic.management.configuration.JDBCConnectionPoolMBean</p> <p><i>Attribute:</i> HighestNumUnavailable</p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 0</p> <p><i>Dynamic:</i> yes</p>

Table 113-2

Attribute Label	Description	Value Constraints
Test Table Name	<p>The name of the table used when testing a physical database connection. The default SQL code used to test a connection is</p> <pre>"select count(*) from TestTableName"</pre> <p>The <code>TestTableName</code> 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 <code>TestTableName</code> to the name of a table that is known to have few rows, or even no rows.</p> <p>If <code>TestTableName</code> begins with "SQL ", then the rest of the string following that leading token will be taken as a literal SQL statement that will be used to test a connection.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCConnectionPoolMBean</code></p> <p><i>Attribute:</i> <code>TestTableName</code></p>	<i>Dynamic:</i> no
Init SQL	<p>The SQL code used to initialize a connection. If you specify a value for <code>Init SQL</code>, WebLogic Server will run the query when it creates a database connection. If no value is set for <code>Init SQL</code>, WebLogic Server does not run any SQL code to initialize the connection.</p> <p>Start the code with SQL followed by a space.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCConnectionPoolMBean</code></p> <p><i>Attribute:</i> <code>InitSQL</code></p>	<i>Dynamic:</i> no

Table 113-2

Attribute Label	Description	Value Constraints
Keep XA Connection Till Transaction Complete Note: This option is available only when an XA JDBC driver is used to create the physical database connections in the connection pool.	Select this option to force a connection pool to reserve a physical connection and provide the same connection to an application throughout transaction processing until the distributed transaction is complete. <i>MBean:</i> weblogic.management.configuration.JDBCConnectionPoolMBean <i>Attribute:</i> keepXAConnTillTxComplete	<i>Default:</i> false <i>Dynamic:</i> no
Need Transaction Context On Close Note: This option is available only when an XA JDBC driver is used to create the physical database connections in the connection pool.	Select this option if the XA driver used to create database connections requires a distributed transaction context when closing various JDBC objects (result sets, statements, connections, and so on). If selected, SQL exceptions that are thrown while closing JDBC objects without a transaction context will be swallowed. <i>MBean:</i> weblogic.management.configuration.JDBCConnectionPoolMBean <i>Attribute:</i> needTxCtxOnClose	<i>Default:</i> false <i>Dynamic:</i> no
New XA Connection For Commit Note: This option is available only when an XA JDBC driver is used to create the physical database connections in the connection pool.	Select this option if the XA driver used to create database connections or the DBMS requires that a dedicated XA connection be used for commit/rollback processing of distributed transactions. <i>MBean:</i> weblogic.management.configuration.JDBCConnectionPoolMBean <i>Attribute:</i> newXAConnForCommit	<i>Default:</i> false <i>Dynamic:</i> no

Table 113-2

Attribute Label	Description	Value Constraints
<p>XA End Only Once</p> <p>Note: This option is available only when an XA JDBC driver is used to create the physical database connections in the connection pool.</p>	<p>Select this option if the XA driver used to create database connections or the DBMS requires that <code>XAResource.end()</code> is called only once for each pending <code>XAResource.start()</code>. If this option is selected, the XA driver will not call <code>XAResource.end(TMSUSPEND)</code>, <code>XAResource.end(TMSUCCESS)</code> successively.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCConnectionPoolMBean</code></p> <p><i>Attribute:</i> <code>xAEndOnlyOnce</code></p>	<p><i>Default:</i> false</p> <p><i>Dynamic:</i> no</p>
<p>Keep Connection Open On Release</p> <p>Note: This option is available only when an XA JDBC driver is used to create the physical database connections in the connection pool.</p>	<p>Select this option if the XA driver used to create database connections or the DBMS requires that a logical JDBC connection be kept open while transaction processing continues (although the physical XA connection can returned to the XA connection pool).</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCConnectionPoolMBean</code></p> <p><i>Attribute:</i> <code>keepLogicalConnOpenOnRelease</code></p>	<p><i>Default:</i> false</p> <p><i>Dynamic:</i> no</p>
<p>Supports Local Transaction</p> <p>Note: This option is available only when an XA JDBC driver is used to create the physical database connections in the connection pool.</p>	<p>Select this option if the XA driver used to create physical database connections supports SQL without global transactions.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCConnectionPoolMBean</code></p> <p><i>Attribute:</i> <code>supportsLocalTransaction</code></p>	<p><i>Default:</i> false</p> <p><i>Dynamic:</i> no</p>



JDBC Connection Pool --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

In the JDBC Connection Pool—Configuration—General tab, you specify the general attributes of a connection pool necessary for connecting to a database. A connection pool contains a group of JDBC connections that are created when the connection pool is registered, usually when starting up WebLogic Server or when deploying on a new server. Your application borrows a connection from the connection pool, uses it, then returns it to the connection pool by closing it.

Tasks

[“Creating and Configuring a JDBC Connection Pool” on page 127-4](#)

[“Deploying a JDBC Connection Pool to One or More Servers or Clusters” on page 127-7](#)

[“Cloning a JDBC Connection Pool” on page 127-6](#)

[“Monitoring Connections in a JDBC Connection Pool” on page 127-36](#)

Related Topics

[“JDBC” on page 111-1](#)

[“Configuring JDBC Connection Pools” on page 127-1](#)

[“Connection Pool and Data Source Configuration Guidelines” on page 127-15](#)

Attributes

Table 114-1

Attribute Label	Description	Value Constraints
Name	The name of this connection pool. <i>MBean:</i> weblogic.management.configuration.JDBCCConnectionPoolMBean <i>Attribute:</i> Name	<i>Dynamic:</i> no
URL	The database URL used to create the connections in this Connection Pool. <i>MBean:</i> weblogic.management.configuration.JDBCCConnectionPoolMBean <i>Attribute:</i> URL	<i>Dynamic:</i> no

Table 114-1

Attribute Label	Description	Value Constraints
Driver Classname	<p>The full package name of the JDBC driver class used to create the physical connections between WebLogic Server and the DBMS for this connection pool. For example:</p> <pre>com.pointbase.jdbc.jdbcUniversalDriver</pre> <p>It must be the name of a class that implements the <code>java.sql.Driver</code> interface. Check the documentation for the JDBC driver to find the full pathname.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCConnectionPoolMBean</code></p> <p><i>Attribute:</i> <code>DriverName</code></p>	<i>Dynamic:</i> no
Properties	<p>The list of properties passed to the the JDBC Driver to use when creating physical database connections. The list consists of attribute=value tags. Enter each pair on a separate line. For example</p> <pre>user=scott server=myDB</pre> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCConnectionPoolMBean</code></p> <p><i>Attribute:</i> <code>Properties</code></p>	<i>Dynamic:</i> no
Password	<p>The database password used to create the physical database connections. This password overrides the password that you enter in the Properties field.</p> <p>See “Using the JDBC Connection Pool Assistant” on page 127-3</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCConnectionPoolMBean</code></p> <p><i>Attribute:</i> <code>Password</code></p>	<i>Encrypted:</i> yes <i>Dynamic:</i> no

Table 114-1

Attribute Label	Description	Value Constraints
Open String Password	<p>This password is used in an openString to create physical XA database connections. The value is stored in an encrypted form in the config.xml file. If set, this value overrides any password value in the openString defined in Properties.</p> <p>If your JDBC driver requires it, you must create the remainder of the openString in the properties field.</p> <p><i>MBean:</i> weblogic.management.configuration.JDBCConnectionPoolMBean</p> <p><i>Attribute:</i> XAPassword</p>	<p><i>Encrypted:</i> yes</p> <p><i>Dynamic:</i> no</p>

JDBC Connection Pool --> Control

[Tasks](#) [Related Topics](#)

Overview

On this tab, you can manually control each instance of the connection pool, including the following options:

Shrink—Shrinks the connection pool to the maximum of the currently reserved connections or the initial size. Shrinking must be enabled on the Configuration—Connections tab or this operation will fail.

Reset—Resets the database connection pool by shutting down and re-establishing all physical database connections. This also clears the statement cache for each connection in the connection pool. You can only reset a normally running connection pool.

Clear Statement Cache—Clears the cache of prepared and callable statements maintained for each connection in the pool.

Suspend—Disables the pool, suspending all operations on pool connections until the pool is re-enabled. If connections are in use, applications using a connection will get an exception.

Force Suspend—Forcibly disables the pool, suspending all operations on pool connections until the pool is re-enabled. All current users of the pool are forcibly disconnected.

Destroy—Closes all database connections and deletes the configuration from the configuration file. If any connections from the pool are currently in use, the operation will fail.

Force Destroy—Forcibly destroys the pool. Forcibly disconnects all current users of the pool. Closes all database connections and deletes the configuration from the configuration file.

Resume—Restores all access to and operations on the pool. Only available after the connection pool is successfully suspended.

Note: You cannot restart a connection pool from this page that did not start correctly at server startup or when deploying the connection pool. Instead you must undeploy and redeploy the pool on the target servers and clusters on which the pool did not start correctly. See [“Restarting a JDBC Connection Pool” on page 127-34](#).

Tasks

[“Shrinking a JDBC Connection Pool” on page 127-32](#)

[“Resetting All Connections in a JDBC Connection Pool” on page 127-31](#)

[“Suspending a JDBC Connection Pool” on page 127-32](#)

[“Resuming a JDBC Connection Pool” on page 127-33](#)

[“Shutting Down a JDBC Connection Pool” on page 127-34](#)

[“Restarting a JDBC Connection Pool” on page 127-34](#)

[“Destroying or Deleting a JDBC Connection Pool” on page 127-35](#)

[“Clearing the Statement Cache for a JDBC Connection Pool” on page 127-36](#)

Related Topics

[“Managing JDBC Connection Pools” on page 127-30](#)

[“Configuring JDBC Connection Pools” on page 127-1](#)

JDBC Connection Pool --> Testing

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

In the JDBC Connection Pool—Testing tab, you can test a JDBC connection in a connection pool on each server on which the connection pool is deployed.

When you test a connection pool, WebLogic Server reserves and releases a connection from the connection pool.

Note: To make the test more meaningful, make sure that Test Reserved Connections or Test Released Connections is selected on the Configuration—Connections tab (under Advanced Options). If either of these options is selected, WebLogic Server not only reserves and releases a connection, but also tests the physical database connection. See [Test Reserved Connections](#) in [Attributes](#).

Tasks

[“Testing a JDBC Connection Pool” on page 127-30](#)

[“Creating and Configuring a JDBC Connection Pool” on page 127-4](#)

[“Deploying a JDBC Connection Pool to One or More Servers or Clusters” on page 127-7](#)

[“Monitoring Connections in a JDBC Connection Pool” on page 127-36](#)

Related Topics

[“Configuring JDBC Connection Pools” on page 127-1](#)

[“Managing JDBC Connection Pools” on page 127-30](#)

Attributes

Server Name—The name of the server on which the connection pool is deployed. Click the Test Pool button for this server to test a connection from this instance of the connection pool.

Pool State—The current state of the instance of the connection pool: RUNNING, SUSPENDED, or UNAVAILABLE. Each instance of the connection is independent and contains its own physical database connections.

Test Pool—Click this button to test a connection from the selected instance of the connection pool. Make sure that Test Reserved Connections or Test Released Connections is selected on the Configuration—Connections tab (under Advanced Options). Test results are displayed at the top of the pane.

Test pool on all servers—Click this button to test a connection from all instances of the connection pool. Test results are displayed at the top of the pane.

JDBC Connection Pool --> Monitoring

[Tasks](#) [Related Topics](#)

Overview

From the JDBC Connection Pool—Monitoring tab, you can view information about the state of each deployed instance of the selected connection pool. That is, for each server on which the connection pool is deployed, you can see current status information about the connection pool. By default, the table of information includes the following columns. You can click [Customize this view](#) to display additional columns.

Server—Server on which the connection pool is deployed.

State—Current state of the connection pool. Connection pool status relies on connection testing to determine if the connection pool is **RUNNING** or **UNHEALTHY**. You must enable connection testing (on create, on reserve, on release, or periodic testing) to make the State value meaningful. Connection pool state can be:

- **RUNNING** - Normal state of the connection pool; the connection pool is enabled (deployed and not **SUSPENDED**).
- **SUSPENDED** - Connection pool is disabled.
- **SHUTDOWN** - Connection pool is shutdown and all database connections have been closed.
- **UNKNOWN** - Connection pool state is unknown.
- **UNHEALTHY** - All connections are unavailable. This state occurs if the database server is unavailable when the connection pool is created (creation retry must be enabled) or if all connections have failed connection tests (on creation, on reserve, on release, or periodic testing).

Connections—Number of physical database connections from this instance of the connection pool *that are currently in use*.

Waiters—Current number of application requests waiting for a connection.

Num Unavailable—Current number of connections that are currently unavailable to applications because the connection is being tested or refreshed.

Additional columns include:

Active Connections Average Count—Average number of active connections in this instance of the JDBC connection pool.

Connection Delay Time—Average time in milliseconds it takes to create a physical database connection. The time is calculated as the sum of the time it takes to create all physical database connections in the connection pool divided by the total number of connections created.

Connections High—Highest number of *active* database connections in this instance of the connection pool since the connection pool was instantiated. Active connections are connections in use by an application.

Connections Total—Total number of database connections created in this instance of the connection pool since the connection pool was instantiated.

Curr Capacity—Current number of database connections in this instance of the connection pool. The number includes available connections, connections in use by applications, and connections unavailable for testing.

Driver Version—Version of the JDBC driver used in the physical database connections in the connection pool. The driver is displayed as the driver class name followed by the major and minor release number of the driver.

Failures To Reconnect Count—Number of times when this instance of the connection pool attempted to refresh a connection to a database and failed. Failures occur when the database is unavailable or when the network fails.

Highest Num Unavailable—Highest number of unavailable connections in this instance of the connection pool since the connection pool was instantiated.

Max Capacity—Maximum number of database connections that this connection pool can contain.

Wait Seconds High—Highest number of seconds that an application waited for a connection from this instance of the connection pool since the connection pool was instantiated.

Waiters High—Highest number of application requests concurrently waiting for a connection from this instance of the connection pool.

Tasks

[“Monitoring Connections in a JDBC Connection Pool” on page 127-36](#)

[“Creating and Configuring a JDBC Connection Pool” on page 127-4](#)

Related Topics

[“Managing JDBC Connection Pools” on page 127-30](#)

[“Connection Testing Options” on page 127-40](#)



JDBC Connection Pool --> Notes

[Tasks](#) [Related Topics](#)

Overview

You can use the JDBC Connection Pool—Notes tab to store notes about the selected connection pool.

Tasks

[“Adding a Note to a JDBC Connection Pool” on page 127-8](#)

[“Creating and Configuring a JDBC Connection Pool” on page 127-4](#)

Related Topics

[“Managing JDBC Connection Pools” on page 127-30](#)

[“JDBC Connection Pools” on page 127-1](#)



JDBC Connection Pool --> Target and Deploy

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

On this tab, you select the servers and clusters on which you want to deploy the connection pool. When you deploy a connection pool on a server, WebLogic Server creates an instance of the connection pool on the server. If you deploy the connection pool on two servers, WebLogic Server creates the connection pool on each server. Each instance is configured the same, but is independent of the other. If you deploy the connection pool to a cluster, WebLogic Server creates an instance of the connection pool on each server in the cluster unless you specify otherwise.

Tasks

“Deploying a JDBC Connection Pool to One or More Servers or Clusters” on page 127-7

“Shutting Down a JDBC Connection Pool” on page 127-34

“Creating and Configuring a JDBC Connection Pool” on page 127-4

Related Topics

“Configuring JDBC Connection Pools” on page 127-1

“Managing JDBC Connection Pools” on page 127-30

“Configuring JDBC DataSources” on page 141-1

“Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1

Attributes

Independent Servers—Lists the servers in your configuration on which you can deploy the connection pool. Select servers and click Apply.

Clusters—Lists clusters and comprising servers on which you can deploy the connection pool. Select the cluster on which you want to deploy the connection pool and click Apply. In almost all cases, you should deploy to an entire cluster, not individual servers in a cluster. When you deploy the connection pool to a cluster, WebLogic Server creates an instance of the connection pool on each server in the cluster.

JDBC Connection Pools

A JDBC connection pool contains a group of JDBC connections that are created when the connection pool is registered, usually when starting up WebLogic Server. Your application borrows a connection from the connection pool, uses it, then returns it to the connection pool by closing it.

After creating a JDBC connection pool, you can create a data source that applications can use to access the JDBC connection pool.

The main JDBC Connection Pools page contains two tabs:

- **Configuration**—Which provides a link for creating a connection pool and lists currently configured connection pools.
- **Monitoring**—Which displays information about the state of each deployed instance of all connection pools. By default, the table includes the following columns. You can click Customize this view to display additional columns.

Server—The server on which the connection pool is deployed.

State—The current state of the connection pool: **RUNNING** or **SUSPENDED**.

Connections—The current number of physical database connections in the connection pool.

Waiters—The current number of applications waiting for a connection.

Num Unavailable—The current number of connections that are currently unavailable to applications because the connection is being tested or refreshed.

To create a JDBC connection pool, click the Configure a new JDBC Connection Pool link on the Configuration tab.

- To create a JDBC connection pool, see [“Creating and Configuring a JDBC Connection Pool” on page 127-4](#).
- To create a JDBC data source, see [“Creating and Configuring a JDBC Data Source” on page 141-5](#).



Configure a JDBC Connection Pool --> Test Database Connection

[Tasks](#) [Related Topics](#)

Overview

On the Configure a JDBC Connection Pool --> Test Database Connection page, the JDBC Connection Pool Assistant assembles the information you provided in earlier steps into standard connection attributes—attributes typically used to create a database connection. You should do the following:

1. Review the information displayed to make sure it is correct. You can edit any errors or provide additional properties as necessary. The JDBC Connection Pool Assistant will save your configuration changes when you click the Test Driver Configuration or Skip this Step buttons.
2. Test the driver configuration or skip the test. When you test the driver configuration, the JDBC Connection Pool Assistant attempts to establish a connection from the administration server to the database using the JDBC driver you selected and using all of the parameters displayed on the page.

Skip this step if the JDBC driver is not installed on the administration server.

If the test fails, you should do one of the following:

- Fix any connection property errors on the page and try the test again.
- Skip this step to continue creating the configuration. You will need to trouble-shoot the problem again later. The connection problem may not be related to the connection properties.

The Configure a JDBC Connection Pool --> Test Database Connection page includes the following options:

Driver Classname—The full package name of JDBC driver class used to create the physical database connections in the connection pool.

URL—The database URL used to create the connections in this JDBC connection pool. The format and information included in the URL varies greatly between DBMSs and JDBC drivers. For some XA drivers, the URL can be blank. Other JDBC drivers require additional information in the URL, such as the host name and port and the database username and password.

Database User Name—The user name to use to connect to the database. In most cases, the user account must be established on the database server by you database administrator.

Password—The password for the Database User Name to use to connect to the database.

Properties—Additional properties required by the DBMS or JDBC driver, such as an `openString` for Oracle or an `informixServer` for Informix.. If you need to enter additional properties, enter each additional property on a separate line as a `key=value` pair. See the documentation from the JDBC driver vendor for more details.

Tasks

[“Creating and Configuring a JDBC Connection Pool” on page 127-4](#)

Related Topics

[“Configuring JDBC Connection Pools” on page 127-1](#)

[Overview of Third-Party JDBC Drivers](#)

[“Connection Pool and Data Source Configuration Guidelines” on page 127-15](#)

[“Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1](#)

Configure a JDBC Connection Pool --> Choose database

[Tasks](#) [Related Topics](#)

Overview

The JDBC Connection Pool Assistant is designed to help you configure a pool of connections from your database. It lists many drivers for many database types and prompts you for connection properties for the JDBC driver you select.

The Configure a JDBC Connection Pool Assistant --> Choose Database page includes the following options:

Database type—Select the DBMS of the database that you want to connect to. If your DBMS is not listed, select *Other*.

Database driver—Select the JDBC driver you want to use to connect to the database. The list includes common JDBC drivers for the selected database type. The driver you select will affect choices that you see in the next step in the JDBC Connection Pool Assistant. Select an XA driver if you want connections from the connection pool to participate in distributed transactions.

Notes: JDBC drivers listed in the JDBC Connection Pool Assistant are not necessarily certified for use with WebLogic Server. In keeping with the goal of the JDBC Connection Pool Assistant, JDBC drivers are listed as a convenience to help you create a connection to many of the database management systems available.

You must install JDBC drivers in order to use them to create database connections in a connection pool. Drivers are listed in the JDBC Connection Pool Assistant with known required configuration options to help you configure a connection pool. The JDBC drivers in the list are not necessarily installed. Driver installation can include setting system Path, Classpath, and other environment variables.

When a JDBC driver is updated, configuration requirements may change. The JDBC Connection Pool Assistant uses known configuration requirements at the time the WebLogic Server software was released. If configuration options for your JDBC driver have changed, you may need to manually override the configuration options displayed in Test Database Connection step of the JDBC Connection Pool Assistant or in the property pages for the connection pool.

Tasks

[“Creating and Configuring a JDBC Connection Pool” on page 127-4](#)

Related Topics

[“Configuring JDBC Connection Pools” on page 127-1](#)

[Overview of Third-Party JDBC Drivers](#)

[“Connection Pool and Data Source Configuration Guidelines” on page 127-15](#)

[“Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1](#)

Configure a JDBC Connection Pool --> Define and test connection

[Tasks](#) [Related Topics](#)

Overview

The Configure a JDBC Connection Pool --> Define and Test Connection page includes the following attributes. See the documentation from your JDBC driver vendor for the correct format for each of these attributes.

Name—Enter a name for the connection pool. This name will be used to identify the connection pool in the configuration file and will be used when creating a data source.

Driver Classname—Enter the full package name of JDBC driver class used to create the physical database connections in the connection pool.

URL—Enter the database URL used to create the connections in this JDBC connection pool. The format and information included in the URL varies greatly between DBMSs and JDBC drivers. For some XA drivers, the URL can be blank. Other JDBC drivers require additional information in the URL, such as the host name and port and the database username and password.

Database User Name—Enter the user name to use to connect to the database. In most cases, the user account must be established on the database server by you database administrator.

Password—Enter the password for the Database User Name to use to connect to the database.

Properties—Enter additional properties required by the DBMS or JDBC driver, such as an `openString` for Oracle or an `informixServer` for Informix.. Enter each property on a separate line as a `key=value` pair.

Tasks

[“Creating and Configuring a JDBC Connection Pool” on page 127-4](#)

Related Topics

[“Connection Pool and Data Source Configuration Guidelines” on page 127-15](#)

[“Configuring JDBC Connection Pools” on page 127-1](#)

[Overview of Third-Party JDBC Drivers](#)

[“Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1](#)

JDBC Connection Pool Assistant --> Create and deploy

[Tasks](#) [Related Topics](#)

Overview

On the Configure a JDBC Connection Pool --> Create and Deploy page, you select the servers and clusters on which you want to create and deploy the new connection pool.

If only there is only one server in your configuration, the JDBC Connection Pool Assistant automatically selects it, and does not display a list of potential deployment targets.

After you select the servers and clusters on which you to create the connection pool, click the Create and Deploy button. WebLogic Server adds the connection pool to the configuration (`config.xml`) and creates the connection pool with physical database connections on each server that you select.

If an error message is displayed, the connection pool configuration is saved even though the connection pool may not actually be deployed on all targets. You should check for proper JDBC driver installation for each server. Also check the server window (standard out) and the JDBC log for more information to help you trouble-shoot problems. You can deploy the connection pool again from the connection pool property pages. See [“Deploying a JDBC Connection Pool to One or More Servers or Clusters”](#) on page 127-7.

Tasks

[“Creating and Configuring a JDBC Connection Pool”](#) on page 127-4

Related Topics

[“Configuring JDBC Connection Pools” on page 127-1](#)

[“JDBC Objects in a Cluster” on page 111-3](#)

[“Monitoring Connections in a JDBC Connection Pool” on page 127-36](#)

[Overview of Third-Party JDBC Drivers](#)

[“Connection Pool and Data Source Configuration Guidelines” on page 127-15](#)

[“Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1](#)

Configure a JDBC Connection Pool --> Define connection properties

[Tasks](#) [Related Topics](#)

Overview

In the Configure a JDBC Connection Pool --> Define Connection Properties page, you select and define options for each of the database connections to create in the connection pool. Properties and options available will vary depending on the JDBC driver you selected in the previous step in the JDBC Connection Pool Assistant. The page can include the following options:

Database Name—The name of the database instance that you want to connect to. For an Oracle database, this would be the Oracle system identifier (SID).

Host Name—The name or IP address of the database server.

Port—The port that the database server uses to communicate with clients. In many cases, a typical default value is provided, based on the database type (DBMS) that you selected in Step 1.

Database User Name—The user name to use to connect to the database. In most cases, the user account must be established on the database server by you database administrator.

Password—The password for the Database User Name to use to connect to the database.

There may be additional connection properties on this page, depending on your DBMS.

Tasks

[“Creating and Configuring a JDBC Connection Pool” on page 127-4](#)

Related Topics

[“Configuring JDBC Connection Pools” on page 127-1](#)

[Overview of Third-Party JDBC Drivers](#)

[“Connection Pool and Data Source Configuration Guidelines” on page 127-15](#)

[“Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1](#)

Active JDBC Connections

The Active JDBC Connections table shows information for all active JDBC connection pools assigned to the server or for the selected connection pool. You can click column headings in the table to sort the information in the table. You can also click Customize this view to select the columns to display in the table.

If there is no table of connections on the page, then there are no active connection pools or the selected connection pool is inactive. To activate a connection pool, assign it to a server.

- To create a JDBC connection pool, see [“Creating and Configuring a JDBC Connection Pool” on page 127-4](#).
- To assign a connection pool to a server or cluster, see [“Deploying a JDBC Connection Pool to One or More Servers or Clusters” on page 127-7](#).
- To create a JDBC data source, see [“Creating and Configuring a JDBC Data Source” on page 141-5](#).



1 JDBC Connection Pools

[“Attributes and Console Screen Reference for JDBC Connection Pools” on page 128-1]

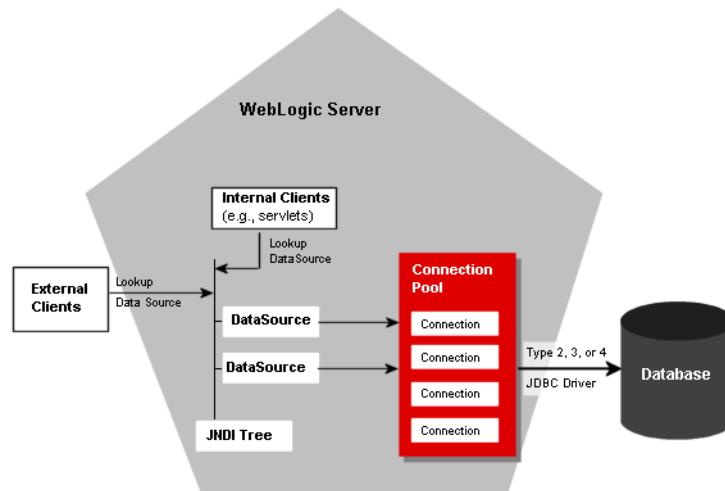
The following sections explain how to configure and manage JDBC connection pools in the Administration Console:

- “Configuring JDBC Connection Pools” on page 127-1
- “Using the JDBC Connection Pool Assistant” on page 127-3
- “Application-Scoped JDBC Data Sources and Connection Pools” on page 127-9
- “Connection Pool and Data Source Configuration Guidelines” on page 127-15
- “Security for JDBC Connection Pools” on page 127-30
- “Managing JDBC Connection Pools” on page 127-30
- “Monitoring Connections in a JDBC Connection Pool” on page 127-36
- “Tuning Connection Pools” on page 127-37

Configuring JDBC Connection Pools

A connection pool contains a group of JDBC connections that are created when the connection pool is registered—when starting up WebLogic Server or when deploying the connection pool to a target server or cluster. Connection pools use a JDBC driver to create physical database connections. Your application borrows a connection from the pool, uses it, then returns it to the pool by closing it.

Figure 127-1 Connection Pool Architecture



All of the settings you make with the Administration Console are static; that is, all settings persist even after you stop and restart WebLogic Server. You can create dynamic connection pools—those that you expect to use and delete while the server is running—using the command line (see [Commands for Managing JDBC Connection Pools](#) in the *WebLogic Server Command Reference* at [{DOCRROOT}/admin_ref/cli.html#jdbc](#)) or programmatically using the API (see [Creating a Connection Pool Dynamically](#) in *Programming WebLogic JDBC* at [{DOCRROOT}/jdbc/programming.html#dynamic_conn_pool](#)).

Connection pool settings are persisted in the `config.xml` file, including settings for dynamically created connection pools (until you programmatically delete the connection pool). For information about entries in the `config.xml` file, see the [JDBCConnectionPool](#) section of the *Configuration Reference Guide* at [{DOCRROOT}/config_xml/JDBCConnectionPool.html](#).

Related Information

- “Creating and Configuring a JDBC Connection Pool” on page 127-4
- “JDBC DataSources” on page 141-1
- [Connection Pools](#) in *Programming WebLogic JDBC*.

Using the JDBC Connection Pool Assistant

You use the JDBC Connection Pool Assistant to create JDBC connection pools. The JDBC Connection Pool Assistant helps you create and deploy a connection pool by prompting you for database and driver information and then constructing the connection attributes required by your JDBC driver, such as the driver class name and the database URL.

When you create a connection pool with the JDBC Connection Pool Assistant, many attributes for the connection pool are set with the default value. You may need to change connection pool settings to suit your environment. For example, you may need to increase the maximum number of connections available in the connection pool if your application consistently cannot reserve a connection because all connections in the connection pool are in use.

Notes: JDBC drivers listed in the JDBC Connection Pool Assistant are not necessarily certified for use with WebLogic Server. In keeping with the goal of the JDBC Connection Pool Assistant, JDBC drivers are listed as a convenience to help you create a connection to many of the database management systems available.

You must install JDBC drivers in order to use them to create database connections in a connection pool on each server on which the connection pool is deployed. Drivers are listed in the JDBC Connection Pool Assistant with known required configuration options to help you configure a connection pool. The JDBC drivers in the list are not necessarily installed. Driver installation can include setting system Path, Classpath, and other environment variables. See [Setting the Environment for Your Third-Party JDBC Driver](#) in *Programming WebLogic JDBC*.

When a JDBC driver is updated, configuration requirements may change. The JDBC Connection Pool Assistant uses known configuration requirements at the time the WebLogic Server software was released. If configuration options for your JDBC driver have changed, you may need to manually override the configuration options displayed in Step 3 of the JDBC Connection Pool Assistant or in the property pages for the connection pool after the pool is created.

Creating and Configuring a JDBC Connection Pool

1. Click to expand the Services and JDBC nodes.
2. Right-click the Connection Pools node and select Configure a New JDBC Connection Pool. The JDBC Connection Pool Assistant opens in the right pane.
3. In Choose Database, follow these steps:
 - a. Database type, select the DBMS of the database that you want to connect to. If your DBMS is not listed, select Other.
 - b. In Database driver, select the JDBC driver you want to use to connect to the database. The list includes common JDBC drivers for the selected DBMS. Click Continue.

Note: You must install JDBC drivers in order to use them to create database connections in a connection pool. Drivers are listed in the JDBC Assistant with known required configuration options to help you configure a connection pool. Driver installation also includes setting system Path, Classpath, and other environment variables.
4. In Define Connection Properties, follow these steps:
 - a. In Name, enter a name for the new connection pool. The name should be unique within the domain.
 - b. Under Connection Properties, provide the information requested. The required attributes vary by the DBMS and JDBC driver you selected in the previous step. Many attributes include a common default value. Verify these values for your environment.
 - c. Click Continue.
5. In Test Database Connection, verify the connection properties and then click Test Driver Configuration. WebLogic Server attempts to load the driver and create a direct connection to the database server using the connection properties you provided. The JDBC driver must be installed and configured on the server (on the Administration server in multi-server environments) for the test to succeed.

If the test is successful, the Create and Deploy page is displayed. If the test is unsuccessful, an error message is displayed at the top of the page. Check the values on the page and correct any errors, then test the connection again.

You can click Skip this Step to skip the test and continue configuring the connection pool. Note that if you create and deploy a connection pool with errors, the connection pool configuration will be created, but the connection pool will not actually be deployed to servers or clusters. Also, when you restart servers, the servers will start with errors.

6. In Create and Deploy, select the servers and clusters on which you want to deploy the connection pool. If you only have one server in your domain, the connection pool is automatically deployed to the server. Click Create and Deploy to complete the process.

In most cases, you should create a data source to use with a connection pool. To create a data source, see “Creating and Configuring a JDBC Data Source” on page 141-6.

Database Passwords in Connection Pool Configuration

When you create a connection pool, you typically include at least one password to connect to the database. If you use an open string to enable XA, you may use two passwords. You can enter the passwords as a name-value pair in the `Properties` field or you can enter them in their respective fields:

- **Password.** Use this field to set the database password. This value overrides any password value defined in the `Properties` passed to the tier-2 JDBC Driver when creating physical database connections. The value is encrypted in the `config.xml` file (stored as the `Password` attribute in the `JDBCConnectionPool` tag) and is hidden on the administration console.
- **Open String Password.** Use this field to set the password in the open string that the transaction manager in WebLogic Server uses to open a database connection. This value overrides any password defined as part of the open string in the `Properties` field. The value is encrypted in the `config.xml` file (stored as the `XAPassword` attribute in the `JDBCConnectionPool` tag) and is hidden on the Administration Console. At runtime, WebLogic Server reconstructs the open string with the password you specify in this field. The open string in the `Properties` field should follow this format:

```
openString=Oracle_XA+Acc=P/userName/+SesTm=177+DB=dbHost+Thread  
s=true=Sqlnet=dvi0+logDir=.
```

Note that after the `userName` there is no password.

If you specify a password in the `Properties` field when you first configure the connection pool, WebLogic Server removes the password from the `Properties` string and sets the value as the `Password` value in an encrypted form the next time you start WebLogic Server. If there is already a value for the `Password` attribute for the connection pool, WebLogic Server does not change any values. However, the value for the `Password` attribute overrides the password value in the `Properties` string. The same behavior applies to any password that you define as part of an open string. For example, if you include the following properties when you first configure a connection pool:

```
user=scott;
password=tiger;
openString=Oracle_XA+Acc=p/scott/tiger+SesTm=177+db=dbHost+Threads=true+Sqlnet=lcs817+logDir=.+dbgFl=0x15;server=dbHost
```

The next time you start WebLogic Server, it moves the database password and the password included in the open string to the `Password` and `Open String Password` attributes, respectively, and the following value remains for the `Properties` field:

```
user=scott;
openString=Oracle_XA+Acc=p/scott/+SesTm=177+db=dbHost+Threads=true+Sqlnet=lcs817+logDir=.+dbgFl=0x15;server=dbHost
```

After a value is established for the `Password` or `Open String Password` attributes, the values in these attributes override the respective values in the `Properties` attribute. That is, continuing with the previous example, if you specify `tiger2` as the database password in the `Properties` attribute, WebLogic Server ignores the value and continues to use `tiger` as the database password, which is the current encrypted value of the `Password` attribute. To change the database password, you must change the `Password` attribute.

Note: The value for `Password` and `Open String Password` do not need to be the same.

Cloning a JDBC Connection Pool

1. Click to expand the `Services`, `JDBC`, and `Connection Pool` nodes.
2. Right-click the connection pool you want to clone and select `Clone poolname`. A dialog displays in the right pane showing the tabs associated with cloning a connection pool. All attribute values except `Name` are the same as those in cloned pool, including `Connection` attributes and deployment targets.

3. Enter a new Name. Optionally, you can modify the URL, Driver Classname, and Properties attribute fields. For more information about connection pool general attributes, see [Attributes](#).
4. Click Clone to create a connection pool with the attributes you specified on the General tab and with cloned values on all other tabs. The new connection pool is added under the Connection Pools node in the left pane.
5. Optionally, click the remaining tabs for the connection pool and change the attribute fields. Click Apply to save any changes you make.

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

1. In the left pane, click to expand the Services, JDBC, and Connection Pools nodes to display the list of connection pools in the current domain.
2. Click the connection pool that you want to deploy. A dialog displays in the right pane showing the tabs associated with this instance.
3. Click the Target and Deploy tab and select the servers or clusters on which you want to deploy the connection pool. Click Apply to save your changes.

When deploying a JDBC connection pool on a cluster, in most cases you should deploy the connection pool to the entire cluster. You should deploy the related data source to the same targets.

Configuring the Statement Cache for a JDBC Connection Pool

On the JDBC Connection Pool—Configuration—Connections tab, you can configure statement cache attributes for a connection pool. For more information about the statement cache, see “Increasing Performance with the Statement Cache” on page 127-43. To configure the statement cache, follow these steps:

1. In the left pane, click to expand the Services, JDBC, and Connection Pool nodes to display the list of connection pools in the current domain.

2. Click the connection pool that you want to configure. A dialog displays in the right pane showing the tabs associated with this instance.
3. Click the Configuration tab, then click the Connections tab.
4. In Statement Cache Type, select one of the following options:
 - LRU - After the `statementCacheSize` is met, the Least Recently Used statement is removed when a new statement is used.
 - Fixed - The first `statementCacheSize` number of statements is stored and stay fixed in the cache. No new statements are cached unless the cache is manually cleared or the cache size is increased.

See “Statement Cache Algorithms” on page 127-44 for more information.
5. In Statement Cache Size, enter the number of statements to cache per connection per connection pool instance. The default value is 10. See “Statement Cache Size” on page 127-45 for more information.
6. Click Apply to save your changes.

Adding a Note to a JDBC Connection Pool

1. In the left pane, click to the JDBC node to expand it.
2. Click the Connection Pools node to expand it and show the list of connection pools defined in your domain.
3. Click the connection pool to which you want to add a note. A dialog displays in the right pane showing tabs with attributes for the connection pool.
4. Click the Notes tab. Type the note in the Notes field.
5. Click Apply to save your changes.

Application-Scoped JDBC Data Sources and Connection Pools

When you package your enterprise applications, you include the `weblogic-application.xml` supplemental deployment descriptor, which you use to configure *application scoping*. Within the `weblogic-application.xml` file, you can configure JDBC connection pools and associated data sources that are created when you deploy the enterprise application.

Data sources and connection pools created in this manner are known as *application-scoped connection pools*, *app scoped pools*, *application local pools*, *app local pools*, or *local pools*, and are scoped for the enterprise application only. That is, they are isolated for use by the enterprise application. You deploy each application-scoped connection pool/data source pair as a module in the application.

Each application scoped connection pool that you create must reference a data source factory. The data source factory creates the application-scoped data source and underlying connection pool. Values in the `weblogic-application.xml` can override the default values provided in the data source factory.

An instance of the data source and connection pool is created with each instance of your application (unless you deploy modules individually). This means an instance of the pool is created with the application on each server that the application is targeted to. It is important to keep this in mind when considering connection pool sizing. For example if each instance of the connection pool has 10 database connections, and the application is deployed on 10 servers, your domain will have 100 connections to the database. You may need to consider database limits, including the maximum number of open cursors.

For more information about application scoping and application scoped resources, see:

- [Configuring and Using Application-Scoped JDBC Connection Pools](#)
- [Overview of Application Scoping](#)
- [weblogic-application.xml Deployment Descriptor Elements](#)
- [Quickstart Guide to Deploying Modules](#)
- [Two-Phase Deployment](#)

Configuring Application-Scoped Connection Pool Attributes

If you deploy your enterprise application as an archive (.EAR, .WAR, .JAR, or .RAR extension), you configure and make all changes to the application-scoped connection pool for the application in the `weblogic-application.xml` deployment descriptor. However, if you deploy your application in an *exploded* archive directory, you can make configuration changes to an application-scoped connection pool in the Administration Console. Changes are made directly in the `weblogic-application.xml` deployment descriptor and take effect immediately. You do not have to redeploy the application module (application-scoped connection pool).

To change configuration attributes for an application-scoped connection pool (deployed in an exploded archive):

1. In the left pane, click to expand Deployments, Applications, and the deployed application name to display a list of application components.
2. Select the application-scoped data source from the list of application components. In the right pane, tabs display with attributes for the application-scoped connection pool associated with the application-scoped data source.
3. Select the Configuration tab, then select the Descriptor tab.
4. Make changes as necessary, and click Apply to save your changes. For more information about each attribute that you can change, see “Attributes” on page 23-2.

Deploying Application-Scoped Connection Pools

When you deploy an application in an environment with several managed servers or one or more clusters, you can choose to deploy all components at once or choose to deploy application components individually. For the latter option, make sure to deploy application-scoped data sources (which includes the associated connection pools) to the same deployment targets as other application components that will use database

connections from the connection pool. For example, if Application Module 1 is deployed on Managed Server 1 and Application Module 1 uses connections from Data Source 1, make sure that Data Source 1 is deployed on Managed Server 1.

After you deploy an application-scoped data source, application components access connections from the application-scoped connection pool by looking up the application-scoped data source on the local JNDI tree at `java:comp/env`. The data source must be available on the local server.

Selecting Deployment Targets for an Application-Scoped Connection Pool

To change deployment targets for an application-scoped data source and connection pool:

1. In the left pane, click to expand Deployments, Applications, and the deployed application name to display a list of application components.
2. Select the application-scoped data source from the list of application components. In the right pane, tabs display with attributes for the application-scoped connection pool associated with the application-scoped data source.
3. Select the Targets tab, then select the servers and clusters on which you want to deploy the application-scoped data source and connection pool. Select the same targets as other application modules that use connections from the pool.

Stopping and Redeploying an Application-Scoped Connection Pool

On the Deploy tab, you can view the status of each deployed instance of an application-scoped connection pool. For example, if your application-scoped connection pool is deployed on three servers in your domain, you will see each deployment target listed and the status of the connection pool on that target. You can stop or redeploy an active connection pool or deploy an inactive connection pool. When you stop a connection pool, all database connections are closed. When you redeploy a connection pool, all connections are closed and the connection pool is recreated, which includes recreating the initial number of database connections.

Note: To add deployment targets, select additional targets on the Targets tab. See “Selecting Deployment Targets for an Application-Scoped Connection Pool” on page 127-11.

To stop, redeploy, or deploy and application-scoped connection pool:

1. In the left pane, click to expand Deployments, Applications, and the deployed application name to display a list of application components.
2. Select the application-scoped data source from the list of application components. In the right pane, tabs display with attributes for the application-scoped connection pool associated with the application-scoped data source.
3. Select the Deploy tab, which lists each deployed instance of the connection pool and its status.
4. Choose from the following options:
 - Click Stop to stop the selected deployed instance of the connection pool.
 - Click Redeploy to stop the selected deployed instance of the connection pool and then recreate the connection pool with the initial number of connections.
 - Click Deploy to start an inactive deployed instance of a connection pool.

Monitoring Application-Scoped Connection Pools

To monitor an application-scoped connection pool:

1. In the left pane, click to expand Deployments, Applications, and the deployed application name to display a list of application components.
2. Select the application-scoped data source from the list of application components. In the right pane, tabs display with attributes for the application-scoped connection pool associated with the application-scoped data source.
3. Select the Monitoring tab. A table displays with information about connections in the selected connection pool.

Manually Administering an Application-Scoped Connection Pool

On the Control tab for an application-scoped connection pool, you can manually shrink or reset the connection pool or clear the statement cache for the connection pool. In many cases, shrink and refresh operations, as well as statement cache management, are handled automatically by WebLogic Server based on settings for the connection pool.

Shrinking an Application-Scoped Connection Pool

1. In the left pane, click to expand Deployments, Applications, and the deployed application name to display a list of application components.
2. Select the application-scoped data source from the list of application components. In the right pane, tabs display with attributes for the application-scoped connection pool associated with the application-scoped data source.
3. Select the Control tab. A table displays with a list of deployed instances of the connection pool.
4. Click Shrink for the instance of the connection pool that you want to shrink. When you shrink a connection pool, the server reduces the number of database connections in the connection pool to the greater of either the initial number of connections or the current number of connections in use.

Resetting an Application-Scoped Connection Pool

1. In the left pane, click to expand Deployments, Applications, and the deployed application name to display a list of application components.
2. Select the application-scoped data source from the list of application components. In the right pane, tabs display with attributes for the application-scoped connection pool associated with the application-scoped data source.
3. Select the Control tab. A table displays with a list of deployed instances of the connection pool.
4. Click Reset for the instance of the connection pool that you want to reset. When you reset a connection pool, the server closes and reopens all database connections in the connection pool, including any connections in use.

Clearing the Statement Cache for an Application-Scoped Connection Pool

1. In the left pane, click to expand Deployments, Applications, and the deployed application name to display a list of application components.
2. Select the application-scoped data source from the list of application components. In the right pane, tabs display with attributes for the application-scoped connection pool associated with the application-scoped data source.
3. Select the Control tab. A table displays with a list of deployed instances of the connection pool.
4. Click Clear Statement Cache for the instance of the connection pool for which you want to clear the statement cache.

Testing an Application-Scoped Connection Pool

On the Testing tab, you can test a JDBC connection in a connection pool on each server on which the connection pool is deployed.

When you test a connection pool, WebLogic Server reserves and releases a connection from the connection pool.

To make the test more meaningful, make sure that Check On Reserve Enabled or Check On Release Enabled is selected on the Configuration—Descriptor tab. If either of these options is selected, WebLogic Server not only reserves and releases a connection, but also tests the physical database connection. See [Check On Reserve Enabled](#) in [Attributes](#).

For more information about connection testing and configuration options, see “Connection Testing Options” on page 127-40.

To test a connection in an application-scoped connection pool, follow these steps:

1. In the left pane, click to expand Deployments, Applications, and the deployed application name to display a list of application components.
2. Select the application-scoped data source from the list of application components. In the right pane, tabs display with attributes for the application-scoped connection pool associated with the application-scoped data source.

3. Click the Testing tab. The Testing tab displays a list of instances of the selected connection pool. Each server on which the connection pool is deployed is listed. Each server can have only one instance of a connection pool.
4. Click the Test Pool button for each instance of the connection pool. Test results are displayed at the top of the pane.
5. Optionally, click the Test pool on all servers button to test all instances of the connection pool. This button is only available if you have more than one instance of the connection pool in your domain.

Adding a Note to an Application-Scoped Connection Pool

1. In the left pane, click to expand Deployments, Applications, and the deployed application name to display a list of application components.
2. Select the application-scoped data source from the list of application components. In the right pane, tabs display with attributes for the application-scoped connection pool associated with the application-scoped data source.
3. Click the Notes tab and enter a note in the text field.

Connection Pool and Data Source Configuration Guidelines

The following sections provide configuration guidelines and examples for JDBC connection pools and data sources.

Drivers Supported for Local Transactions

JDBC 2.0 drivers that support the JDBC Core 2.0 API (`java.sql`), such as the WebLogic `jdbcDriver` for Oracle. The API allows you to create the class objects necessary to establish a connection with a data source, send queries and update statements to the data source, and process the results.

Drivers Supported for Distributed Transactions Using XA

Any JDBC driver that supports JDBC 2.0 distributed transactions standard extension interfaces (`javax.sql.XADataSource`, `javax.sql.XAConnection`, `javax.transaction.xa.XAResource`), such as the WebLogic `jdbcDriver` for Oracle/XA.

Drivers Supported for Distributed Transactions without XA

Any JDBC driver that supports JDBC 2.0 Core API but does not support JDBC 2.0 distributed transactions standard extension interfaces (non-XA). Only one non-XA JDBC driver can participate in a distributed transaction. See [“Configuring Non-XA JDBC Drivers for Distributed Transactions”](#) on page 127-27

Configuring JDBC Drivers for Local Transactions

To configure JDBC drivers for local transactions, set up the JDBC connection pool as follows:

- Specify the `Driver Classname` attribute as the name of the class supporting the `java.sql.Driver` interface.
- Specify the data properties. These properties are passed to the specific `Driver` as driver properties.

For more information on WebLogic JDBC drivers, refer to the BEA documentation for the specific driver you are using: [Using WebLogic jDriver for Oracle at {DOCR00T}/oracle/index.html](#) and [Using WebLogic jDriver for Microsoft SQL Server at {DOCR00T}/mssqlserver4/index.html](#). If you are using a third-party driver, refer to [Using Third-Party JDBC XA Drivers with WebLogic Server in Programming WebLogic JTA at {DOCR00T}/jta/thirdpartytx.html](#) and the documentation from the JDBC driver vendor. The following tables show sample JDBC connection pool and Data Source configurations using the WebLogic jDrivers.

The following table shows a sample connection pool configuration using the WebLogic jDriver for Oracle.

Note: The following configuration examples use a Password attribute. The Password attribute value overrides any password defined in Properties (as a name/value pair). This attribute is passed to the JDBC driver when creating physical database connections. The value is stored in an encrypted form in the `config.xml` file and can be used to avoid storing passwords in clear text in that file.

Table 127-2 WebLogic jDriver for Oracle: Connection Pool Configuration

Attribute Name	Attribute Value
General Tab (JDBC Connection Pool --> Configuration --> General)	
Name	myConnectionPool
URL	jdbc:weblogic:oracle
Driver Classname	weblogic.jdbc.oci.Driver
Properties	user=scott;server=localdb
Password	tiger (This value overrides any password defined in Properties as a name value pair)
Connections Tab (JDBC Connection Pool --> Configuration --> Connections)	
Initial Capacity	1
Max Capacity	15
Capacity Increment	1

Table 127-2 WebLogic jDriver for Oracle: Connection Pool Configuration

Attribute Name	Attribute Value
Shrink Frequency Seconds	900
Test Table Name	dual
Target and Deploy (JDBC Connection Pool --> Target and Deploy)	
Targets	myserver

The following table shows a sample Data Source configuration using the WebLogic jDriver for Oracle.

Table 127-3 Data Source Configuration

Attribute Name	Attribute Value
Configuration Tab (JDBC Data Source --> Configuration)	
Name	myDataSource
JNDI Name	myconnection
Pool Name	myConnectionPool
Row Prefetch Size	48
Stream Chunk Size	256
Target and Deploy Tab (JDBC Data Source --> Target and Deploy)	
Targets	myserver

The following table shows a sample connection pool configuration using the IBM Informix JDBC Driver.

Table 127-4 IBM Informix JDBC Driver: Connection Pool Configuration

Attribute Name	Attribute Value
General Tab (JDBC Connection Pool --> Configuration --> General)	

Table 127-4 IBM Informix JDBC Driver: Connection Pool Configuration

Attribute Name	Attribute Value
Name	myConnectionPool
URL	jdbc:informix-sqli:ifxserver:1543
Driver Classname	com.informix.jdbc.IfxDriver
Properties	informixserver=ifxserver;user=informix
Password	informix (Displayed as *****)
Connections Tab (JDBC Connection Pool --> Configuration --> Connections)	
Initial Capacity	1
Max Capacity	15
Capacity Increment	1
Login Delay Seconds	1
Shrink Frequency Seconds	900
Target and Deploy (JDBC Connection Pool --> Target and Deploy)	
Targets	myserver

Configuring XA JDBC Drivers for Distributed Transactions

To allow XA JDBC drivers to participate in distributed transactions, configure the JDBC data source and connection pool as follows:

- Specify the `Driver Classname` attribute as the name of the class supporting the `javax.sql.XADataSource` interface.
- Make sure that the database properties are specified. These properties are passed to the specified `XADataSource` as data source properties. For more information on data source properties for the WebLogic `JDriver` for Oracle, see “WebLogic

jDriver for Oracle/XA Data Source Properties.” For information about data source properties for third-party drivers, see the vendor documentation.

- In the data source, make sure that Honor Global Transactions is selected (the default).

The following table shows an example of a JDBC connection pool configuration using the WebLogic jDriver for Oracle in XA mode.

Table 127-5 WebLogic jDriver for Oracle/XA: Connection Pool Configuration

Attribute Name	Attribute Value
General Tab (JDBC Connection Pool --> Configuration --> General)	
Name	fundsXferAppPool
URL	(none required)
Driver Classname	weblogic.jdbc.oci.xa.XADataSource
Properties	user=scott;server=localdb
Password	tiger (This value overrides any password defined in Properties as a name value pair)
Connections Tab (JDBC Connection Pool --> Configuration --> Connections)	
Initial Capacity	1
Max Capacity	15
Capacity Increment	1
Shrink Frequency Seconds	900
Test Table Name	dual
Target and Deploy (JDBC Connection Pool --> Target and Deploy)	
Targets	myserver

The following table shows an example of a data source (Tx Data Source in the `config.xml` file) configuration using the WebLogic jDriver for Oracle in XA mode.

Table 127-6 WebLogic jDriver for Oracle/XA: Data Source Configuration

Attribute Name	Attribute Value
Configuration Tab (JDBC Data Source --> Configuration)	
Name	<code>fundsXferDataSource</code>
JNDI Name	<code>myapp.fundsXfer</code>
Pool Name	<code>fundsXferAppPool</code>
Honor Global Transactions	<code>true</code> (Must be selected when you create the data source.)
Target and Deploy Tab (JDBC Data Source --> Target and Deploy)	
Targets	<code>myserver</code>

Note: Do not create two Tx Data Sources that point to the same connection pool. If a transaction uses two different Tx Data Sources which are both pointed to the same connection pool, you will get an `XA_PROTO` error when you try to access the second connection.

You can also configure the JDBC connection pool to use a third-party vendor's driver in XA mode. In such cases, the data source properties are set via reflection on the `XADataSource` instance using the JavaBeans design pattern. In other words, for property `abc`, the `XADataSource` instance must support get and set methods with the names `getAbc` and `setAbc`, respectively.

The following attributes are an example of a JDBC connection pool configuration using the Oracle Thin Driver.

Table 127-7 Oracle Thin Driver: Connection Pool Configuration

Attribute Name	Attribute Value
General Tab (JDBC Connection Pool --> Configuration --> General)	
Name	<code>jtaXAPool</code>
URL	<code>jdbc:oracle:thin:@server:port:sid</code>

Table 127-7 Oracle Thin Driver: Connection Pool Configuration

Attribute Name	Attribute Value
Driver Classname	<code>oracle.jdbc.xa.client.OracleXADataSource</code>
Properties	<code>user=scott</code>
Password	<code>tiger</code> (This value overrides any password defined in Properties as a name value pair)
Connections Tab (JDBC Connection Pool --> Configuration --> Connections)	
Initial Capacity	1
Max Capacity	15
Capacity Increment	1
Shrink Frequency Seconds	900
Test Table Name	dual
Target and Deploy (JDBC Connection Pool --> Target and Deploy)	
Targets	myserver

The following table shows an example of a Data Source configuration using the Oracle Thin Driver for XA.

Table 127-8 Oracle Thin Driver: Data Source Configuration for XA

Attribute Name	Attribute Value
Configuration Tab (JDBC Data Source --> Configuration)	
Name	<code>jtaXADS</code>
JNDI Name	<code>jtaXADS</code>
Pool Name	<code>jtaXAPool</code>
Honor Global Transactions	<code>true</code> (Must be selected when you create the data source.)
Target and Deploy Tab (JDBC Data Source --> Target and Deploy)	

Table 127-8 Oracle Thin Driver: Data Source Configuration for XA

Attribute Name	Attribute Value
Targets	myserver

The following table shows an example of a JDBC connection pool configuration for distributed transactions using the PointBase JDBC driver.

Table 127-9 PointBase: Connection Pool Configuration

Attribute Name	Attribute Value
General Tab (JDBC Connection Pool --> Configuration --> General)	
Name	demoXAPool
URL	jdbc:pointbase:server://localhost/demo
Driver Classname	com.pointbase.xa.xaDataSource
Properties	user=public DatabaseName=jdbc:pointbase:server://localhost/demo
Password	public (Displayed as *****)
Connections Tab (JDBC Connection Pool --> Configuration --> Connections)	
Initial Capacity	1
Max Capacity	15
Capacity Increment	1
Supports Local Transaction	true
Shrink Frequency Seconds	900
Test Table Name	users
Target and Deploy (JDBC Connection Pool --> Target and Deploy)	
Targets	myserver

Configure the Data Source for use with a PointBase XA driver as follows.

Table 127-10 PointBase: Data Source Configuration for XA

Attribute Name	Attribute Value
Configuration Tab (JDBC Data Source --> Configuration)	
Name	jtaXADS
JNDI Name	JTAXADS
Pool Name	demoXAPool
Honor Global Transactions	true (Must be selected when you create the data source.)
Target and Deploy Tab (JDBC Data Source --> Target and Deploy)	
Targets	myserver

WebLogic jDriver for Oracle/XA Data Source Properties

Table 127-11 lists the data source properties supported by the WebLogic jDriver for Oracle. The JDBC 2.0 column indicates whether a specific data source property is a JDBC 2.0 standard data source property (S) or a WebLogic Server extension to JDBC (E).

The Optional column indicates whether a particular data source property is optional or not. Properties marked with Y* are mapped to the corresponding fields of the Oracle `xa_open` string (value of the `openString` property) as listed in Table 127-11. If they are not specified, their default values are taken from the `openString` property. If they are specified, their values should match those specified in the `openString` property. If the properties do not match, a `SQLException` is thrown when you attempt to make an XA connection.

Mandatory properties marked with N* are also mapped to the corresponding fields of the Oracle `xa_open` string. Specify these properties when specifying the Oracle `xa_open` string. If they are not specified or if they are specified but do not match, an `SQLException` is thrown when you attempt to make an XA connection.

Property Names marked with ** are supported but not used by WebLogic Server.

Table 127-11 Data Source Properties for WebLogic jDriver for Oracle/XA

Property Name	Type	Description	JDBC 2.0 standard/extension	Optional	Default Value
databaseName**	String	Name of a particular database on a server.	S	Y	None
dataSourceName	String	A data source name; used to name an underlying XADataSource.	S	Y	Connection Pool Name
description	String	Description of this data source.	S	Y	None
networkProtocol**	String	Network protocol used to communicate with the server.	S	Y	None
password	String	A database password.	S	N*	None
portNumber**	Int	Port number at which a server is listening for requests.	S	Y	None
roleName**	String	The initial SQL role name.	S	Y	None
serverName	String	Database server name.	S	Y*	None
user	String	User's account name.	S	N*	None
openString	String	Oracle's XA open string.	E	Y	None
oracleXATrace	String	Indicates whether XA tracing output is enabled. If enabled (true), a file with a name in the form of <code>xa_poolnamedate.trc</code> is placed in the directory in which the server is started.	E	Y	true

Table 127-12 lists the mapping between Oracle’s `xa_open` string fields and data source properties.

Table 127-12 Mapping of `xa_open` String Names to JDBC Data Source Properties

Oracle <code>xa_open</code> String Field Name	JDBC 2.0 Data Source Property	Optional
<code>acc</code>	<code>user, password</code>	N
<code>sqlnet</code>	<code>ServerName</code>	

Note: You must specify `Threads=true` in Oracle’s `xa_open` string.

For a complete description of Oracle’s `xa_open` string fields, see your Oracle documentation.

Additional XA Connection Pool Properties

When using connections from a connection pool in distributed transactions, you may need to set additional properties for the connection pool so that the connection pool handles the connection properly within WebLogic Server in the context of the transaction. You set these properties in the configuration file (`config.xml`) within the `JDBCConnectionPool` tag. By default, all additional properties are set to false. You set the properties to true to enable them.

In many cases, WebLogic Server automatically sets the proper value for these properties internally so that you do not have to set them manually.

For additional XA connection pool properties, see “Advanced Attributes” on page 113-5.

KeepXAConnTillTxComplete

Some DBMSs require that you start and end a transaction in the same physical database connection. In some cases, a transaction in WebLogic Server may start in one physical database connection and end in another physical database connection. To force a connection pool to reserve a physical connection and provide the *same*

connection to an application throughout transaction processing until the transaction is complete, you select the Keep XA Connection Till Transaction Complete option on the [JDBC Connection Pool --> Configuration --> Connections](#) tab.

Note: This property is *required* to support distributed transactions with DB2 and Sybase.

Configuring Non-XA JDBC Drivers for Distributed Transactions

When configuring the JDBC connection pool to allow non-XA JDBC drivers to participate with other resources in distributed transactions, select the Emulate Two-Phase Commit for non-XA Driver attribute (`EnableTwoPhaseCommit` in the `JDBCTxDataSource` MBean) for the JDBC Tx Data Source. This parameter is ignored by resources that support the `XAResource` interface. Note that only one non-XA connection pool may participate in a distributed transaction. See “Emulating Two-Phase Commit” on page 141-3 for more information.

Note: There are risks to data integrity when using the Emulate Two-Phase Commit for non-XA Driver option. BEA recommends that you use an XA-compliant JDBC driver rather than use this option. Make sure you consider the risks below before enabling this option. See “Limitations and Risks When Using a Non-XA Driver in Global Transactions” on page 141-4.

Non-XA Driver/Single Resource

If you are using only one non-XA driver and it is the only resource in the transaction, leave the Emulate Two-Phase Commit for non-XA Driver option unselected in the Console (accept the default `EnableTwoPhaseCommit = false`). In this case, the Transaction Manager performs a one-phase optimization.

Non-XA Driver/Multiple Resources

If you are using one non-XA JDBC driver with other XA resources, select Emulate Two-Phase Commit in the Administration Console (`EnableTwoPhaseCommit = true`).

When the Emulate Two-Phase Commit for non-XA Driver option is selected (`EnableTwoPhaseCommit` is set to `true`), the non-XA JDBC resource always returns `XA_OK` during the `XAResource.prepare()` method call. The resource attempts to commit or roll back its local transaction in response to subsequent `XAResource.commit()` or `XAResource.rollback()` calls. If the resource commit or rollback fails, a heuristic error results. Application data may be left in an inconsistent state as a result of a heuristic failure.

When the Emulate Two-Phase Commit for non-XA Driver option is not selected in the Console (`EnableTwoPhaseCommit` is set to `false`), the non-XA JDBC resource causes `XAResource.prepare()` to fail. This mechanism ensures that there is only one participant in the transaction, as `commit()` throws a `SystemException` in this case. When there is only one resource participating in a transaction, the one phase optimization bypasses `XAResource.prepare()`, and the transaction commits successfully in most instances.

The following table shows configuration attributes for a sample JDBC connection pool using a non-XA JDBC driver.

Table 127-13 WebLogic jDriver for Oracle: Connection Pool Configuration

Attribute Name	Attribute Value
General Tab (JDBC Connection Pool --> Configuration --> General)	
Name	<code>fundsXferAppPool</code>
URL	<code>jdbc:weblogic:oracle</code>
Driver Classname	<code>weblogic.jdbc.oci.Driver</code>
Properties	<code>user=scott;server=localdb</code>
Password	<code>tiger</code> (Displayed as <code>*****</code> when typed, hidden thereafter; this value overrides any password defined in Properties as a name value pair)
Connections Tab (JDBC Connection Pool --> Configuration --> Connections)	
Initial Capacity	0
Max Capacity	5
Capacity Increment	1

Table 127-13 WebLogic jDriver for Oracle: Connection Pool Configuration

Attribute Name	Attribute Value
Shrink Frequency Seconds	900
Test Table Name	dual
Target and Deploy (JDBC Connection Pool --> Target and Deploy)	
Targets	myserver

The following table shows configuration attributes for a sample Data Source using a non-XA JDBC driver.

Table 127-14 WebLogic j Driver for Oracle: Data Source Configuration

Attribute Name	Attribute Value
Configuration Tab (JDBC Data Source --> Configuration)	
Name	fundsXferDataSource
JNDI Name	myapp.fundsXfer
Pool Name	fundsXferAppPool
Honor Global Transactions	true (Must be selected when you create the data source.)
Emulate Two-Phase Commit for non-XA Driver	selected (EnableTwoPhaseCommit = true)
Target and Deploy Tab (JDBC Data Source --> Target and Deploy)	
Targets	myserver

Security for JDBC Connection Pools

You can optionally restrict access to JDBC connection pools. In WebLogic Server, security policies answer the question “who has access” to a WebLogic resource. A security policy is created when you define an association between a WebLogic resource and a user, group, or role. A WebLogic resource has no protection until you assign it a security policy. For instructions on how to set up security for all WebLogic Server resources, see “Protecting WebLogic Resources” on page 428-17. For more information about securing server resources, see [Securing WebLogic Resources](#).

Managing JDBC Connection Pools

From the JDBC Connection Pool property tabs in the Administration Console, you can manage the connections pools in your domain. The following sections provide detailed instructions for manually performing management tasks on JDBC connection pools.

Testing a JDBC Connection Pool

On the JDBC Connection Pool—Testing tab, you can manually test a JDBC connection in a connection pool on each server on which the connection pool is deployed.

When you test a connection pool, WebLogic Server reserves and releases a connection from the connection pool.

To make the test more meaningful, make sure that Test Reserved Connections or Test Released Connections is selected on the Configuration—Connections tab (under Advanced Options). If either of these options is selected, WebLogic Server not only reserves and releases a connection, but also tests the physical database connection. See Test Reserved Connections in [Attributes](#).

To see a description of the information displayed on the JDBC connection Pool—Testing tab, see Attributes.

Also see “Connection Testing Options” on page 127-40 for more information about connection testing options and the default value for Test Table Name.

To test a connection in a connection pool, follow these steps:

1. In the left pane, click to expand the Services, JDBC, and Connection Pool nodes to display the list of connection pools in the current domain.
2. Click the connection pool that you want to deploy. A dialog displays in the right pane showing the tabs associated with this instance.
3. Click the Testing tab. The Testing tab displays a list of instances of the selected connection pool. Each server on which the connection pool is deployed is listed. Each server can have only one instance of a connection pool.
4. Click the Test Pool button for each instance of the connection pool. Test results are displayed at the top of the pane.
5. Optionally, click the Test pool on all servers button to test all instances of the connection pool. This button is only available if you have more than one instance of the connection pool in your domain.

Resetting All Connections in a JDBC Connection Pool

When you reset a connection pool, WebLogic Server shuts down and recreates all database connections in the connection pool.

1. In the left pane, click to expand the Services, JDBC, and Connection Pools nodes to display the list of connection pools in the current domain.
2. Click the connection pool that you want to reset. A dialog displays in the right pane showing the tabs associated with this connection pool.
3. Click the Control tab. The control tab lists each server on which the connection pool is deployed.
4. Click Reset for each server on which you want to reset all connections in the connection pool.

Shrinking a JDBC Connection Pool

If you configure a connection pool so that it can add database connections as demand for connections increases, you can click the Shrink button on the Control tab to manually shrink the connection pool. When you shrink a connection pool, WebLogic Server reduces the number of connections in the pool to the greater of either the initial capacity or the number of connections currently in use.

1. In the left pane, click to expand the Services, JDBC, and Connection Pools nodes to display the list of connection pools in the current domain.
2. Click the connection pool that you want to reset. A dialog displays in the right pane showing the tabs associated with the connection pool.
3. Click the Control tab. The control tab lists each server on which the connection pool is deployed.
4. Click Shrink for each server on which you want to shrink the connection pool instance.

Suspending a JDBC Connection Pool

When you suspend a connection pool, you make the connections in the pool unavailable for applications to use. WebLogic Server provides the following options for suspending a connection pool:

- Suspend—which marks the pool as disabled and blocks any new requests for a connection from the connection pool. An application will get an exception if it requests a connection or tries to use a connection it reserved before the connection pool was suspended.
- Force Suspend—which marks the pool as disabled, blocks any new requests for a connection from the connection pool, and closes and recreates connections currently in use.

Connections in a suspended connection pool remain intact. The connections are not recreated when you resume the connection pool, except when the connection pool was Force Suspended.

To suspend a connection pool, follow these steps:

1. In the left pane, click to expand the Services, JDBC, and Connection Pool nodes to display the list of connection pools in the current domain.
2. Click the connection pool that you want to suspend. A dialog displays in the right pane showing the tabs associated with the connection pool.
3. Click the Control tab. The control tab lists each server on which the connection pool is deployed.
4. For each server listed, choose one of the following options:
 - Click Suspend to block new requests to reserve a connection from the connection pool and mark the connection pool as disabled. If connections are currently in use, this operation will fail.
 - Click Force Suspend to block new requests to reserve a connection from the connection pool and to stop all current use of connections from the connection pool. This operation also marks the connection pool as disabled.

Resuming a JDBC Connection Pool

After manually suspending a connection pool, you can re-enable it by clicking Resume on the JDBC Connection Pool—Control tab. You cannot use the Resume functionality to restart a connection pool that failed to start properly.

Follow these instructions.

1. In the left pane, click to expand the Services, JDBC, and Connection Pool nodes to display the list of connection pools in the current domain.
2. Click the connection pool that you want to resume. A dialog displays in the right pane showing the tabs associated with the connection pool.
3. Click the Control tab. The control tab lists each server on which the connection pool is deployed.
4. Click the Resume button for the instance of the connection pool that you want to re-enable. This option is only available for connection pools that were successfully suspended.

Shutting Down a JDBC Connection Pool

To shut down an instance of a connection pool, you can un-deploy the connection pool on the server. This operation closes all physical database connections in the connection pool. To shut down the connection pool on more than one target, you must un-deploy on each deployment target.

Note: If connections are currently in use, the shutdown operation will fail and the connection pool will go into the Suspended state. You must resume the connection pool to restore normal operations.

If you want to force the connection pool to shut down, force suspend the connection pool and then follow the steps below.

To shut down a connection pool, follow these steps:

1. In the left pane, click to expand the Services, JDBC, and Connection Pools nodes to display the list of connection pools in the current domain.
2. Click the connection pool that you want to shut down. A dialog displays in the right pane showing the tabs associated with this instance.
3. Click the Target and Deploy tab.
4. Clear the check box for the servers or clusters on which you want to shut down the connection pool. Click Apply to save your changes.

See the following related information:

- “Resetting All Connections in a JDBC Connection Pool” on page 127-31
- “Suspending a JDBC Connection Pool” on page 127-32

Restarting a JDBC Connection Pool

To restart a connection pool after shutting it down by undeploying it (see “Shutting Down a JDBC Connection Pool” on page 127-34), you re-deploy the connection pool to servers and clusters. See “Deploying a JDBC Connection Pool to One or More Servers or Clusters” on page 127-7 for instructions.

Destroying or Deleting a JDBC Connection Pool

When you destroy a JDBC connection pool, all database connections *in all instances* of the connection pool are closed and the connection pool configuration is removed from the domain.

Note: When you destroy a connection pool, you destroy all instances of the connection pool, not just the instance for which you clicked the Destroy button.

There are two Destroy options for connection pools in WebLogic Server:

- **Destroy**—Closes all database connections in all instances of the connection pool and permanently deletes the connection pool configuration from the domain. If connections in the connection pool are in use, the operation will fail.
- **Force Destroy**—Forcibly closes all database connections in all instances of the connection pool, even if connections are in use, and permanently deletes the connection pool configuration from the domain.

To destroy a connection pool, follow these steps:

1. In the left pane, click to expand the Services, JDBC, and Connection Pools nodes to display the list of connection pools in the current domain.
2. Click the connection pool that you want to destroy. A dialog displays in the right pane showing the tabs associated with the connection pool.
3. Click the Control tab. The control tab lists each server on which the connection pool is deployed.
4. For any server listed, choose one of the following options:
 - Click **Destroy** to close all database connections and delete the connection pool. This action applies to all servers. If a connection is in use, the operation will fail.
 - Click **Force Destroy** to forcibly close all database connections and delete the connection pool. This action applies to all servers.

Clearing the Statement Cache for a JDBC Connection Pool

To clear the statement cache for all connections in a connection pool, follow these steps:

1. In the left pane, click to expand the Services, JDBC, and Connection Pools nodes to display the list of connection pools in the current domain.
2. Click the connection pool for which you want to clear the statement cache. A dialog displays in the right pane showing the tabs associated with this connection pool.
3. Click the Control tab. The control tab lists each server on which the connection pool is deployed.
4. Click Clear Statement Cache for each server on which you want to clear the statement cache for all connections in the connection pool. Repeat for each instance of the connection pool as required.

For more information about the statement cache for a connection pool, see “Increasing Performance with the Statement Cache” on page 127-43.

Monitoring Connections in a JDBC Connection Pool

1. In the left pane, click to the JDBC node to expand it.
2. Click the Connection Pools node to expand it and show the list of connection pools defined in your domain.
3. Click the connection pool for which you want to see database connection information. A dialog displays in the right pane showing tabs with attributes for the connection pool.
4. Click the Monitoring tab. A table displays with information about connections in the selected JDBC connection pool for each server on which the connection pool is deployed.

For details about the information displayed, see “JDBC Connection Pool --> Monitoring” on page 117-1.

Tuning Connection Pools

By properly configuring connection pools in your WebLogic Server domain, you can improve application and system performance.

Enabling Connection Requests to Wait for a Connection

On the JDBC—Connection Pool—Configuration—Connections tab, there are two attributes that you can set to enable connection requests to wait for a connection from a connection pool: Connection Reserve Timeout and Maximum Waiting for Connection.

Connection Reserve Timeout

When an application requests a connection from a connection pool, if all connections in the connection pool are in use and if the connection pool has expanded to its maximum capacity, you can configure a Connection Reserve Timeout value (in seconds) so that connection requests will wait for a connection to become available. After the Connection Reserve Timeout has expired, if no connection has become available, the request will fail.

If you set Connection Reserve Timeout to -1, a connection request will wait indefinitely.

See “Connection Reserve Timeout” on page 113-10 for more attribute details.

Maximum Waiting for Connection

Note that connection requests that wait for a connection block a thread. If too many connection requests concurrently wait for a connection and block threads, your system performance can degrade. To avoid this, you can set the Maximum Waiting for Connection attribute, which limits the number connection requests that can concurrently wait for a connection.

If you set Maximum Waiting for Connection to 0, the feature is disabled and connection requests will not be able to wait for a connection.

See “Maximum Waiting for Connection” on page 113-12 for more attribute details.

To Enable a Connection Request to Wait for a Connection

1. In the left pane, click to expand the Services, JDBC, and Connection Pool nodes to display the list of connection pools in the current domain.
2. Click the connection pool that you want to configure. A dialog displays in the right pane showing the tabs associated with this instance.
3. Click the Configuration tab, then click the Connections tab.
4. Click Show to show the advanced connection options.
5. In Connection Reserve Timeout, enter the number of seconds that connection requests can wait for a connection.
6. In Maximum Waiting for a Connection, enter the maximum number of connection requests that can wait for a connection from the connection pool while blocking threads.
7. Click Apply.

Automatically Recovering Leaked Connections

A leaked connection is a connection that was not properly returned to the connection pool. To automatically recover leaked connections, you can specify a value for Inactive Connection Timeout on the JDBC—~~Connection~~ Pool—~~Configuration~~—~~Connections~~ tab. When you set a value for Inactive

Connection Timeout, WebLogic Server will forcibly return a connection to the connection pool when there is no activity on a reserved connection for the number of seconds that you specify. When set to 0 (the default value), this feature is turned off.

See “Inactive Connection Timeout” on page 113-12 for more attribute details.

Note that the actual timeout could exceed the configured value for Inactive Connection Timeout. The internal connection pool maintenance thread runs every 5 seconds. When it reaches the Inactive Connection Timeout (for example 30 seconds), it checks for inactive connections. To avoid timing out a connection that was reserved just before the current check or just after the previous check, the server gives an inactive connection a “second chance.” On the next check, if the connection is still inactive, the server times it out and forcibly returns it to the connection pool. On average, there could be a lag of 50% more than the configured value.

Enabling Automatic Leaked Connection Recovery

1. In the left pane, click to expand the Services, JDBC, and Connection Pool nodes to display the list of connection pools in the current domain.
2. Click the connection pool that you want to configure. A dialog displays in the right pane showing the tabs associated with this instance.
3. Click the Configuration tab, then click the Connections tab.
4. Click Show to show the advanced connection options.
5. In Inactive Connection Timeout, enter the number of seconds of inactivity after which a connection will forcibly be returned to the connection pool.

Viewing Leaked Connections

After enabling automatic leaked connection recovery, you can view statistics about connections that leaked from a connection pool:

1. In the left pane, click to expand the Services, JDBC, and Connection Pool nodes to display the list of connection pools in the current domain.
2. Right-click the connection pool that you suspect is leaking connections and select View Leaked Connections. If any connections leaked and were recovered, information about the application that reserved the connection is displayed in the right pane.

Initializing Database Connections with SQL Code

When WebLogic Server creates database connections in a connection pool, the server can automatically run SQL code to initialize the database connection. To enable this feature, enter `SQL` followed by a space and the SQL code you want to run in the `Init SQL` attribute on the `JDBC—Connection Pool—Configuration—Connections` tab. If you leave this attribute blank (the default), WebLogic Server does not run any code to initialize database connections.

WebLogic Server runs this code whenever it creates a database connection for the connection pool, which includes at server startup, when expanding the connection pool, and when refreshing a connection.

To initialize a database connection with SQL code:

1. In the left pane, click to expand the `Services`, `JDBC`, and `Connection Pool` nodes to display the list of connection pools in the current domain.
2. Click the connection pool that you want to configure. A dialog displays in the right pane showing the tabs associated with this instance.
3. Click the `Configuration` tab, then click the `Connections` tab.
4. Click `Show` to show the advanced connection options.
5. In `Init SQL`, enter `SQL` followed by a space and the SQL code you want to run to initialize database connections.

Connection Testing Options

To make sure that the database connections in a connection pool remain healthy, you should periodically test the connections. WebLogic Server includes two basic types of testing: automatic testing that you configure with options on the `JDBC—Connection Pool—Configuration—Connections` tab and manual testing that you can do to trouble-shoot a connection pool from the `JDBC—Connection Pool—Testing` tab. The following section discusses automatic connection testing options. For more information about manual connection testing, see “Testing a JDBC Connection Pool” on page 127-30.

On the JDBC—Connection Pool—Configuration—Connections, you use the following settings to configure connection testing:

- **Test Frequency**—Use this attribute to specify the number of seconds between tests of unused connections. The server tests unused connection and reopens any faulty connections. You must also set the Maximum Connections Made Unavailable and a Test Table Name.
- **Test Reserved Connections**—Select this option to test each connection before giving to a client. This may add a slight delay to the request, but it guarantees that the connection is healthy. You must also set a Test Table Name.
- **Test Created Connections**—Select this option to test each database connection after it is created. This applies to connections created at server startup and when the connection pool is expanded. You must also set a Test Table Name.
- **Test Released Connections**—Select this option to test connections when they are returned to the connection pool. You must also set a Test Table Name.
- **Maximum Connections Made Unavailable**—Use this option to limit the number idle connections that the server will test. For example, if you have 10 connections in your connection pool and five are in use, if the server were to begin testing all five connections that are not in use, there would be no connections available to fill a connection request. If you set the Maximum Connections Made Unavailable attribute to 3, there would still be two connections available to fill a connection request.
- **Test Table Name**—Use this attribute to specify a table name to use in a connection test. You can also specify SQL code to run in place of the standard test by entering SQL followed by a space and the SQL code you want to run as a test. Test Table Name is required to enable any database connection testing.

You should set connection testing attributes so that they best fit your environment. For more details about these attributes, see “Attributes” on page 113-2.

To enable a database connection testing:

1. In the left pane, click to expand the Services, JDBC, and Connection Pool nodes to display the list of connection pools in the current domain.
2. Click the connection pool that you want to configure. A dialog displays in the right pane showing the tabs associated with this instance.
3. Click the Configuration tab, then click the Connections tab.

4. Click Show to show the advanced connection options.
5. Select or specify a value for at least one of the following attributes:

- Test Frequency
- Test Reserved Connections
- Test Created Connections
- Test Released Connections

If you specify a value for Test Frequency, you must also specify a value for Maximum Connections Made Unavailable.

6. Specify a value for Test Table Name: either a table that exists in your database or SQL followed by a space and SQL code.
7. Click Apply.

Default Test Table Name

When you create a connection pool, the JDBC Connection Pool Assistant automatically sets the Test Table Name attribute for a connection pool based on the DBMS of the JDBC driver that you select. The Test Table Name attribute is used in connection testing which is optionally performed periodically or when you create, reserve, or release a connection, depending on how you configure the connection pool. For database tests to succeed, the database user used to create database connections in the connection pool must have access to the database table. If not, you should either grant access to the user (make this change in the DBMS) or change the Test Table Name attribute to the name of a table to which the user does have access (make this change in the WebLogic Server Administration Console).

Table 127-1 Default Test Table Name by DBMS

DBMS	Default Test Table Name
Cloudscape	SELECT 1
DB2	SELECT COUNT(*) FROM SYSIBM.SYSTABLES
Informix	SELECT COUNT(*) FROM SYSTABLES
Microsoft SQL Server	SELECT COUNT(*) FROM SYSOBJECTS

Table 127-1 Default Test Table Name by DBMS

DBMS	Default Test Table Name
MySQL	SELECT 1
Oracle	SELECT 1 FROM DUAL
PointBase	SELECT COUNT(*) FROM SYSTABLES
PostgreSQL	SELECT 1
Progress	SELECT COUNT(*) FROM SYSTABLES
Sybase	SELECT COUNT(*) FROM SYSOBJECTS

Increasing Performance with the Statement Cache

When you use a prepared statement or callable statement in an application or EJB, there is considerable processing overhead for the communication between the application server and the database server and on the database server itself. To minimize the processing costs, WebLogic Server can cache prepared and callable statements used in your applications. When an application or EJB calls any of the statements stored in the cache, WebLogic Server reuses the statement stored in the cache. Reusing prepared and callable statements reduces CPU usage on the database server, improving performance for the current statement and leaving CPU cycles for other tasks.

Each connection in a connection pool has its own individual cache of prepared and callable statements used on the connection. However, you configure statement cache options per connection pool. That is, the statement cache for each connection in a connection pool uses the statement cache options specified for the connection pool. Statement cache configuration options include:

- **Statement Cache Type**—The algorithm that determines which statements to store in the statement cache. See “Statement Cache Algorithms” on page 127-44.
- **Statement Cache Size**—The number of statements to store in the cache for each connection. The default value is 10. See “Statement Cache Size” on page 127-45.

You can use the following methods to set statement cache options for a connection pool:

- Using the Administration Console (preferred). See “Configuring the Statement Cache for a JDBC Connection Pool” on page 127-7.
- Using the WebLogic management API. See the following methods in the [Javadocs for WebLogic Classes at {DOCR00T}/javadocs/weblogic/management/configuration/JDBCConnectionPoolMBean.html](#):
 - `getStatementCacheType()`
 - `setStatementCacheType(string type)`
 - `getStatementCacheSize()`
 - `setStatementCacheSize(int cacheSize)`

You can also manually clear the statement cache for a connection pool. See “Clearing the Statement Cache for a JDBC Connection Pool” on page 127-36.

Statement Cache Algorithms

The Statement Cache Type (or algorithm) determines which prepared and callable statements to store in the cache for each connection in a connection pool. You can choose from the following options:

- LRU (Least Recently Used)
- Fixed

LRU (Least Recently Used)

When you select LRU (Least Recently Used, the default) as the Statement Cache Type, WebLogic Server caches prepared and callable statements used on the connection until the statement cache size is reached. When an application calls `Connection.prepareStatement()`, WebLogic Server checks to see if the statement is stored in the statement cache. If so, WebLogic Server returns the cached statement (if it is not already being used). If the statement is not in the cache, and the cache is full (number of statements in the cache = statement cache size), WebLogic Server determines which existing statement in the cache was the least recently used and replaces that statement in the cache with the new statement.

The LRU statement cache algorithm in WebLogic Server uses an approximate LRU scheme.

Fixed

When you select **FIXED** as the Statement Cache Type, WebLogic Server caches prepared and callable statements used on the connection until the statement cache size is reached. When additional statements are used, they are not cached.

With this statement cache algorithm, you can inadvertently cache statements that are rarely used. In many cases, the LRU algorithm is preferred because rarely used statements will eventually be replaced in the cache with frequently used statements.

Statement Cache Size

The Statement Cache Size attribute determines the total number of prepared and callable statements to cache for each connection in each instance of the connection pool. By caching statements, you can increase your system performance. However, you must consider how your DBMS handles open prepared and callable statements. In many cases, the DBMS will maintain a cursor for each open statement. This applies to prepared and callable statements in the statement cache. If you cache too many statements, you may exceed the limit of open cursors on your database server.

For example, if you have a connection pool with 10 connections deployed on 2 servers, if you set the Statement Cache Size to 10 (the default), you may open 200 (10 x 2 x 10) cursors on your database server for the cached statements.

Usage Restrictions for the Statement Cache

Using the statement cache can dramatically increase performance, but you must consider its limitations before you decide to use it. Please note the following restrictions when using the statement cache.

There may be other issues related to caching statements that are not listed here. If you see errors in your system related to prepared or callable statements, you should set the statement cache size to 0, which turns off statement caching, to test if the problem is caused by caching prepared statements.

Calling a Stored Statement After a Database Change May Cause Errors

Prepared statements stored in the cache refer to specific database objects at the time the prepared statement is cached. If you perform any DDL (data definition language) operations on database objects referenced in prepared statements stored in the cache, the statements may fail the next time you run them. For example, if you cache a statement such as `select * from emp` and then drop and recreate the `emp` table, the next time you run the cached statement, the statement may fail because the exact `emp` table that existed when the statement was prepared, no longer exists.

Likewise, prepared statements are bound to the data type for each column in a table in the database at the time the prepared statement is cached. If you add, delete, or rearrange columns in a table, prepared statements stored in the cache are likely to fail when run again.

These limitations depend on the behavior of your DBMS.

Using `setNull` In a Prepared Statement

When using the WebLogic `jdbcDriver` for Oracle to connect to the database, if you cache a prepared statement that uses a `setNull` bind variable, you must set the variable to the proper data type. If you use a generic data type, as in the following example, data may be truncated or the statement may fail when it runs with a value other than null.

```
java.sql.Types.Long sal=null

.
.
.

if (sal == null)
    setNull(2,int)//This is incorrect
else
    setLong(2,sal)
```

Instead, use the following:

```
if (sal == null)
    setNull(2,long)//This is correct
else
    setLong(2,sal)
```

This issue occurs consistently when using the WebLogic `jdbcDriver` for Oracle. The WebLogic `jdbcDriver` for Oracle converts data types according to Table B-5 in the JDBC specification. This issue may also occur when using other JDBC drivers.

Statements in the Cache May Reserve Database Cursors

When WebLogic Server caches a prepared or callable statement, the statement may open a cursor in the database. If you cache too many statements, you may exceed the limit of open cursors for a connection. To avoid exceeding the limit of open cursors for a connection, you can change the limit in your database management system or you can reduce the statement cache size for the connection pool.

Attributes and Console Screen Reference for JDBC Connection Pools

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“Active JDBC Connections” on page 126-1](#)

[“JDBC Connection Pools” on page 120-1](#)

[“JDBC Connection Pool --> Configuration --> General” on page 114-1](#)

[“JDBC Connection Pool --> Configuration --> Connections” on page 113-1](#)

[“JDBC Connection Pool --> Target and Deploy” on page 119-1](#)

[“JDBC Connection Pool --> Monitoring” on page 117-1](#)

[“JDBC Connection Pool --> Control” on page 115-1](#)

[“JDBC Connection Pool --> Testing” on page 116-1](#)

[“JDBC Connection Pool --> Notes” on page 118-1](#)

[“Configure a JDBC Connection Pool --> Choose database” on page 122-1](#)

[“Configure a JDBC Connection Pool --> Define connection properties” on page 125-1](#)

[“Configure a JDBC Connection Pool --> Define and test connection” on page 123-1](#)

[“Configure a JDBC Connection Pool --> Test Database Connection” on page 121-1](#)

[“JDBC Connection Pool Assistant --> Create and deploy” on page 124-1](#)

[“JDBC Connection Leak Profile” on page 148-1](#)



JDBC Data Source --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

On the JDBC Data Source—Configuration tab, you specify attributes for the selected Data Source. Applications get a database connection from a connection pool by looking up a data source on the Java Naming and Directory Interface (JNDI) tree and then request a connection. The connection pool associated with the datasource provides the connection to the application.

Tasks

[“Creating and Configuring a JDBC Data Source” on page 141-5](#)

[“Cloning a JDBC Data Source” on page 141-6](#)

[“Deploying a JDBC Data Source to a Server or Cluster” on page 141-7](#)

Related Topics

[“Configuring JDBC DataSources” on page 141-1](#)

[“Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1](#)

Attributes

Table 129-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> weblogic.management.configuration.JDBCDataSourceMBean</p> <p><i>Attribute:</i> Name</p>	
JNDI Name	<p>The JNDI path to where this DataSource is bound.</p> <p>Applications that look up the JNDI path will get a <code>javax.sql.DataSource</code> instance that corresponds to this DataSource.</p> <p><i>MBean:</i> weblogic.management.configuration.JDBCDataSourceMBean</p> <p><i>Attribute:</i> JNDIName</p>	
Pool Name	<p>The name of the JDBCConnectionPool to use for this DataSource.</p> <p><i>MBean:</i> weblogic.management.configuration.JDBCDataSourceMBean</p> <p><i>Attribute:</i> JDBCPoolName</p>	

Advanced Attributes

Table 129-2

Attribute Label	Description	Value Constraints
Row Prefetch Enabled	<p>Controls row prefetching between a client and WebLogic Server for each ResultSet. When an external client accesses a database using JDBC through WebLogic Server, row prefetching improves performance by fetching multiple rows from the server to the client in one server access. WebLogic Server will ignore this setting and not use row prefetching when the client and WebLogic Server are in the same JVM.</p> <p><i>MBean:</i> weblogic.management.configuration.JDBCDataSourceMBean</p> <p><i>Attribute:</i> RowPrefetchEnabled</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false
Row Prefetch Size	<p>The number of result set rows to prefetch for a client. The optimal value depends on the particulars of the query. In general, 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.</p> <p><i>MBean:</i> weblogic.management.configuration.JDBCDataSourceMBean</p> <p><i>Attribute:</i> RowPrefetchSize</p>	<p><i>Minimum:</i> 2</p> <p><i>Maximum:</i> 65536</p> <p><i>Default:</i> 48</p>

Table 129-2

Attribute Label	Description	Value Constraints
Stream Chunk Size	Data chunk size for steaming data types. Streaming data types (for example resulting from a call to <code>getBinaryStream()</code>) will be pulled in <code>StreamChunkSize</code> sized chunks from WebLogic Server to the client as needed. <i>MBean:</i> <code>weblogic.management.configuration.JDBCDataSourceMBean</code> <i>Attribute:</i> <code>StreamChunkSize</code>	<i>Units:</i> bytes <i>Minimum:</i> 1 <i>Maximum:</i> 65536 <i>Default:</i> 256
Honor Global Transactions	Specifies whether the data source can participate in a global transaction. When selected, the data source is created as a <code>TxDataSource</code> in the <code>config.xml</code> file. You select this option (the default) when creating the data source. You cannot change this setting after the data source is created.	<i>Default:</i> true

JDBC Data Source --> Notes

[Tasks](#) [Related Topics](#)

Overview

On the JDBC Data Source—Notes tab, you can add a note to the data source configuration. You can use this to save information about the data source that is not apparent in the configuration, such as a history of configuration changes and reasons.

Tasks

[“Adding a Note to a Data Source” on page 141-7](#)

[“Creating and Configuring a JDBC Data Source” on page 141-5](#)

Related Topics

[“Configuring JDBC DataSources” on page 141-1](#)

[“Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1](#)



JDBC Data Source --> Target and Deploy

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

On the JDBC Data Source—Target and Deploy tab, you deploy the selected data source on servers and clusters in the domain.

You should deploy the data source to the same target servers and clusters on which the underlying connection pool is deployed.

Tasks

“Deploying a JDBC Data Source to a Server or Cluster” on page 141-7

“Creating and Configuring a JDBC Data Source” on page 141-5

Related Topics

“Configuring JDBC DataSources” on page 141-1

“Connection Pool and Data Source Configuration Guidelines” on page 127-15

“Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1

Attributes

You should deploy the data source to the same target servers and clusters on which the underlying connection pool is deployed.

Independent Servers—Lists the servers in your configuration on which you can deploy the data source. Select servers and click Apply.

Clusters—Lists clusters and comprising servers on which you can deploy the data source. Select the cluster on which you want to deploy the data source and click Apply. In almost all cases, you should deploy to an entire cluster, not individual servers in a cluster.

JDBC Data Source

A JDBC data source is an object bound to the JNDI tree that points to a JDBC connection pool or MultiPool. Applications can use a JDBC data source to get a database connection from a connection pool or MultiPool.

To create a JDBC data source, click the [Configure a new JDBC Data Source](#) link.

- For more information about creating a JDBC data source, see [“Creating and Configuring a JDBC Data Source”](#) on page 141-5.
- To create a JDBC connection pool, see [“Creating and Configuring a JDBC Connection Pool”](#) on page 127-4.
- To learn more about JDBC data sources, see [“Configuring JDBC DataSources”](#) on page 141-1 and [Introduction to WebLogic JDBC](#) in *Programming WebLogic JDBC*.



JDBC Data Source Assistant --> Target the data source

[Tasks](#) [Related Topics](#)

Overview

On this page, you select the servers and clusters on which you want to create and deploy the new data source. You should deploy the data source to the same servers and clusters on which the underlying connection pool is deployed. By default, the JDBC Data Source Assistant selects the deployment targets on which the underlying connection pool is deployed. In almost all cases, you should accept the default selections.

If only there is only one server in your configuration, the JDBC Data Source Assistant automatically selects it, and does not display a list of potential deployment targets.

After you select the servers and clusters on which you to create the connection pool, click the Create button. WebLogic Server adds the data source to the configuration (`config.xml`) and creates the data source on each server and cluster that you select.

If an error message is displayed, the data source configuration is saved even though the data source may not actually be created on all targets. You should check the server window (standard out) and the JDBC log for information to help you trouble-shoot problems. You can deploy the data source again from the data source property pages. See [“Deploying a JDBC Data Source to a Server or Cluster” on page 141-7](#).

Tasks

[“Creating and Configuring a JDBC Data Source” on page 141-5](#)

Related Topics

[“JDBC Objects in a Cluster” on page 111-3](#)

[“Configuring JDBC DataSources” on page 141-1](#)

[“Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1](#)

[“Connection Pool and Data Source Configuration Guidelines” on page 127-15](#)

JDBC Data Source Assistant --> Configure the data source

[Tasks](#) [Related Topics](#)

Overview

Enter the following information:

Name—The name of this JDBC data source. This name is used in the configuration file (`config.xml`) and throughout the Administration Console whenever referring to this data source.

JNDI Name—The JNDI path to where this JDBC data source is bound.

Honor Global Transactions—Select this check box (the default) to enable global transactions using this data source. Clear this check box to disable (ignore) global transactions using this data source. In most cases, you should leave the option selected. See [“When to Enable Global Transactions in a Data Source” on page 141-2](#).

Emulate Two-Phase Commit for non-XA Driver—Select this check box to enable connections from the associated connection pool to participate in global transactions by emulating two-phase commit. This option applies when the connection pool uses a non-XA JDBC driver to make database connections.

Use this option if the JDBC connection is the only participant in the transaction and there is no XA compliant JDBC driver available. With more than one resource participating in a transaction where one of them (the JDBC driver) is emulating an XA resource, you may see heuristic failures. See [“Emulating Two-Phase Commit” on page 141-3](#).

If this DataSource is associated with an XA connection pool, or if there is only one resource participating in the distributed transaction, this setting is ignored.

Tasks

[“Creating and Configuring a JDBC Data Source” on page 141-5](#)

Related Topics

[“Configuring JDBC DataSources” on page 141-1](#)

[“Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1](#)

[“Connection Pool and Data Source Configuration Guidelines” on page 127-15](#)

JDBC Data Source Assistant --> Connect to connection pool

[Tasks](#) [Related Topics](#)

Overview

This page includes the following option:

Pool Name—Select the connection pool to which the data source will point. Applications get a connection from the underlying connection pool by looking up the data source on the JNDI tree and then requesting a connection from the data source. The pool you select will be used to provide the connection when applications request a connection from this data source.

Note: The list of connection pools is constrained by the Honor Global Transactions option setting in the previous step. If you deselect the option, the list of connection pools only includes connection pools that use a non-XA JDBC driver and MultiPools configured in your domain.

If there are no appropriate connection pools, you cannot create the data source. You must first create a connection pool. For data sources without support for global transactions, you must create a connection pool that uses a non-XA JDBC driver.

Tasks

[“Creating and Configuring a JDBC Data Source” on page 141-5](#)

Related Topics

[“Configuring JDBC DataSources” on page 141-1](#)

[“Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1](#)

[“Connection Pool and Data Source Configuration Guidelines” on page 127-15](#)

JDBC Data Source --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

On the JDBC Data Source—Configuration tab, you specify attributes for the selected Data Source. Applications get a database connection from a connection pool by looking up a data source on the Java Naming and Directory Interface (JNDI) tree and then request a connection. The connection pool associated with the datasource provides the connection to the application.

Tasks

[“Creating and Configuring a JDBC Data Source” on page 141-5](#)

[“Cloning a JDBC Data Source” on page 141-6](#)

[“Deploying a JDBC Data Source to a Server or Cluster” on page 141-7](#)

Related Topics

[“Configuring JDBC DataSources” on page 141-1](#)

[“Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1](#)

[“Monitoring Transactions” on page 237-5](#)

Attributes

Table 136-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this JDBC data source. This name is used to represent the JDBC data source in the Administration Console and in the configuration file (<code>config.xml</code>). Applications looking to request a connection do not use this name to look up the data source. Instead, they use the JNDI name.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCTxDataSourceMBean</code></p> <p><i>Attribute:</i> <code>Name</code></p>	
JNDI Name	<p>The JNDI path to where this <code>TxDataSource</code> is bound.</p> <p>Applications that look up the JNDI path will get a <code>javax.sql.DataSource</code> instance that corresponds to this <code>DataSource</code>.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCTxDataSourceMBean</code></p> <p><i>Attribute:</i> <code>JNDIName</code></p>	

Advanced Attributes

Table 136-2

Attribute Label	Description	Value Constraints
Row Prefetch Enabled	<p>Controls row prefetching between a client and WebLogic Server for each ResultSet. When an external client accesses a database using JDBC through WebLogic Server, row prefetching improves performance by fetching multiple rows from the server to the client in one server access. WebLogic Server will ignore this setting and not use row prefetching when the client and WebLogic Server are in the same JVM.</p> <p><i>MBean:</i> weblogic.management.configuration.JDBCTxDataSourceMBean</p> <p><i>Attribute:</i> RowPrefetchEnabled</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false
Row Prefetch Size	<p>The number of result set rows to prefetch for a client. The optimal value depends on the particulars of the query. In general, 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.</p> <p><i>MBean:</i> weblogic.management.configuration.JDBCTxDataSourceMBean</p> <p><i>Attribute:</i> RowPrefetchSize</p>	<p><i>Minimum:</i> 2</p> <p><i>Maximum:</i> 65536</p> <p><i>Default:</i> 48</p>

Table 136-2

Attribute Label	Description	Value Constraints
Stream Chunk Size	<p>Data chunk size for steaming data types. Streaming data types (for example resulting from a call to <code>getBinaryStream()</code>) will be pulled in <code>StreamChunkSize</code> sized chunks from the WebLogic Server to the client as needed.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCTxDataSourceMBean</code></p> <p><i>Attribute:</i> <code>StreamChunkSize</code></p>	<p><i>Units:</i> bytes</p> <p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 65536</p> <p><i>Default:</i> 256</p>
Honor Global Transactions	<p>Specifies whether the data source can participate in a global transaction. When selected, the data source is created as a <code>TxDataSource</code> in the <code>config.xml</code> file.</p> <p>You select this option (the default) when creating the data source. You cannot change this setting after the data source is created.</p>	<p><i>Default:</i> true</p>

Table 136-2

Attribute Label	Description	Value Constraints
Emulate Two-Phase Commit for non-XA Driver	<p>When set to true, this attribute allows a non-XA JDBC driver to emulate participation in distributed transactions using JTA.</p> <p>Use this option if the JDBC connection is the only participant in the transaction and there is no XA compliant JDBC driver available. With more than one resource participating in a transaction where one of them (the JDBC driver) is emulating an XA resource, you may see heuristic failures. See “Emulating Two-Phase Commit” on page 141-3.</p> <p>If this DataSource is associated with an XA connection pool, or if there is only one resource participating in the distributed transaction, then this setting is ignored.</p> <p>Note: This option is only available when Honor Global Transactions is selected.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCTxDataSourceMBean</code></p> <p><i>Attribute:</i> <code>EnableTwoPhaseCommit</code></p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false



JDBC Data Source --> Notes

[Tasks](#) [Related Topics](#)

Overview

On the JDBC Data Source—Notes tab, you can add a note to the data source configuration. You can use this to save information about the data source that is not apparent in the configuration, such as a history of configuration changes and reasons.

Tasks

[“Adding a Note to a Data Source” on page 141-7](#)

[“Creating and Configuring a JDBC Data Source” on page 141-5](#)

Related Topics

[“Configuring JDBC DataSources” on page 141-1](#)

[“Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1](#)

[“Monitoring Transactions” on page 237-5](#)



JDBC Data Source --> Target and Deploy

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

On the JDBC Data Source—Target and Deploy tab, you deploy the selected data source on servers and clusters in the domain.

You should deploy the data source to the same target servers and clusters on which the underlying connection pool is deployed.

Tasks

“Deploying a JDBC Data Source to a Server or Cluster” on page 141-7

“Creating and Configuring a JDBC Data Source” on page 141-5

Related Topics

“Configuring JDBC DataSources” on page 141-1

“Connection Pool and Data Source Configuration Guidelines” on page 127-15

“Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1

Attributes

You should deploy the data source to the same target servers and clusters on which the underlying connection pool is deployed.

Independent Servers—Lists the servers in your configuration on which you can deploy the data source. Select servers and click Apply.

Clusters—Lists clusters and comprising servers on which you can deploy the data source. Select the cluster on which you want to deploy the data source and click Apply. In almost all cases, you should deploy to an entire cluster, not individual servers in a cluster.

JDBC Data Source Factory --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

A JDBC data source factory is an instance of a JDBC data source resource bound to the WebLogic Server JNDI tree as a resource factory. An application-scoped JDBC connection pool relies on a JDBC data source factory to provide default connection pool values. You must create a data source factory before you deploy an enterprise application that includes an application-scoped connection pool. The `weblogic-application.xml` supplemental deployment descriptor for the application must reference the data source factory by the factory name that you specify in the Administration Console.

Properties that you specify for a data source factory are used as default values for application-scoped connection pools that reference the factory. You can override the default values in the `weblogic-application.xml` supplemental deployment descriptor.

Tasks

[“Creating and Configuring a JDBC Data Source Factory” on page 141-8](#)

Related Topics

[“Application-Scoped JDBC Data Sources and Connection Pools” on page 127-9](#)

Resource Factories in [Programming WebLogic Server Enterprise JavaBeans](#)

Attributes

Table 139-1

Attribute Label	Description	Value Constraints
Name	The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration. <i>MBean:</i> weblogic.management.configuration.JDBCDataSourceFactoryMBean <i>Attribute:</i> Name	<i>Dynamic:</i> no
User Name	The database user name. This may be overridden by user-name in the descriptor. <i>MBean:</i> weblogic.management.configuration.JDBCDataSourceFactoryMBean <i>Attribute:</i> UserName	<i>Dynamic:</i> no
URL	The connection URL. This may be overridden by url in the descriptor <i>MBean:</i> weblogic.management.configuration.JDBCDataSourceFactoryMBean <i>Attribute:</i> URL	<i>Dynamic:</i> no
Driver Class Name	The name of the driver. This may be overridden by driver-name in the descriptor. <i>MBean:</i> weblogic.management.configuration.JDBCDataSourceFactoryMBean <i>Attribute:</i> DriverClassName	<i>Dynamic:</i> no

Table 139-1

Attribute Label	Description	Value Constraints
Factory Name	The name of this data source factory. This is referenced from the connection-factory element in <code>weblogic-application.xml</code> <i>MBean:</i> <code>weblogic.management.configuration.JDBCDataSourceFactoryMBean</code> <i>Attribute:</i> <code>FactoryName</code>	<i>Dynamic:</i> no
Properties	Connection properties passed to the JDBC driver to create physical database connections. <i>MBean:</i> <code>weblogic.management.configuration.JDBCDataSourceFactoryMBean</code> <i>Attribute:</i> <code>Properties</code>	<i>Dynamic:</i> no



JDBC Data Source Factory

A JDBC Data Source Factory is an instance of a JDBC data source resource bound to the WebLogic Server JNDI tree as a resource factory. Using resource factories enables the EJB to map a resource factory reference in the EJB deployment descriptor to an available resource factory in a running WebLogic Server. The EJB can then use a JDBC data source factory to get a database connection from a connection pool.

To create a JDBC data source factory, click the [Configure a new JDBC Data Source Factory](#) link.

- For more information about creating a JDBC data source factory, see [“Creating and Configuring a JDBC Data Source Factory”](#) on page 141-8.



1 JDBC DataSources

[“Attributes and Console Screen Reference for JDBC Data Sources” on page 142-1]

This section includes the following subsections:

- “Configuring JDBC DataSources” on page 141-1
- “JDBC Data Source Factories” on page 141-8
- “Application-Scoped JDBC Data Sources” on page 141-9
- “Monitoring Data Sources” on page 141-9

Configuring JDBC DataSources

A Data Source object enables JDBC applications to obtain a DBMS connection from a connection pool. Each Data Source object binds to the JNDI tree and points to a connection pool or MultiPool. Applications look up the Data Source on the JNDI tree and then request a connection from the Data Source. Data Source objects can be defined with support for global transactions (enables support for JTA). Support for global transactions is required if your applications use distributed transactions. See [When to Enable Global Transactions in a Data Source](#) for more information about using Data Sources and transaction-enabled Data Sources.

When to Enable Global Transactions in a Data Source

If your applications or environment meet any of the following criteria, you should enable global transactions in the Data Source (select Honor Global Transactions during configuration). This creates a Tx Data Source in the `config.xml` file.

- Use the Java Transaction API (JTA)
- Use the EJB container in WebLogic Server to manage transactions
- Include multiple database updates within a single transaction
- Access multiple resources, such as a database and the Java Messaging Service (JMS), during a transaction
- Use the same connection pool on multiple servers

When not to enable global transactions:

- The underlying connection pool uses a non-XA JDBC driver
- You don't plan to use distributed transactions in your applications.
- You plan to use database definition language (DDL) SQL commands through this data source (only applies to some DBMSs or JDBC drivers)

With an EJB architecture, it is common for multiple EJBs that are doing database work to be invoked as part of a single transaction. Without XA, the only way for this to work is if all transaction participants use the exact same database connection. WebLogic Server uses the JTS driver and a Tx Data Source to do this behind the scenes without requiring you to explicitly pass the JDBC connection from EJB to EJB.

If multiple EJBs are participating in a transaction and you do not use an XA JDBC driver for database connections, configure a Data Source with the following options:

- Honor Global Transaction selected (creates a Tx Data Source)
- Emulate Two-Phase Commit selected
- A non-XA connection pool as the Pool Name

This configuration will force the JTS driver to internally use the same database connection for all database work within the same transaction.

With XA (requires an XA driver), you can use a Tx Data Source in WebLogic Server for distributed transactions with two-phase commit so that EJBs can use a different database connection for each part of the transaction. In either case (with or without XA), you should use a Tx Data Source (a data source with Honor Global Transactions selected).

Note: Do not create two Tx Data Sources that point to the same connection pool. If a transaction uses two different Tx Data Sources which are both pointed to the same connection pool, you will get an XA_PROTO error when you try to access the second connection.

Read more about [Data Sources](#) in *Programming WebLogic JDBC at {DOCR00T}/jdbc/programming.html*.

Note: In previous releases, Data Sources and Tx Data Sources were listed as different object types in the Administration Console. In this release, Data Sources and Tx Data Sources are listed as Data Sources in the Administration console.

Emulating Two-Phase Commit

If you need to support distributed transactions with a JDBC connection pool, but there is no available XA-compliant driver for your DBMS, you can select the Emulate Two-Phase Commit for non-XA Driver option for a data source to emulate two-phase commit for the transactions in which the connection pool participates. This option is an advanced option on the Data Source—Configuration tab.

When the Emulate Two-Phase Commit for non-XA Driver option is selected (EnableTwoPhaseCommit is set to true), the non-XA JDBC resource always returns XA_OK during the XAResource.prepare() method call. The resource attempts to commit or roll back its local transaction in response to subsequent XAResource.commit() or XAResource.rollback() calls. If the resource commit or rollback fails, a heuristic error results. Application data may be left in an inconsistent state as a result of a heuristic failure.

When the Emulate Two-Phase Commit for non-XA Driver option is not selected in the Console (EnableTwoPhaseCommit is set to false), the non-XA JDBC resource causes XAResource.prepare() to fail. When there is only one resource participating in a transaction, the one phase optimization bypasses XAResource.prepare(), and the transaction commits successfully in most instances.

See “Configuring Non-XA JDBC Drivers for Distributed Transactions” on page 127-27 for more information.

Note: There are risks to data integrity when using the Emulate Two-Phase Commit for non-XA Driver option. BEA recommends that you use an XA-compliant JDBC driver rather than use this option. Make sure you consider the risks below before enabling this option.

This non-XA JDBC driver support is often referred to as the "JTS driver" because WebLogic Server uses the WebLogic JTS Driver internally to support the feature. For more information about the WebLogic JTS Driver, see "[Using the WebLogic JTS Driver](#)" in *Programming WebLogic JDBC*.

Limitations and Risks When Using a Non-XA Driver in Global Transactions

WebLogic Server supports the participation of non-XA JDBC resources in global transactions, but there are limitations that you must consider when designing applications to use such resources. Because a non-XA driver does not adhere to the XA/2PC contracts and only supports one-phase commit and rollback operations, WebLogic Server (through the JTS driver) has to make compromises to allow the resource to participate in a transaction controlled by the Transaction Manager.

Consider the following limitations and risks before using the Emulate Two-Phase Commit for non-XA Driver option.

Heuristic Completions and Data Inconsistency

When Emulate Two-Phase Commit is selected for a non-XA resource, (`enableTwoPhaseCommit = true`), the prepare phase of the transaction for the non-XA resource always succeeds. Therefore, the non-XA resource does not truly participate in the two-phase commit (2PC) protocol and is susceptible to failures. If a failure occurs in the non-XA resource after the prepare phase, the non-XA resource is likely to roll back the transaction while XA transaction participants will commit the transaction, resulting in a heuristic completion and data inconsistencies.

Because of the data integrity risks, the Emulate Two-Phase Commit option should only be used in applications that can tolerate heuristic conditions.

Cannot Recover Pending Transactions

Because a non-XA driver manipulates local database transactions only, there is no concept of a transaction pending state in the database with regard to an external transaction manager. When `XAResource.recover()` is called on the non-XA resource, it always returns an empty set of Xids (transaction IDs), even though there may be transactions that need to be committed or rolled back. Therefore, applications that use a non-XA resource in a global transaction cannot recover from a system failure and maintain data integrity.

Possible Performance Loss with Non-XA Resources in Multi-Server Configurations

Because WebLogic Server relies on the database local transaction associated with a particular JDBC connection to support non-XA resource participation in a global transaction, when the same JDBC data source is accessed by an application with a global transaction context on multiple WebLogic Server instances, the JTS driver will always route JDBC operations to the first connection established by the application in the transaction. For example, if an application starts a transaction on one server, accesses a non-XA JDBC resource, then makes a remote method invocation (RMI) call to another server and accesses a data source that uses the same underlying JDBC driver, the JTS driver recognizes that the resource has a connection associated with the transaction on another server and sets up an RMI redirection to the actual connection on the first server. All operations on the connection are made on the one connection that was established on the first server. This behavior can result in a performance loss due to the overhead associated with setting up these remote connections and making the RMI calls to the one physical connection.

Only One Non-XA Participant

When a non-XA resource (with Emulate Two-Phase Commit selected) is registered with the WebLogic Server Transaction Manager, it is registered with the name of the class that implements the `XAResource` interface. Since all non-XA resources with Emulate Two-Phase Commit selected use the JTS driver for the `XAResource` interface, all non-XA resources (with Emulate Two-Phase Commit selected) that participate in a global transaction are registered with the same name. If you use more than one non-XA resource in a global transaction, you will see naming conflicts or possible heuristic failures.

Creating and Configuring a JDBC Data Source

1. In the left pane, 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. The JDBC Data Source Assistant opens in the right pane.
4. In Step 1, enter the following information:

Name—Enter a name for this JDBC data source. This name is used in the configuration file (`config.xml`) and throughout the Administration Console whenever referring to this data source.

JNDI Name—Enter the JNDI path to where this JDBC data source will be bound.

Honor Global Transactions—Select this check box (the default) to enable global transactions using this data source. Clear this check box to disable (ignore) global transactions using this data source. In most cases, you should leave the option selected. See [“When to Enable Global Transactions in a Data Source” on page 141-2](#).

Emulate Two-Phase Commit for non-XA Driver—Select this check box to enable connections from the associated non-XA connection pool to participate in global transactions by emulating two-phase commit. Use caution when selecting this option. See [“Limitations and Risks When Using a Non-XA Driver in Global Transactions” on page 141-4](#).

Click Next to continue.

5. In Step 2, select the connection pool to which the data source will point. Applications get a connection from the underlying connection pool by looking up the data source on the JNDI tree and then requesting a connection from the data source. The pool you select will be used to provide the connection when applications request a connection from this data source.

Click Next to continue.

6. In Step 3, select the servers and clusters on which you want to create and deploy the new data source. You should deploy the data source to the same servers and clusters on which the underlying connection pool is deployed. By default, the

JDBC Data Source Assistant selects the deployment targets on which the underlying connection pool is deployed. In almost all cases, you should accept the default selections.

7. Click Create.

Cloning a JDBC Data Source

1. In the left pane, click to expand the JDBC node.
2. Click Data Sources to view any configured data sources for your domain.
3. Right-click the data source you want to clone and select *Clone Data Source name*. A dialog displays in the right pane showing the tabs associated with cloning a data source.
4. Enter values in the Name and JNDI Name attribute fields. Optionally, you can modify the other attribute values on the Configuration tab.
Note: You must specify a new JNDI Name for the new data source. The new data source cannot use the same JNDI Name as the original data source.

For more information about the attributes on the Configuration tab, see [Attributes](#).

5. Click Clone to create a data source with the attributes you specified on the Configuration tab and with cloned values on all other tabs. The new data source is added under the Data Sources node in the left pane.
6. Optionally, click the remaining tabs for the data source and change the attribute fields or accept the current values. Click Apply to save any changes you make.

Deploying a JDBC Data Source to a Server or Cluster

1. In the left pane, click to expand the JDBC and Data Sources nodes.
2. Click the data source you want to assign. A dialog displays in the right pane showing the tabs with attributes for the data source.

3. Click the Targets tab and select the servers and clusters on which you want to create and deploy the new data source. You should deploy the data source to the same servers and clusters on which the underlying connection pool is deployed. By default, the JDBC Data Source Assistant selects the deployment targets on which the underlying connection pool is deployed.
4. Click Apply to save your changes.

Adding a Note to a Data Source

1. In the left pane, click to expand the JDBC node.
2. Click the Data Sources node to expand it and show the list of data sources defined in your domain.
3. Click the data source to which you want to add a note. A dialog displays in the right pane showing tabs with attributes for the selected object.
4. Click the Notes tab. Type the note in the Notes field.
5. Click Apply to save your changes.

Deleting a Data Source

1. In the left pane, click to expand the JDBC node, then click to expand the Data Sources node to display the list of data sources in the current domain.
2. Right-click the object you want to delete and select *Delete object name*. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click Yes to delete the data source.

JDBC Data Source Factories

An application-scoped JDBC connection pool relies on a JDBC data source factory to provide default connection pool values. You must create a data source factory before you deploy an enterprise application that includes an application-scoped connection pool. The `weblogic-application.xml` supplemental deployment descriptor for the application must reference the data source factory by the factory name that you specify in the Administration Console.

For more information about application-scoped JDBC connection pools, see “Application-Scoped JDBC Data Sources and Connection Pools” on page 127-9.

For more information about resource factories, see [Programming WebLogic Enterprise JavaBeans at {DOCROOT}/ejb/index.html](#).

Creating and Configuring a JDBC Data Source Factory

1. In the left pane, click Services and expand JDBC.
2. Select JDBC Data Source Factory, and in the right pane click the Configure a New JDBC Data Source Factory text link.
3. Enter values in the attribute fields to use as the default values for application-scoped connection pools that reference this data source factory.
4. Click Create to create the JDBC Data Source Factory. The new Data Source Factory is added under the Data Source Factories node in the left pane.

Application-Scoped JDBC Data Sources

See “Application-Scoped JDBC Data Sources and Connection Pools” on page 127-9.

Monitoring Data Sources

There are no monitoring options for data sources in WebLogic Server. However, you can monitor connection pools and transactions. See the following sections for more information:

“Monitoring Connections in a JDBC Connection Pool” on page 127-36

“Monitoring Transactions” on page 237-5

Attributes and Console Screen Reference for JDBC Data Sources

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“JDBC Data Source” on page 132-1](#)

[“JDBC Data Source --> Configuration” on page 136-1](#)

[“JDBC Data Source --> Target and Deploy” on page 138-1](#)

[“JDBC Data Source --> Notes” on page 137-1](#)

[“JDBC Data Source --> Configuration” on page 129-1](#)

[“JDBC Data Source --> Target and Deploy” on page 131-1](#)

[“JDBC Data Source --> Notes” on page 130-1](#)

[“JDBC Data Source Assistant --> Configure the data source” on page 134-1](#)

[“JDBC Data Source Assistant --> Connect to connection pool” on page 135-1](#)

[“JDBC Data Source Assistant --> Target the data source” on page 133-1](#)

[“JDBC Data Source Factory” on page 140-1](#)

[“JDBC Data Source Factory --> Configuration” on page 139-1](#)



JDBC MultiPool --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

On the JDBC MultiPool—Configuration—General tab, you specify general attributes for the MultiPool. A MultiPool is a "pool of connection pools."

Note: MultiPools are not supported for use in distributed transactions. You cannot use connections from a connection pool within a MultiPool if the connection pool uses an XA driver to create database connections.

Tasks

[“Creating and Configuring a JDBC MultiPool” on page 149-2](#)

[“Cloning a JDBC MultiPool” on page 149-3](#)

[“Deploying a JDBC MultiPool on One or More Servers and Clusters” on page 149-3](#)

[“Creating and Configuring a JDBC Connection Pool” on page 127-4](#)

Related Topics

[“JDBC MultiPools” on page 149-1](#)

[“Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1](#)

Attributes

Table 143-1

Attribute Label	Description	Value Constraints
Name	The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration. <i>MBean:</i> weblogic.management.configuration.JDBCMultiPoolMBean <i>Attribute:</i> Name	

Table 143-1

Attribute Label	Description	Value Constraints
Algorithm Type	<p>The algorithm type for this Multipool.</p> <p>If set to "High availability", the connection pools are set up as an ordered list. That is, every time an application asks the Multipool 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. The process is repeated until a valid connection is obtained, or until the end of the list is reached, in which case an exception will be thrown.</p> <p>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.</p> <p>If the algorithm is set to "Load balancing," the Multipool will distribute the connection requests evenly to its member pools. This algorithm also performs the same failover behavior as the high availability algorithm.</p> <p>Default value for this attribute is "High availability".</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JDBCMultiPoolMBean</code></p> <p><i>Attribute:</i> <code>AlgorithmType</code></p>	<p><i>Default:</i> High-Availability</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ High-Availability ■ Load-Balancing



JDBC MultiPool --> Configuration --> Pools

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

On the JDBC MultiPool—Configuration—Pools tab, you select the connection pools to include in the multipool. A multipool is a "pool of connection pools."

Note: MultiPools are not supported for use in distributed transactions. You cannot use connections from a connection pool within a MultiPool if the connection pool uses an XA driver to create database connections.

Tasks

[“Creating and Configuring a JDBC MultiPool” on page 149-2](#)

[“Cloning a JDBC MultiPool” on page 149-3](#)

[“Deploying a JDBC MultiPool on One or More Servers and Clusters” on page 149-3](#)

[“Creating and Configuring a JDBC Connection Pool” on page 127-4](#)

Related Topics

[“JDBC MultiPools” on page 149-1](#)

[“Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1](#)

[Using MultiPools](#) in *Programming WebLogic JDBC*

Attributes

Table 144-1

Attribute Label	Description	Value Constraints
Pool List	The list of connection pools in the MultiPool. <i>MBean:</i> weblogic.management.configuration.JDBCMultiPoolMBean <i>Attribute:</i> PoolList	

JDBC MultiPool --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

On the JDBC MultiPool—Notes tab, you can add a note to save information about the MultiPool. A MultiPool is a "pool of JDBC connection pools."

Tasks

[“Adding a Note to a JDBC MultiPool” on page 149-4](#)

[“Creating and Configuring a JDBC MultiPool” on page 149-2](#)

Related Topics

[“JDBC MultiPools” on page 149-1](#)

Attributes

Table 145-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.JDBCMultiPoolMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

JDBC MultiPool --> Target and Deploy

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

On this tab, you select the servers and clusters on which you want to deploy the MultiPool. When you deploy a MultiPool on a server, WebLogic Server creates an instance of the MultiPool on the server. If you deploy the MultiPool on two servers, WebLogic Server creates the MultiPool on each server. Each instance is configured the same, but is independent of the other. If you deploy the MultiPool to a cluster, WebLogic Server creates an instance of the connection pool on each server in the cluster unless you specify otherwise.

You should deploy MultiPools to the same servers and clusters on which the underlying connection pools are deployed. Also, all underlying connection pools should be deployed on the same target servers and clusters.

Tasks

“Deploying a JDBC MultiPool on One or More Servers and Clusters” on page 149-3

Related Topics

“Configuring MultiPools” on page 149-1

“Configuring JDBC Connection Pools” on page 127-1

“Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1

Attributes

Independent Servers—Lists the servers in your configuration on which you can deploy the MultiPool. Select servers and click Apply.

Clusters—Lists clusters and comprising servers on which you can deploy the MultiPool. Select the cluster on which you want to deploy the MultiPool and click Apply. In almost all cases, you should deploy to an entire cluster, not individual servers in a cluster. When you deploy the connection pool to a cluster, WebLogic Server creates an instance of the connection pool on each server in the cluster.

JDBC MultiPool

A MultiPool is a pool of connection pools configured for load balancing or high availability. Typically, you create connection pools, then create a MultiPool and assign connection pools to the MultiPool.

Note: MultiPools are not supported for use in distributed transactions. You cannot use connections from a connection pool within a MultiPool if the connection pool uses an XA driver to create database connections.

To create a MultiPool, click the Configure a new JDBC MultiPool link.

- To create a MultiPool, see [“Creating and Configuring a JDBC MultiPool” on page 149-2](#).
- To create a JDBC connection pool, see [“Creating and Configuring a JDBC Connection Pool” on page 127-4](#).
- To create a JDBC data source, see [“Creating and Configuring a JDBC Data Source” on page 141-5](#).

To learn more about MultiPools, see:

- “Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1
- “JDBC MultiPools” on page 149-1



JDBC Connection Leak Profile

The Connection Leak Profile page displays leaked connections. Connection leaks result from application components such as EJBs failing to close connections when they are no longer needed.

Only leaked connections for non-transactional connections are displayed.



1 JDBC MultiPools

[“Attributes and Console Screen Reference for JDBC MultiPools” on page 150-1]

A MultiPool is a pool of connection pools. Used in local (non-distributed) transactions, MultiPools aid in either:

- Load Balancing—pools are accessed using a round-robin scheme. When switching connections, WebLogic Server selects a connection from the next connection pool in the order listed.
- High Availability—connection pools are listed in the order that determines the order in which connection pool switching occurs. That is, WebLogic Server provides database connections from the first connection pool on the list. If that connection pool fails, it attempts to use a database connection from the second, and so forth.

Note: MultiPools are not supported for use in distributed transactions. You cannot use connections from a connection pool within a MultiPool if the connection pool uses an XA driver to create database connections.

For more general information about MultiPools, see "[Configuring and Using MultiPools](#)" in *Programming WebLogic JDBC*.

Configuring MultiPools

The following sections provide detailed instructions for configuring MultiPools.

Creating and Configuring 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:
 - a. Enter a value in the Name attribute field.
 - b. Select one of the following options in the Algorithm Type drop-down list:

Load Balancing—Select this option if you want to use connections from all connection pools in the MultiPool to balance the load between connection pools and underlying DBMS servers.

High Availability—Select this option if you want to use all available connections in one pool before using connections in another.
5. Click Create to create a MultiPool instance with the name and attributes you specified on the General tab. 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 changes.

You must deploy the MultiPool before you can use connections from it. See “Deploying a JDBC MultiPool on One or More Servers and Clusters” on page 149-3.

Cloning a JDBC MultiPool

1. In the left pane, click to expand the JDBC and MultiPool nodes.
2. Right-click the MultiPool you want to clone and select *Clone MultiPool name*. A dialog displays in the right pane showing the tabs associated with cloning a MultiPool.
3. Enter a value in the Name attribute field and optionally select a value for Algorithm Type.
4. Click Clone to create a MultiPool with the attributes you specified on the General tab and with cloned values on all other tabs. The new MultiPool is added under the MultiPools node in the left pane.
5. Optionally, click the remaining tabs for the MultiPool and change the attribute fields or accept the current values.
6. Click Apply to save any changes you make.

Deploying a JDBC MultiPool on One or More Servers and Clusters

1. In the left pane, click to expand the JDBC and MultiPool nodes.
2. Click the MultiPool you want to assign. A dialog displays in the right pane showing the tabs associated with this MultiPool.
3. Click the Target and Deploy tab and select the servers or clusters on which you want to deploy the MultiPool. You should deploy the MultiPool on the same deployment targets that the underlying connection pools are deployed on. Click Apply to save your changes.

When deploying a MultiPool on a cluster, in most cases you should deploy the MultiPool to the entire cluster. You should deploy the related data source and underlying connection pools on the same targets.

Adding a Note to a JDBC MultiPool

1. In the left pane, click the JDBC node to expand it.
2. Click the MultiPools node to expand it and show the list of multipools defined in your domain.
3. Click the MultiPool to which you want to add a note. A dialog displays in the right pane showing tabs with attributes for the MultiPool.
4. Click the Notes tab. Type the note in the Notes field.
5. Click Apply to save your changes.

Deleting a JDBC MultiPool

1. In the left pane, click to expand the JDBC and MultiPools nodes to display the list of MultiPools in the current domain.
2. Right-click the MultiPool you want to delete and select Delete *MultiPool name*. A dialog displays in the right pane asking you to confirm your deletion request.
3. Click Yes to delete the MultiPool.

Attributes and Console Screen Reference for JDBC MultiPools

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“JDBC MultiPool” on page 147-1](#)

[“JDBC MultiPool --> Configuration --> General” on page 143-1](#)

[“JDBC MultiPool --> Configuration --> Pools” on page 144-1](#)

[“JDBC MultiPool --> Target and Deploy” on page 146-1](#)

[“JDBC MultiPool --> Notes” on page 145-1](#)



JMS Connection Consumer --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines general configuration attributes for a JMS connection consumer, which are queues (Point-To-Point) or topics (Pub/Sub) that retrieve server sessions and process messages. Once you have defined a session pool, you can configure one or more connection consumers for each session pool.

Tasks

[“Creating a JMS Connection Consumer” on page 232-30](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 151-1

Attribute Label	Description	Value Constraints
Name	The name of this JMS connection consumer configuration. WebLogic Server uses an MBean to implement and persist the configuration. <i>MBean:</i> weblogic.management.configuration.JMSConnectionConsumerMBean <i>Attribute:</i> Name	
Messages Maximum	The maximum number of messages that the connection consumer can load at one time into a server session. <i>MBean:</i> weblogic.management.configuration.JMSConnectionConsumerMBean <i>Attribute:</i> MessagesMaximum	<i>Minimum:</i> -1 <i>Default:</i> -1 <i>Dynamic:</i> yes
Selector	The JMS message selector for filtering the connection consumer's messages. <i>MBean:</i> weblogic.management.configuration.JMSConnectionConsumerMBean <i>Attribute:</i> Selector	<i>Default:</i> null
Destination	The JNDI name of the associated destination (queue or topic) for the connection consumer. <i>MBean:</i> weblogic.management.configuration.JMSConnectionConsumerMBean <i>Attribute:</i> Destination	

JMS Connection Consumer --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to supply optional information about this JMS connection consumer configuration.

Tasks

[“Creating a JMS Connection Consumer” on page 232-30](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 152-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.JMSConnectionConsumerMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

JMS Connection Consumer

Connection consumers are queues (Point-To-Point) or topics (Pub/Sub) that retrieve server sessions and process messages. Once you have defined a JMS session pool, you can configure one or more connection consumers for each session pool.

To configure a new JMS connection consumer, click the Configure a new JMS Connection Consumer link.

- For more information on creating a JMS connection consumer, see [“Creating a JMS Connection Consumer” on page 232-30](#).
- For more information on creating a JMS session pool, see [“Creating a JMS Session Pool” on page 232-29](#).



JMS Connection Factory --> Configuration --> Flow Control

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines the flow control configuration attributes that instruct a message producer to adjust its message flow during peak message load. After defining a JMS server, you can configure one or more connection factories to create connections with predefined attributes.

Tasks

[“Configuring a JMS Connection Factory” on page 232-10](#)

[“Deploying a Connection Factory on Multiple Individual Servers” on page 232-12](#)

[“Controlling the Flow of Messages on JMS Servers and Destinations” on page 235-20](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 154-1

Attribute Label	Description	Value Constraints
Flow Maximum	<p>The maximum number of messages per second allowed for a producer that is experiencing a threshold condition on the JMS server or destination (queue or topic). When a producer is flow controlled it will never be allowed to go faster than the FlowMaximum messages per second.</p> <p>If a producer is not currently limiting its flow when a threshold condition is reached, the initial flow limit for that producer is set to FlowMaximum. If a producer is already limiting its flow when a threshold condition is reached (the flow limit is less than FlowMaximum), then the producer will continue at its current flow limit until the next time the flow is evaluated.</p> <p>Note: Once a threshold condition has subsided, the producer is not permitted to ignore its flow limit. If its flow limit is less than the FlowMaximum, then the producer must gradually increase its flow to the FlowMaximum each time the flow is evaluated. When the producer finally reaches the FlowMaximum, it can then ignore its flow limit and send without limiting its flow.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSConnectionFactoryMBean</code></p> <p><i>Attribute:</i> FlowMaximum</p>	<p><i>Units:</i> messages/second</p> <p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 500</p> <p><i>Dynamic:</i> yes</p>

Table 154-1

Attribute Label	Description	Value Constraints
Flow Minimum	<p>The minimum number of messages per second allowed for a producer that is experiencing a threshold condition on the JMS server or destination (queue or topic). This is the lower boundary of a producer's flow limit. That is, WebLogic JMS will not further slow down a producer whose message flow limit is at its FlowMinimum.</p> <p>When a producer is flow controlled it will never be required to go slower than FlowMinimum messages per second.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSConnectionFactoryMBean</code></p> <p><i>Attribute:</i> <code>FlowMinimum</code></p>	<p><i>Units:</i> messages/second</p> <p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 50</p> <p><i>Dynamic:</i> yes</p>
Flow Interval (seconds)	<p>The adjustment period of time, in seconds, when a producer adjusts its flow from the FlowMaximum number of messages to the FlowMinimum amount, or vice versa.</p> <p>When a producer is flow controlled, it is slowed down from its FlowMaximum to its FlowMinimum over the FlowInterval seconds.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSConnectionFactoryMBean</code></p> <p><i>Attribute:</i> <code>FlowInterval</code></p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 60</p> <p><i>Dynamic:</i> yes</p>

Table 154-1

Attribute Label	Description	Value Constraints
Flow Steps	<p>The number of steps used when a producer is adjusting its flow from the Flow Maximum amount of messages to the Flow Minimum amount, or vice versa. Specifically, the Flow Interval adjustment period is divided into the number of Flow Steps (for example, 60 seconds divided by 6 steps is 10 seconds per step).</p> <p>Also, the movement (i.e., the rate of adjustment) is calculated by dividing the difference between the Flow Maximum and the Flow Minimum into steps. At each Flow Step, the flow is adjusted upward or downward, as necessary, based on the current conditions, as follows:</p> <ul style="list-style-type: none">■ The downward movement (the decay) is geometric over the specified period of time (Flow Interval) and according to the specified number of Flow Steps. (For example, 100, 50, 25, 12.5).■ The movement upward is linear. The difference is simply divided by the number of steps. <p><i>MBean:</i> weblogic.management.configuration.JMSConnectionFactoryMBean</p> <p><i>Attribute:</i> FlowSteps</p>	<p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 10</p> <p><i>Dynamic:</i> yes</p>
Flow Control Enabled	<p>Specifies whether flow control is enabled for a producer created using this connection factory. If true, the associated message producers will be slowed down if the JMS server reaches Bytes/MessagesThresholdHigh.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSConnectionFactoryMBean</p> <p><i>Attribute:</i> FlowControlEnabled</p>	<p><i>Default:</i> true</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false <p><i>Dynamic:</i> yes</p>

Table 154-1

Attribute Label	Description	Value Constraints
Send Timeout (milliseconds)	<p>The maximum length of time, in milliseconds, that a sender will wait when there isn't enough available space (no quota) on a destination to accommodate the message being sent.</p> <p>The default time is 10 milliseconds. A value of 0 indicates that the sender does not want to wait for space.</p> <p>This attribute is dynamic. It can be changed at any time. However, changing the value does not affect existing connections or their producers. It only affects new connections made with this connection factory. Producers inherit the setting from the connection factory used to create their session and connection. The value can then be overridden at run time by setting the value on the producer.</p> <p>Note: Also see the JMS Server -> Configuration -> Thresholds & Quota -> Blocking Send Policy attribute.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSConnectionFactoryMBean</code></p> <p><i>Attribute:</i> <code>SendTimeout</code></p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> 10</p> <p><i>Dynamic:</i> yes</p>



JMS Connection Factory --> Configuration

--> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines general configuration attributes for a JMS connection factory. After defining a JMS server, you can configure one or more connection factories to create connections with predefined attributes.

Tasks

[“Configuring a JMS Connection Factory” on page 232-10](#)

[“Deploying a Connection Factory on Multiple Individual Servers” on page 232-12](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 155-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this JMS connection factory configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSConnectionFactoryMBean</p> <p><i>Attribute:</i> Name</p>	
JNDIName	<p>The JNDI name used to look up the connection factory within the JNDI namespace. The connection factory name is configured separately.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSConnectionFactoryMBean</p> <p><i>Attribute:</i> JNDIName</p>	
Client Id	<p>An optional client ID for a durable subscriber that uses this connection factory.</p> <p>Configuring this value prevents more than one JMS client from using a connection from the factory. Generally, JMS durable subscriber applications set their client IDs dynamically using the <code>javax.jms.Connection.setClientID()</code> call.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSConnectionFactoryMBean</p> <p><i>Attribute:</i> ClientId</p>	<i>Dynamic:</i> yes

Table 155-1

Attribute Label	Description	Value Constraints
Default Priority	<p>The default priority used for messages when a priority is not explicitly defined.</p> <p>Message producers can get the priority explicitly by calling the <code>javax.jms.MessageProducer.getPriority()</code> method.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSConnectionFactoryMBean</code></p> <p><i>Attribute:</i> <code>DefaultPriority</code></p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 9</p> <p><i>Default:</i> 4</p> <p><i>Dynamic:</i> yes</p>
Default Time To Live	<p>The maximum length of time, in milliseconds, that a message will exist. This value is used for messages when a priority is not explicitly defined. A value of 0 indicates that the message has an infinite amount time to live.</p> <p>Message producers can get the time-to-live explicitly by calling the <code>javax.jms.MessageProducer.getTimeToLive()</code> method.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSConnectionFactoryMBean</code></p> <p><i>Attribute:</i> <code>DefaultTimeToLive</code></p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> 0</p> <p><i>Dynamic:</i> yes</p>
Default Time To Deliver	<p>The delay time, in milliseconds, between when a message is produced and when it is made visible on its destination.</p> <p>Message producers can get the time-to-deliver explicitly by calling the <code>weblogic.jms.extensions.WLMes sageProducer.getTimeToDeliver()</code> method.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSConnectionFactoryMBean</code></p> <p><i>Attribute:</i> <code>DefaultTimeToDeliver</code></p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> 0</p> <p><i>Dynamic:</i> yes</p>

Table 155-1

Attribute Label	Description	Value Constraints
Default Delivery Mode	<p>The delivery mode assigned to all messages sent by a producer using this connection factory</p> <p>Message producers can get the delivery mode explicitly by calling the <code>javax.jms.MessageProducer.getDeliveryMode()</code> method.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSConnectionFactoryMBean</code></p> <p><i>Attribute:</i> <code>DefaultDeliveryMode</code></p>	<p><i>Default:</i> Persistent</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ Persistent■ Non-Persistent <p><i>Dynamic:</i> yes</p>
Default Redelivery Delay	<p>The delay time, in milliseconds, before rolled back or recovered messages are redelivered.</p> <p>Message consumers can get the redelivery delay explicitly by calling the <code>weblogic.jms.extensions.WLSession.getRedeliveryDelay()</code> method.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSConnectionFactoryMBean</code></p> <p><i>Attribute:</i> <code>DefaultRedeliveryDelay</code></p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> 0</p> <p><i>Dynamic:</i> yes</p>

Table 155-1

Attribute Label	Description	Value Constraints
Messages Maximum	<p>The maximum number of messages that can 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.</p> <p>When the number of messages reaches the MessagesMaximum value:</p> <ul style="list-style-type: none">■ For multicast sessions, new messages are discarded according the policy specified by the OverrunPolicy 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. <p>For multicast sessions, when a connection is stopped, messages will continue to be delivered, but only until the MessagesMaximum value is reached. Once this value is reached, messages will be discarded based on the Overrun policy.</p> <p>Note: For topic subscribers that use the multicast extension, also see the Overrun Policy attribute.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSConnectionFactoryMBean</p> <p><i>Attribute:</i> MessagesMaximum</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 10</p> <p><i>Dynamic:</i> yes</p>

Table 155-1

Attribute Label	Description	Value Constraints
Overrun Policy	<p>The policy to use when the number of outstanding multicast messages reaches the value specified in MessagesMaximum and some messages must be discarded.</p> <ul style="list-style-type: none">■ If set to Keep New, the most recent messages are given priority over the oldest messages, and the oldest messages are discarded, as needed.■ If set to Keep Old, the oldest messages are given priority over the most recent messages, and the most recent messages are discarded, as needed. <p>Message age is defined by the order of receipt, not by the JMSTimestamp value.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSConnectionFactoryMBean</p> <p><i>Attribute:</i> OverrunPolicy</p>	<p><i>Default:</i> KeepOld</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ KeepOld■ KeepNew <p><i>Dynamic:</i> yes</p>

Table 155-1

Attribute Label	Description	Value Constraints
Allow Close In On Message	<p>Specifies whether the connection factory creates message consumers that allow a <code>close()</code> or <code>stop()</code> method to be issued within its <code>onMessage()</code> method call.</p> <ul style="list-style-type: none"> ■ If selected (true), a <code>close()</code> or <code>stop()</code> method call from within an <code>onMessage()</code> method call will succeed instead of blocking forever. If the acknowledge mode of the session is set to <code>AUTO_ACKNOWLEDGE</code>, the current message will still be acknowledged automatically when the <code>onMessage()</code> call completes. ■ If not selected (false), it will cause the <code>stop()</code> and <code>close()</code> methods to hang if called from <code>onMessage()</code>. <p>This attribute is dynamic and can be changed at any time. However, changing the value does not affect existing connections. It only affects new connections made with this connection factory.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSConnectionFactoryMBean</code></p> <p><i>Attribute:</i> <code>AllowCloseInOnMessage</code></p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ true ■ false <p><i>Dynamic:</i> yes</p>

Table 155-1

Attribute Label	Description	Value Constraints
Acknowledge Policy	<p>The acknowledge policy for non-transacted sessions that use the CLIENT_ACKNOWLEDGE mode.</p> <p>The All policy specifies that calling acknowledge on a message acknowledges all unacknowledged messages received on the session.</p> <p>The Previous policy specifies that calling acknowledge on a message acknowledges only unacknowledged messages up to, and including, the given message.</p> <p>Note: This attribute works around a change in the JMS specification. Specifically, the specification allowed users to acknowledge all messages before and including the message geing acknowledged. The specification was changed so that acknowledging any message acknowledges all messages ever received (even those received after the message being acknowledge), as follows:</p> <p><i>MBean:</i> weblogic.management.configuration.JMSConnectionFactoryMBean</p> <p><i>Attribute:</i> AcknowledgePolicy</p>	<p><i>Default:</i> All</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ All■ Previous <p><i>Dynamic:</i> yes</p>

Table 155-1

Attribute Label	Description	Value Constraints
Load Balancing Enabled	<p>For distributed destinations, specifies whether non-anonymous producers sending to a distributed queue or topic are load balanced on a per-send basis.</p> <ul style="list-style-type: none">■ If true, the associated message producers will be load balanced on every <code>send()</code> or <code>publish()</code>.■ If false, the associated message producers will be load balanced on the first <code>send()</code> or <code>publish()</code>. <p><i>MBean:</i> <code>weblogic.management.configuration.JMSConnectionFactoryMBean</code> <i>Attribute:</i> <code>LoadBalancingEnabled</code></p>	<p><i>Default:</i> true</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false <p><i>Dynamic:</i> yes</p>
Server Affinity Enabled	<p>For distributed destinations, specifies whether a WebLogic Server that is load balancing consumers or producers across multiple physical destinations (queues or topics) in a distributed destination, will first attempt to load balance across any other physical destinations that are also running on the same WebLogic Server.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSConnectionFactoryMBean</code> <i>Attribute:</i> <code>ServerAffinityEnabled</code></p>	<p><i>Default:</i> true</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false <p><i>Dynamic:</i> yes</p>



JMS Connection Factory --> Configuration --> Transactions

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines transaction configuration attributes for a JMS connection factory. After defining a JMS server, you can configure one or more connection factories to create connections with predefined attributes.

Tasks

[“Configuring a JMS Connection Factory” on page 232-10](#)

[“Deploying a Connection Factory on Multiple Individual Servers” on page 232-12](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Using Transactions WebLogic JMS”](#) in *Programming WebLogic JMS*

Attributes

Table 156-1

Attribute Label	Description	Value Constraints
Transaction Timeout	<p>The timeout value (in seconds) for all transactions on transacted sessions created with this connection factory.</p> <p>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.</p> <p>Note: This setting has no effect on the transaction-timeout for JTA user transactions.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSConnectionFactoryMBean</code></p> <p><i>Attribute:</i> <code>TransactionTimeout</code></p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 3600</p> <p><i>Dynamic:</i> yes</p>

Table 156-1

Attribute Label	Description	Value Constraints
XA Connection Factory Enabled	<p>Indicates whether a XA queue or XA topic connection factory is returned, instead of a queue or topic connection factory.</p> <p>An XA factory is required for JMS applications to use JTA user-transactions, but is not required for transacted sessions. All connections created from an XA factory, whether they are XAConnections or plain Connections, become JTA user-transaction-aware.</p> <p>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.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSConnectionFactoryMBean</code></p> <p><i>Attribute:</i> <code>XAConnectionFactoryEnabled</code></p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false



JMS Connection Factory --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to supply optional information about this JMS connection factory configuration.

Tasks

[“Configuring a JMS Connection Factory” on page 232-10](#)

[“Deploying a Connection Factory on Multiple Individual Servers” on page 232-12](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 157-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.JMSConnectionFactoryMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

JMS Connection Factory --> Target and Deploy

[Tasks](#) [Related Topics](#)

Overview

This tab enables you to target a cluster on which to deploy a connection factory, in order to support cluster-wide, transparent access to JMS destinations from any server in the cluster. A connection factory can be deployed on multiple WebLogic Server instances simultaneously.

Tasks

[“Configuring a JMS Connection Factory” on page 232-10](#)

[“Deploying a Connection Factory on Multiple Individual Servers” on page 232-12](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*



JMS Connection Factory

Connection factories are objects that enable JMS clients to create JMS connections. A connection factory supports concurrent use, enabling multiple threads to access the object simultaneously. After defining a JMS server, you can configure one or more connection factories to create connections with predefined attributes.

To create a connection factory, click the [Configure a new JMS Connection Factory](#) link.

- For more information on creating a JMS connection factory, see [“Configuring a JMS Connection Factory” on page 232-10](#).
- For more information on targeting and deploying a JMS connection factory, see [“Deploying a Connection Factory on Multiple Individual Servers” on page 232-12](#).



Active JMS Connections

This page provides statistics on the active JMS connections (an open communication channel to the messaging system) on your server and allows you to sort the objects by the following criteria: Client ID, active sessions, most sessions, and total sessions.

Click the link under the Sessions column for each active JMS connection to monitor session information for the JMS connection. To modify the layout and/or change the sort order of the Active JMS Connections table, click the Customize this view link.

- For more information monitoring all active JMS servers, see [“Active JMS Servers” on page 197-1](#).
- For more information on monitoring sessions for an active JMS connection, see [“Active JMS Sessions” on page 203-1](#).

Active JMS Connections Information

Client ID — The client ID for the connection.

Sessions — Current number of JMS sessions for the JMS connection. Clicking the Session number takes you to the Active JMS Sessions page, as described in [“Active JMS Sessions” on page 203-1](#).

Most Sessions — Greatest number of JMS sessions for the JMS connection since the last time the server was booted.

Total Sessions — Total number of JMS sessions for the JMS connection since the last time the server was booted.



Active JMS Consumers

Use this page to monitor information about a JMS consumer.

- For more information, see “Active JMS Sessions” on page 203-1.
- For more information, see “[Fundamentals of WebLogic JMS](#)” in *Programming WebLogic JMS*.

Active JMS Consumers Information

Name — The name of the destination for this consumer.

Active — Indicates whether the consumer active and whether it has a message listener set up or a synchronous receive in progress.

Selected — The message selector associated with this consumer, if any.

Durable — Indicates whether the consumer is durable.

Messages Pending — Current number of messages pending by the consumer as the result of an uncommitted transaction or an acknowledgement.

Messages Received — Total number of messages received by the consumer since the last time the server was booted.

Bytes Pending — Current number of bytes pending by the consumer as the result of an uncommitted transaction or an acknowledgement.

Bytes Received — Total number of bytes received by the consumer since the last time the server was booted.



JMS Destination Key --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines general configuration attributes for a JMS destination key (queue or topic). Use destination keys to define the sort order for messages arriving on a specific destination.

Tasks

[“Creating a JMS Destination Key” on page 232-21](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 161-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this JMS destination key configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSDestinationKeyMBean</code></p> <p><i>Attribute:</i> <code>Name</code></p>	
Sort Key	<p>The message sort key or the name of a message header field on which to sort. Message header field keys start with the letters JMS and ignore the key type setting.</p> <p>Note: For better performance, use message header fields as sorting keys, rather than message sort keys.</p> <p>This attribute is not dynamically configurable.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSDestinationKeyMBean</code></p> <p><i>Attribute:</i> <code>Property</code></p>	<p><i>Valid values:</i></p> <ul style="list-style-type: none">■ <code>JMSMessageID</code>■ <code>JMSTimestamp</code>■ <code>JMSCorrelationID</code>■ <code>JMSPriority</code>■ <code>JMSExpiration</code>■ <code>JMSType</code>■ <code>JMSRedelivered</code>■ <code>JMSDeliveryTime</code>

Table 161-1

Attribute Label	Description	Value Constraints
Key Type	<p>The expected property type for this destination key.</p> <p>This setting is ignored for message header field keys, which have an implied type</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSDestinationKeyMBean</code></p> <p><i>Attribute:</i> <code>KeyType</code></p>	<p><i>Default:</i> String</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ Boolean■ Byte■ Short■ Int■ Long■ Float■ Double■ String
Direction	<p>The direction (Ascending or Descending) in which this key will sort messages.</p> <p>Selecting the <i>Ascending</i> option for the <code>JMSMessageID</code> property implies a FIFO (first in, first out) sort order (the default for destinations). Select the <i>Descending</i> option for a LIFO (last in, first out) sort order.</p> <p>This attribute is not dynamically configurable.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSDestinationKeyMBean</code></p> <p><i>Attribute:</i> <code>Direction</code></p>	<p><i>Default:</i> Ascending</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ Ascending■ Descending



JMS Destination Key --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to supply optional information about this JMS destination key configuration.

Tasks

[“Creating a JMS Destination Key” on page 232-21](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 162-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.JMSDestinationKeyMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

JMS Destination Key

You can create destination keys to define the sort order for messages arriving on a specific JMS destination (queue or topic).

To create a destination key, click the Configure a new JMS Destination Key link.

- For more information on creating a JMS destination key, see [“Creating a JMS Destination Key” on page 232-21](#).



Active JMS Destinations

This table reports the active destinations (queues and topics) currently running on a JMS server. Monitoring information for a JMS server's destinations is available only after you have targeted the JMS server to either a WebLogic Server instance or a migratable target on the JMS --> Servers --> Target and Deploy tab.

To monitor the active durable subscribers for topics running on the JMS server, click the Monitor all Durable Subscribers link.

- For more information on monitoring durable subscribers, see [“Monitoring Durable Subscribers for Topics” on page 234-5](#).
- For more information on durable subscribers, see [“Durable Subscribers” on page 180-1](#).

Active JMS Destination Information

Name — The name of the JMS destination.

Consumers — Current number of registered message consumers accessing the destination.

Note: For a JMS topic that has a message-driven bean (MDB) listening on it, the number of consumers will always equal *one*. This is because each message to the topic is either processed by a single MDB or by multiple MDB instances in the pool that share the same consumer, which allows multiple messages to be processed in parallel.

Consumers High — Greatest number of registered message consumers accessing the destination since the last time the server was booted.

Consumers Total — Total number of registered message consumers accessing the destination since the last time the server was booted.

Bytes — Current number of bytes stored in the destination. This does not include pending bytes.

Bytes Pending — Number of bytes stored in the destination that are pending as the result of an uncommitted transaction or an acknowledgement. Pending bytes are over and above the current number of bytes.

Bytes Received — Number of bytes received by the destination since the last time the server was booted.

Bytes Threshold Time — Amount of time that the destination has spent in a bytes threshold condition since the last time the server was booted.

Messages — Current number of messages stored in the destination. This does not include pending bytes

Messages High — Greatest number of messages stored in the destination since the last time the server was booted.

Messages Pending — Number of messages stored in the destination that are pending as the result of an uncommitted transaction or an acknowledgement. Pending messages are over and above the current number of messages.

Messages Received — Number of messages received by the destination since the last time the server was booted.

Messages Threshold Time — Amount of time that the destination spent in a message threshold condition since the last time the server was booted.

JMS Destination

A JMS destination identifies a queue (Point-To-Point) or a topic (Pub/Sub) for a JMS server. After defining a JMS server, you can configure its destinations. You can configure one or more destinations for each JMS server.

To configure a topic, click the Configure a new JMS Topic link. To configure a queue, click the Configure a new JMS Queue link.

- For more information on configuring a JMS topic, see [“Creating a JMS Topic” on page 232-16](#).
- For more information on configuring a JMS queue, see [“Creating a JMS Queue” on page 232-14](#).



JMS Distributed Queue --> Auto Deploy

[Tasks](#) [Related Topics](#)

Overview

This dialog allows you to assign a distributed queue to either a server cluster or to individual server instances that are not in a cluster. You can then select the available JMS server instances that will host the distributed queue members.

Tasks

[“Creating a Distributed Queue and Creating Members Automatically”](#) on page 232-37

[“Creating a Distributed Queue and Adding Existing Physical Queues as Members Manually”](#) on page 232-40

Related Topics

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*



JMS Distributed Queue --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines general configuration attributes for a set of JMS distributed queues.

Tasks

[“Creating a Distributed Queue and Creating Members Automatically” on page 232-37](#)

[“Creating a Distributed Queue and Adding Existing Physical Queues as Members Manually” on page 232-40](#)

Related Topics

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 167-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this distributed queue configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSDistributedQueueMBean</p> <p><i>Attribute:</i> Name</p>	
JNDI Name	<p>The JNDI name used to look up the distributed queue within the JNDI namespace. Applications can use the JNDI name to look up the distributed queue. If not specified, then the distributed queue is not bound into the JNDI namespace.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSDistributedQueueMBean</p> <p><i>Attribute:</i> JNDIName</p>	

Table 167-1

Attribute Label	Description	Value Constraints
Load Balancing Policy	<p>Determines whether non-anonymous producers created through a connection factory are load balanced within this distributed queue on a per-call basis. Valid values are:</p> <ul style="list-style-type: none"> ■ Round-Robin - The system maintains an ordering of physical queue members within the set by distributing the messaging load across the queue members one at a time in the order that they are defined in the configuration file. Each WebLogic Server maintains an identical ordering, but may be at a different point within the ordering. If weights are assigned to any of the queue members in the set, then those members appear multiple times in the ordering. ■ Random - The weight assigned to the queue members is used to compute a weighted distribution for the members of the set. The messaging load is distributed across the queue members by pseudo-randomly accessing the distribution. In the short run, the load will not be directly proportional to the weight. In the long run, the distribution will approach the limit of the distribution. A pure random distribution can be achieved by setting all the weights to the same value, which is typically set to 1. <p><i>MBean:</i> weblogic.management.configuration.JMSDistributedQueueMBean</p> <p><i>Attribute:</i> LoadBalancingPolicy</p>	<p><i>Default:</i> Round-Robin</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ Round-Robin ■ Random <p><i>Dynamic:</i> yes</p>

Table 167-1

Attribute Label	Description	Value Constraints
Forward Delay (seconds)	<p>The amount of time, in seconds, that a distributed queue member with messages, but which has no consumers, will wait before forwarding its messages to other queue members that do have consumers.</p> <p>A value of -1 indicates that no messages are forwarded to other queue members.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSDistributedQueueMBean</code></p> <p><i>Attribute:</i> <code>ForwardDelay</code></p>	<i>Default:</i> -1

Distributed Queue --> Configuration --> Members

You can configure multiple physical JMS queues as members of a single distributed queue set. WebLogic JMS distributes the messaging load across all available queue members within the distributed queue. When a queue member becomes unavailable, traffic is then redirected toward other available queue members in the set.

To create a new distributed queue member, click the Configure a new Distributed Queue Member link.

- For more information on creating a new distributed queue member, see [“Creating a JMS Distributed Queue Member” on page 232-42](#)



JMS Distributed Queue --> Configuration --> Thresholds and Quotas

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab allows you to define the upper and lower bytes/ messages thresholds, maximum bytes/ messages quotas, enable bytes/messages paging to disk, and specify the maximum message size for the members of a JMS distributed queue.

Tasks

[“Creating a Distributed Queue and Creating Members Automatically” on page 232-37](#)

[“Creating a Distributed Queue and Adding Existing Physical Queues as Members Manually” on page 232-40](#)

Related Topics

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 169-1

Attribute Label	Description	Value Constraints
Bytes Maximum	<p>The maximum bytes quota (total amount of bytes) that can be stored on distributed queue members. The default value of -1 specifies that there is no WebLogic-imposed limit on the number of bytes that can be stored in the distributed queue. However, excessive bytes volume can cause memory saturation, so this value should correspond to the total amount of available system memory relative to the rest of your application load.</p> <p>Range of Values: >= BytesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> If a JMS template is used for configuring distributed queue members, then this setting applies only to those specific queue members and not the distributed queue as a whole.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> BytesMaximum</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 169-1

Attribute Label	Description	Value Constraints
Bytes Threshold High	<p>The upper threshold value that triggers events based on the number of bytes stored on distributed queue members. If the number of bytes exceeds this threshold, the triggered events are:</p> <ul style="list-style-type: none"> ■ Log Messages - A message is logged on the server indicating a high threshold condition. ■ Bytes Paging - If bytes paging is enabled (and a paging store has been configured), then destination-level bytes paging is started. ■ Flow Control - If flow control is enabled, the distributed queue member becomes armed and instructs producers to begin decreasing their message flow. <p>A value of -1 specifies that bytes paging, flow control, and threshold log messages are disabled for the distributed queue members. However, if a JMS template is specified for distributed queue members, then -1 implies that the value will come from the template.</p> <p>Range of Values: <= BytesMaximum; >BytesThresholdLow</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> Bytes paging cannot be dynamically disabled by resetting the value to -1. To dynamically disable paging, you could set the value to a very large number, so that paging would not be triggered.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> BytesThresholdHigh</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 169-1

Attribute Label	Description	Value Constraints
Bytes Threshold Low	<p>The lower threshold value that triggers events based on the number of bytes stored on distributed queue members. If the number of bytes falls below this threshold, the triggered events are:</p> <ul style="list-style-type: none">■ Log Messages - A message is logged on the server indicating that the threshold condition has cleared.■ Bytes Paging - If bytes paging is enabled, paging is stopped (if paging is occurring).■ Flow Control - If flow control is enabled, the distributed queue member becomes disarmed and instructs producers to begin increasing their message flow. <p>A value of -1 specifies that bytes paging, flow control, and threshold log messages are disabled for the distributed queue members. However, if a JMS template is specified for distributed queue members, then -1 implies that the value will come from the template.</p> <p>Range of Values: < BytesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> BytesThresholdLow</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 169-1

Attribute Label	Description	Value Constraints
Bytes Paging Enabled	<p>Specifies whether bytes paging is enabled on the distributed queue for temporarily swapping message bodies out from memory when a queue member's bytes load reaches a specified threshold.</p> <ul style="list-style-type: none"> ■ If the check box is cleared (false), bytes paging is disabled for the distributed queue. ■ If the check box is selected (true), and if a paging store has been configured for the JMS Server, and both the BytesThresholdLow and BytesThresholdHigh attribute values are greater than -1, then bytes paging is enabled for the distributed queue. <p>Note: If no value is defined, then this setting defaults to false and bytes paging is disabled for the template's destinations -- unless the destination setting overrides the template.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTemplateMBean</code></p> <p><i>Attribute:</i> <code>BytesPagingEnabled</code></p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ true ■ false

Table 169-1

Attribute Label	Description	Value Constraints
Messages Maximum	<p>The maximum message quota (total amount of messages) that can be stored on distributed queue members. The default value of -1 specifies that there is no WebLogic-imposed limit on the number of messages that can be stored in a distributed queue member. However, excessive message volume can cause memory saturation, so this value should correspond to the total amount of available system memory relative to the rest of your application load.</p> <p>Range of Values: >= MessagesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> If a JMS template is used for distributed queue members, then this setting applies only to those specific members and not the distributed queue as a whole.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> MessagesMaximum</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 169-1

Attribute Label	Description	Value Constraints
Messages Threshold High	<p>The upper threshold value that triggers events based on the number of messages stored on distributed queue members. If the number of messages exceeds this threshold, the triggered events are:</p> <ul style="list-style-type: none">■ Log Messages - A message is logged on the server indicating a high threshold condition.■ Messages Paging - If messages paging is enabled (and a paging store has been configured), then destination-level messages paging is started.■ Flow Control - If flow control is enabled, the queue member becomes armed and instructs producers to begin decreasing their message flow. <p>A value of -1 specifies that messages paging, flow control, and threshold log messages are disabled for the distributed queue member. However, if a JMS template is specified for distributed queue members, then -1 implies that the value will come from the template.</p> <p>Range of Values: <= MessagesMaximum; >MessagesThresholdLow</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> Messages paging cannot be dynamically disabled by resetting the value to -1. To dynamically disable paging, you could set the value to a very large number, so that paging would not be triggered.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> MessagesThresholdHigh</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 169-1

Attribute Label	Description	Value Constraints
Messages Threshold Low	<p>The lower threshold value that triggers events based on the number of messages stored on distributed queue members. If the number of messages falls below this threshold, the triggered events are:</p> <ul style="list-style-type: none">■ Log Messages - A message is logged on the server indicating that the threshold condition has cleared.■ Messages Paging - If messages paging is enabled, paging is stopped (if paging is occurring).■ Flow Control - If flow control is enabled, the distributed queue member becomes disarmed and instructs producers to begin increasing their message flow. <p>A value of -1 specifies that messages paging, flow control, and threshold log messages are disabled for the distributed queue member. However, if a JMS template is specified for distributed queue members, then -1 implies that the value will come from the template.</p> <p>Range of Values: < MessagesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> MessagesThresholdLow</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 169-1

Attribute Label	Description	Value Constraints
Maximum Message Size	<p>The maximum number of bytes allowed for messages on distributed queue members. The size includes the message body, any user-defined properties, and the user-defined JMS header fields: <code>JMSCorrelationID</code> and <code>JMSType</code>.</p> <p>Producers sending messages that exceed the configured maximum message size for the distributed queue receive a <code>ResourceAllocationException</code>.</p> <p>The maximum message size is only enforced for the initial production of a message. Messages that are redirected to an error destination or forwarded to a member of a distributed destination are not checked for size. For instance, if a destination and its corresponding error destination are configured with a maximum message size of 128K bytes and 64K bytes, respectively, a message of 96K bytes could be redirected to the error destination (even though it exceeds the 64K byte maximum), but a producer could not directly send the 96K byte message to the error destination.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTemplateMBean</code></p> <p><i>Attribute:</i> <code>MaximumMessageSize</code></p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 2147483647</p> <p><i>Dynamic:</i> yes</p>

Table 169-1

Attribute Label	Description	Value Constraints
Messages Paging Enabled	<p>Specifies whether messages paging is enabled on the distributed queue for temporarily swapping message bodies out from memory when a queue member's message load reaches a specified threshold.</p> <ul style="list-style-type: none">■ If the check box is cleared (false), messages paging is disabled for the distributed queue members.■ If the check box is selected (true), and if a paging store has been configured for the JMS Server, and both the MessagesThresholdLow and MessagesThresholdHigh attribute values are greater than -1, then messages paging is enabled for the distributed queue members. <p>Note: If no value is defined, this setting defaults to false and messages paging is disabled for the template's destinations -- unless the destination setting overrides the template.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTemplateMBean</code></p> <p><i>Attribute:</i> <code>MessagesPagingEnabled</code></p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false

JMS Distributed Queue --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to supply optional information about this distributed queue configuration.

Tasks

[“Creating a Distributed Queue and Creating Members Automatically” on page 232-37](#)

[“Creating a Distributed Queue and Adding Existing Physical Queues as Members Manually” on page 232-40](#)

Related Topics

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 170-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.JMSDistributedQueueMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

JMS Distributed Queue Member --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines configuration attributes for distributed queue members. Distributed queue members can be added and removed dynamically at run time.

Tasks

[“Creating a JMS Distributed Queue Member” on page 232-42](#)

[“Deleting a JMS Distributed Queue Member” on page 232-43](#)

Related Topics

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 171-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSDistributedQueueMemberMBean</p> <p><i>Attribute:</i> Name</p>	
JMSQueue	<p><i>MBean:</i> weblogic.management.configuration.JMSDistributedQueueMemberMBean</p> <p><i>Attribute:</i> JMSQueue</p>	
Weight	<p>The weight of a distributed destination member is a measure of its ability to handle message load, with respect to the other member destinations in the same distributed set.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSDistributedQueueMemberMBean</p> <p><i>Attribute:</i> Weight</p>	<p><i>Default:</i> 1</p> <p><i>Dynamic:</i> yes</p>

JMS Distributed Queue

You can configure multiple physical JMS queues as members of a single distributed queue set that can be served by multiple WebLogic Server instances within a cluster. Once configured, your producers and consumers are able to send and receive to the distributed queue. WebLogic JMS then distributes the messaging load across all available queue members within the distributed queue. When a queue member becomes unavailable, traffic is then redirected toward other available queue members in the set.

To create a new distributed queue, click the [Configure a new Distributed Queue](#) link.

For more information on creating a new distributed queue, see:

- [“Creating a Distributed Queue and Creating Members Automatically”](#) on page 232-37.
- [“Creating a Distributed Queue and Adding Existing Physical Queues as Members Manually”](#) on page 232-40



JMS Distributed Topic --> Auto Deploy

[Tasks](#) [Related Topics](#)

Overview

This dialog allows you to assign a distributed topic to either a server cluster or to individual server instances that are not in a cluster. You can then select the available JMS server instances that will host the distributed topic members.

Tasks

“Creating a Distributed Topic and Creating Members Automatically” on page 232-32

“Creating a Distributed Topic and Adding Existing Physical Topics as Members Manually” on page 232-35

Related Topics

“[Developing a WebLogic JMS Application](#)” in *Programming WebLogic JMS*



JMS Distributed Topic --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines general configuration attributes for a set of JMS distributed topics.

Tasks

“Creating a Distributed Topic and Creating Members Automatically” on page 232-32

“Creating a Distributed Topic and Adding Existing Physical Topics as Members Manually” on page 232-35

Related Topics

“[Developing a WebLogic JMS Application](#)” in *Programming WebLogic JMS*

Attributes

Table 174-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSDistributedTopicMBean</p> <p><i>Attribute:</i> Name</p>	
JNDI Name	<p>The JNDI name used to look up the distributed destination within the JNDI namespace. Applications can use the JNDI name to look up the distributed destination. If not specified, then the destination is not bound into the JNDI namespace.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSDistributedTopicMBean</p> <p><i>Attribute:</i> JNDIName</p>	

Table 174-1

Attribute Label	Description	Value Constraints
Load Balancing Policy	<p>The load balancing policy for producers sending messages to a distributed destination in order to balance the message load across the members of the distributed set.</p> <ul style="list-style-type: none"> ■ Round-Robin - The system maintains an ordering of physical topic members within the set by distributing the messaging load across the topic members one at a time in the order that they are defined in the configuration file. Each WebLogic Server maintains an identical ordering, but may be at a different point within the ordering. If weights are assigned to any of the topic members in the set, then those members appear multiple times in the ordering. ■ Random - The weight assigned to the topic members is used to compute a weighted distribution for the members of the set. The messaging load is distributed across the topic members by pseudo-randomly accessing the distribution. In the short run, the load will not be directly proportional to the weight. In the long run, the distribution will approach the limit of the distribution. A pure random distribution can be achieved by setting all the weights to the same value, which is typically set to 1. <p><i>MBean:</i> weblogic.management.configuration.JMSDistributedTopicMBean</p> <p><i>Attribute:</i> LoadBalancingPolicy</p>	<p><i>Default:</i> Round-Robin</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ Round-Robin ■ Random <p><i>Dynamic:</i> yes</p>



Distributed Topic --> Configuration --> Members

You can configure multiple physical JMS topics as members of a single distributed topic set. WebLogic JMS distributes the messaging load across all available topic members within the distributed topic. When a topic member becomes unavailable, traffic is then redirected toward other available topic members in the set.

To create a new distributed topic member, click the Configure a new Distributed Topic Member link.

- For more information on creating a new distributed topic member, see [“Creating a JMS Distributed Topic Member” on page 232-44](#)



JMS Distributed Topic --> Configuration --> Thresholds and Quotas

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab allows you to define the upper and lower bytes/ messages thresholds, maximum bytes/ messages quotas, enable bytes/messages paging to disk, and specify the maximum message size for the members of a JMS distributed topic.

Tasks

“Creating a Distributed Topic and Creating Members Automatically” on page 232-32

“Creating a Distributed Topic and Adding Existing Physical Topics as Members Manually” on page 232-35

Related Topics

“[Developing a WebLogic JMS Application](#)” in *Programming WebLogic JMS*

Attributes

Table 176-1

Attribute Label	Description	Value Constraints
Bytes Maximum	<p>The maximum bytes quota (total amount of bytes) that can be stored on distributed topic members. The default value of -1 specifies that there is no WebLogic-imposed limit on the number of bytes that can be stored in the destination. However, excessive bytes volume can cause memory saturation, so this value should correspond to the total amount of available system memory relative to the rest of your application load.</p> <p>Range of Values: >= BytesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> If a JMS template is used for distributed destination members, then this setting applies only to those specific members and not the distributed destination set as a whole.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> BytesMaximum</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 176-1

Attribute Label	Description	Value Constraints
Bytes Threshold High	<p>The upper threshold value that triggers events based on the number of bytes stored on distributed topic members. If the number of bytes exceeds this threshold, the triggered events are:</p> <ul style="list-style-type: none">■ Log Messages - A message is logged on the server indicating a high threshold condition.■ Bytes Paging - If bytes paging is enabled (and a paging store has been configured), then destination-level bytes paging is started.■ Flow Control - If flow control is enabled, the destination becomes armed and instructs producers to begin decreasing their message flow. <p>A value of -1 specifies that bytes paging, flow control, and threshold log messages are disabled for the destination. However, if a JMS template is specified for distributed queue members, then -1 implies that the value will come from the template.</p> <p>Range of Values: <= BytesMaximum; >BytesThresholdLow</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> Bytes paging cannot be dynamically disabled by resetting the value to -1. To dynamically disable paging, you could set the value to a very large number, so that paging would not be triggered.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> BytesThresholdHigh</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 176-1

Attribute Label	Description	Value Constraints
Bytes Threshold Low	<p>The lower threshold value that triggers events based on the number of bytes stored on distributed topic members. If the number of bytes falls below this threshold, the triggered events are:</p> <ul style="list-style-type: none">■ Log Messages - A message is logged on the server indicating that the threshold condition has cleared.■ Bytes Paging - If bytes paging is enabled, paging is stopped (if paging is occurring).■ Flow Control - If flow control is enabled, the destination becomes disarmed and instructs producers to begin increasing their message flow. <p>A value of -1 specifies that bytes paging, flow control, and threshold log messages are disabled for the destination. However, if a JMS template is specified for distributed topic members, then -1 implies that the value will come from the template.</p> <p>Range of Values: < BytesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> BytesThresholdLow</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 176-1

Attribute Label	Description	Value Constraints
Bytes Paging Enabled	<p>Specifies whether bytes paging is enabled on the distributed topic for temporarily swapping message bodies out from memory when a topic member's message load reaches a specified threshold.</p> <ul style="list-style-type: none">■ If the check box is cleared (false), bytes paging is disabled for the distributed topic.■ If the check box is selected (true), and if a paging store has been configured for the JMS Server, and both the BytesThresholdLow and BytesThresholdHigh attribute values are greater than -1, then bytes paging is enabled for the distributed topic. <p>Note: If no value is defined, then this setting defaults to false and bytes paging is disabled for the template's destinations -- unless the destination setting overrides the template.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> BytesPagingEnabled</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false

Table 176-1

Attribute Label	Description	Value Constraints
Messages Maximum	<p>The maximum message quota (total amount of messages) that can be stored on distributed topic members. The default value of -1 specifies that there is no WebLogic-imposed limit on the number of messages that can be stored in the destination. However, excessive message volume can cause memory saturation, so this value should correspond to the total amount of available system memory relative to the rest of your application load.</p> <p>Range of Values: >= MessagesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> If a JMS template is used for distributed destination members, then this setting applies only to those specific members and not the distributed destination set as a whole.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> MessagesMaximum</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 176-1

Attribute Label	Description	Value Constraints
Messages Threshold High	<p>The upper threshold value that triggers events based on the number of messages stored on distributed topic members. If the number of messages exceeds this threshold, the triggered events are:</p> <ul style="list-style-type: none">■ Log Messages - A message is logged on the server indicating a high threshold condition.■ Messages Paging - If messages paging is enabled (and a paging store has been configured), then destination-level messages paging is started.■ Flow Control - If flow control is enabled, the destination becomes armed and instructs producers to begin decreasing their message flow. <p>A value of -1 specifies that messages paging, flow control, and threshold log messages are disabled for the destination. However, if a JMS template is specified for distributed queue members, then -1 implies that the value will come from the template.</p> <p>Range of Values: <= MessagesMaximum; >MessagesThresholdLow</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> Messages paging cannot be dynamically disabled by resetting the value to -1. To dynamically disable paging, you could set the value to a very large number, so that paging would not be triggered.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> MessagesThresholdHigh</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 176-1

Attribute Label	Description	Value Constraints
Messages Threshold Low	<p>The lower threshold value that triggers events based on the number of messages stored on distributed topic members. If the number of messages falls below this threshold, the triggered events are:</p> <ul style="list-style-type: none">■ Log Messages - A message is logged on the server indicating that the threshold condition has cleared.■ Messages Paging - If messages paging is enabled, paging is stopped (if paging is occurring).■ Flow Control - If flow control is enabled, the destination becomes disarmed and instructs producers to begin increasing their message flow. <p>A value of -1 specifies that messages paging, flow control, and threshold log messages are disabled for the distributed queue member. However, if a JMS template is specified for distributed queue members, then -1 implies that the value will come from the template.</p> <p>Range of Values: < MessagesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> MessagesThresholdLow</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 176-1

Attribute Label	Description	Value Constraints
Maximum Message Size	<p>The maximum number of bytes allowed for messages on distributed topic members. The size includes the message body, any user-defined properties, and the user-defined JMS header fields: <code>JMSCorrelationID</code> and <code>JMSType</code>.</p> <p>Producers sending messages that exceed the configured maximum message size for the distributed queue receive a <code>ResourceAllocationException</code>.</p> <p>The maximum message size is only enforced for the initial production of a message. Messages that are redirected to an error destination or forwarded to a member of a distributed destination are not checked for size. For instance, if a destination and its corresponding error destination are configured with a maximum message size of 128K bytes and 64K bytes, respectively, a message of 96K bytes could be redirected to the error destination (even though it exceeds the 64K byte maximum), but a producer could not directly send the 96K byte message to the error destination.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTemplateMBean</code></p> <p><i>Attribute:</i> <code>MaximumMessageSize</code></p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 2147483647</p> <p><i>Dynamic:</i> yes</p>

Table 176-1

Attribute Label	Description	Value Constraints
Messages Paging Enabled	<p>Specifies whether messages paging is enabled on the distributed topic for temporarily swapping message bodies out from memory when a topic member's messages load reaches a specified threshold.</p> <ul style="list-style-type: none">■ If the check box is cleared (false), messages paging is disabled for the distributed topic members.■ If the check box is selected (true), and if a paging store has been configured for the JMS Server, and both the MessagesThresholdLow and MessagesThresholdHigh attribute values are greater than -1, then messages paging is enabled for the distributed topic members. <p>Note: If no value is defined, this setting defaults to false and messages paging is disabled for the template's destinations -- unless the destination setting overrides the template.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTemplateMBean</code></p> <p><i>Attribute:</i> <code>MessagesPagingEnabled</code></p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false

JMS Distributed Topic --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to supply optional information about this JMS distributed topic configuration.

Tasks

“Creating a Distributed Topic and Creating Members Automatically” on page 232-32

“Creating a Distributed Topic and Adding Existing Physical Topics as Members Manually” on page 232-35

Related Topics

“[Developing a WebLogic JMS Application](#)” in *Programming WebLogic JMS*

Attributes

Table 177-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.JMSDistributedTopicMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

JMS Distributed Topic Member --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines configuration attributes for distributed topic members. Distributed topic members can be added and removed dynamically at run time.

Tasks

[“Creating a JMS Distributed Topic Member” on page 232-44](#)

[“Deleting a JMS Distributed Topic Member” on page 232-45](#)

Related Topics

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 178-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSDistributedTopicMemberMBean</p> <p><i>Attribute:</i> Name</p>	
Distributed Topic	<p><i>MBean:</i> weblogic.management.configuration.JMSDistributedTopicMemberMBean</p> <p><i>Attribute:</i> DistributedTopic</p>	
Weight	<p>The weight of a distributed destination member is a measure of its ability to handle message load, with respect to the other member destinations in the same distributed set.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSDistributedTopicMemberMBean</p> <p><i>Attribute:</i> Weight</p>	<p><i>Default:</i> 1</p> <p><i>Dynamic:</i> yes</p>

JMS Distributed Topic

You can configure multiple physical JMS topics as members of a single distributed topic set that can be served by multiple WebLogic Server instances within a cluster. Once configured, your producers and consumers are able to send and receive to the distributed topic. WebLogic JMS then distributes the messaging load across all available topic members within the distributed topic. When a topic member becomes unavailable, traffic is then redirected toward other available topic members in the set.

To create a new distributed topic, click the [Configure a new Distributed Topic](#) link.

For more information on creating a new distributed topic, see:

- [“Creating a Distributed Topic and Creating Members Automatically” on page 232-32.](#)
- [“Creating a Distributed Topic and Adding Existing Physical Topics as Members Manually” on page 232-35](#)



Durable Subscribers

This page shows the durable subscribers that are running on your JMS topics. WebLogic JMS stores durable subscribers in a persistent, disk-based file store or JDBC-accessible database until the message has been delivered to the subscribers or has expired, even if those subscribers are not active when the message is delivered.

- For more destination monitoring information, see [“Active JMS Destinations” on page 164-1](#).
- For more information on creating durable subscribers, see [“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*.

Durable Subscriber Information

Client ID — The unique client ID for this durable subscriber.

Subscription Name — The subscription name for this durable subscriber.

No Local — The noLocal flag for this durable subscriber, which when enabled, can prevent an application from receiving messages it has published itself.

Active — Determines whether this subscription is being used by a durable subscriber.

Selector — If specified, the message selector for filtering messages.

Messages Pending Count — Number of messages that are pending by this durable subscriber as the result of an uncommitted transaction or an acknowledgement. Pending messages are over and above the current number of messages.

Messages Current Count — The number of messages still available by this durable subscriber.

Bytes Pending Count — Number of bytes that are pending by this durable subscriber as the result of an uncommitted transaction or an acknowledgement. Pending bytes are over and above the current number of bytes.

Bytes Current Count — The number of bytes received by this durable subscriber.



JMS File Store --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines general configuration attributes for a JMS file store. A JMS file store consists of a disk-based file that is used for storing persistent messages and durable subscribers.

Tasks

[“Creating a JMS File Store” on page 232-23](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Improving JMS File Store Performance” on page 235-2](#)

Attributes

Table 181-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this JMS file store configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSFileStoreMBean</code></p> <p><i>Attribute:</i> <code>Name</code></p>	

Table 181-1

Attribute Label	Description	Value Constraints
Synchronous Write Policy	<p>A user-defined policy that determines how the JMS file store writes data to disk. This policy also affects the JMS file store's performance, scalability, and reliability. The valid policy options are:</p> <p>Disabled - Transactions are complete as soon as their writes are cached in memory, instead of waiting for the writes to successfully reach the disk. This policy is the fastest, but the least reliable (that is, transactionally safe). It can be more than 100 times faster than the other policies, but power outages or operating system failures can cause lost and/or duplicate messages.</p> <p>Cache-Flush - Transactions cannot complete until all of their writes have been flushed down to disk. This policy is reliable and scales well as the number of simultaneous users increases.</p> <p>Direct-Write - File store writes are written directly to disk. This policy is supported on Solaris and Windows. If this policy is set on an unsupported platform, the file store automatically uses the Cache-Flush policy instead.</p> <ul style="list-style-type: none">■ With on-disk caching enabled, the Direct-Write policy can be 2-5 times faster than the Cache-Flush policy, except in highly scalable cases where it may be slightly slower.■ With on-disk caching disabled, the Direct-Write policy is faster than the Cache-Flush policy in one-to-many cases, but is much slower otherwise.■ The Direct-Write policy scales well with on-disk caching enabled, but does not scale with it disabled. (Solaris does not allow enabling the on-disk cache for direct writes).	<p><i>Default:</i> Cache-Flush</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ Disabled■ Cache-Flush■ Direct-Write <p><i>Dynamic:</i> yes</p>

Table 181-1

Attribute Label	Description	Value Constraints
Synchronous Write Policy (cont.)	<p>Warning! Although the use of the Direct-Write policy is transactionally reliable on Solaris systems, Windows systems may leave transaction data in the on-disk cache without writing it to disk immediately. For more information, see “Disabling the On-Disk Cache For a Disk Drive on Windows 2000” on page 235-3.</p> <p>Note: If the JMS file store is used exclusively for paging non-persistent messages to disk, the Synchronous Write Policy is ignored.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSFileStoreMBean</p> <p><i>Attribute:</i> SynchronousWritePolicy</p>	
Directory	<p>The pathname to the valid file-system directory where the JMS file store is kept. This attribute is not dynamically configurable.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSFileStoreMBean</p> <p><i>Attribute:</i> Directory</p>	

JMS File Store --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to supply optional information about this JMS file store configuration.

Tasks

[“Creating a JMS File Store” on page 232-23](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

Attributes

Table 182-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.JMSFileStoreMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes



JMS JDBC Store --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines general configuration attributes for a JMS JDBC store. A JMS JDBC store consists of a JDBC-accessible database that is used for persistent messaging.

Tasks

[“Creating a JMS JDBC Store” on page 232-26](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“JDBC Database Utility”](#) in *Programming WebLogic JMS*

Attributes

Table 183-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this JMS JDBC store configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSJDBCStoreMBean</p> <p><i>Attribute:</i> Name</p>	
Connection Pool	<p>The JDBC Connection Pool used to accesss this JMS JDBC store.</p> <p>This attribute is not dynamically configurable.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSJDBCStoreMBean</p> <p><i>Attribute:</i> ConnectionPool</p>	

Table 183-1

Attribute Label	Description	Value Constraints
Prefix Name	<p>The prefix name that is prepended to the JMS tables in this JMS JDBC store. Specifying a format of <code>[schema.[catalog.]]prefix</code> results in a valid database table name when it is prepended to the JMS table name. Prepend a prefix to the JMS table names when:</p> <ul style="list-style-type: none">■ The RDBMS requires fully-qualified names. (You should verify this with your database administrator.)■ You must differentiate between JMS tables for two WebLogic Server instances, thereby enabling multiple tables to be stored on a single RDBMS. <p><i>MBean:</i> <code>weblogic.management.configuration.JMSJDBCStoreMBean</code></p> <p><i>Attribute:</i> <code>PrefixName</code></p>	<i>Default:</i> null



JMS JDBC Store --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to supply optional information about this JMS JDBC store configuration.

Tasks

[“Creating a JMS JDBC Store” on page 232-26](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“JDBC Database Utility”](#) in *Programming WebLogic JMS*

Attributes

Table 184-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.JMSJDBCStoreMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

JMS Pooled Connections

Overview

A pooled JMS connection is a session pool used by EJBs and servlets that use a *resource-ref* element in their EJB or servlet deployment descriptor to define their JMS connection factories (also known as a *wrapped* connection factory). This tab enables you to monitor JMS connection and session objects, as well some message producer objects that are automatically created when declaring a wrapped JMS connection factory in your EJB or servlet.

For more information on using JMS wrappers with EJBs and servlets to create pooled JMS connection objects, see “[Using JMS With EJBs and Servlets](#)” in *Programming WebLogic JMS*.

Num Leaked — The number of JMS sessions that were removed from the session pool, but which were not returned.

Num Failures To Refresh — The number of failed attempts to create a JMS session in the session pool.

Creation Delay Time — The average amount of time that it takes to create each JMS session in the session pool.

Num Waiters — The number of threads waiting to retrieve a JMS session from the session pool.

Highest Num Waiters — The highest number of threads waiting to retrieve a JMS session in this instance of the session pool since it was instantiated.

Highest Wait Seconds — The highest number of seconds that an application waited to retrieve a JMS session in this instance of the session pool since it was instantiated.

Num Reserved — The number of JMS sessions that are currently in use.

Highest Num Reserved — The highest number of concurrent JMS sessions reserved for this instance of the session pool since it was instantiated.

Num Available — The number of available JMS sessions in the session pool that are not currently in use.

Highest Num Available — The highest number of available JMS sessions in this instance of the session pool since it was instantiated.

Num Unavailable — The number of JMS sessions in the session pool that are not currently usable for some reason.

Highest Num Unavailable — The highest number of unusable JMS sessions in this instance of the session pool since it was instantiated.

Total Num Allocated — The total number of JMS sessions allocated by this session pool in this instance of the session pool since it was instantiated.

Total Num Destroyed — The total number of JMS sessions that were created and then destroyed in this instance of the session pool since it was instantiated.

Max Capacity — The maximum number of JMS sessions that can be allocated using the session pool.

Curr Capacity — The current capacity of the session pool, which is always less than or equal to the maximum capacity of JMS sessions.

Average Reserved — The average number of JMS sessions that have been in use in this instance of the session pool since it was instantiated. This generally happens when an EJB or servlet is deployed that requires the session pool.

Num Connection Objects — The number of JMS connections that back this session pool. This value may be greater than one if different sessions were created using different combinations of a username and password to contact the JMS server.

Active JMS Producers

Use this page to monitor information for a JMS producer.

- For more information, see “Active JMS Sessions” on page 203-1.
- For more information, see “[Fundamentals of WebLogic JMS](#)” in *Programming WebLogic JMS*.

Active JMS Producers Information

Bytes Pending — Current number of bytes pending by the producer that are pending as the result of an uncommitted transaction or an acknowledgement.

Bytes Sent — Total number of bytes sent by the producer since the last time the server was booted.

Messages Pending — Current number of messages pending by the producer that are pending as the result of an uncommitted transaction or an acknowledgement.

Messages Sent — Total number of messages sent by the producer since the last time the server was booted.



JMS Queue --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines general configuration attributes for a queue destination. After defining a JMS server, you can configure its queues. You can configure one or more queues for each JMS server.

Note: You configure queues explicitly or by configuring a JMS template that can be used to define multiple queues with similar attribute settings, as described in [“JMS Template Tasks” on page 232-18](#).

Tasks

[“Creating a JMS Queue” on page 232-14](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 186-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this queue destination configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSQueueMBean</p> <p><i>Attribute:</i> Name</p>	
JNDIName	<p>The JNDI name used to look up this queue destination within the JNDI namespace. If not specified, the destination name is not advertised through the JNDI namespace and cannot be looked up and used.</p> <p>This attribute is not dynamically configurable.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSQueueMBean</p> <p><i>Attribute:</i> JNDIName</p>	
Replicate JNDI Name In Cluster	<p>Specifies whether the JNDI name for this queue destination (if specified) is replicated across the cluster. If this option is not selected, then the JNDI name for the queue (if specified) is only visible on the server hosting this queue.</p> <p>This attribute is not dynamically configurable.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSQueueMBean</p> <p><i>Attribute:</i> JNDINameReplicated</p>	<p><i>Default:</i> true</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ false■ true

Table 186-1

Attribute Label	Description	Value Constraints
Enable Store	<p>Specifies whether this queue destination supports persistent messaging by using the JMS store specified by the JMS server.</p> <ul style="list-style-type: none">■ default - The queue destination uses the JMS store defined for the JMS server--if one is defined--and supports persistent messaging. However, if a JMS store is not defined for the JMS server, then persistent messages are automatically downgraded to non-persistent.■ false - The queue destination does not support persistent messaging.■ true - The queue destination does support persistent messaging. However, if a JMS store is not defined for the JMS server, then the configuration will fail and the JMS server will not boot. <p>This attribute is not dynamically configurable.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSQueueMBean</code></p> <p><i>Attribute:</i> <code>StoreEnabled</code></p>	<p><i>Default:</i> default</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ default■ false■ true

Table 186-1

Attribute Label	Description	Value Constraints
Template	<p>The JMS template from which this queue destination is derived.</p> <p>If a JMS template is specified, the queue destination attributes that are set to their default values will inherit their values from the JMS template at run time. However, if this attribute is not defined, then the attributes for the queue destination must be specified as part of the destination.</p> <p>The Template attribute setting per destination is static. The JMS template's attributes, however, can be modified dynamically.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSQueueMBean</code></p> <p><i>Attribute:</i> <code>Template</code></p>	
Destination Keys	<p>The sort ordering for messages that arrive on this queue destination.</p> <p>The keys are ordered from most significant to least significant. If more than one key is specified, a key based on the <code>JMSMessageID</code> property can only be the last key in the list.</p> <p><i>Note:</i> If <code>JMSMessageID</code> is not defined in the key, it is implicitly assumed to be the last key and is set as "ascending" (first-in, first-out) for the sort order.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSQueueMBean</code></p> <p><i>Attribute:</i> <code>DestinationKey</code></p>	

JMS Queue --> Configuration --> Overrides

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines the override configuration attributes for a queue destination. After defining a JMS server, you can configure its queues. You can configure one or more queues for each JMS server.

Note: You configure queues explicitly or by configuring a JMS template that can be used to define multiple queues with similar attribute settings, as described in [“JMS Template Tasks” on page 232-18](#).

Tasks

[“Creating a JMS Queue” on page 232-14](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 187-1

Attribute Label	Description	Value Constraints
Priority Override	<p>The priority assigned to all messages that arrive at the destination, regardless of the Priority specified by the message producer.</p> <p>If no JMS template is specified for this destination, then -1 means that the destination will not override the Priority setting. Otherwise, -1 means that the value comes from the template.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSQueueMBean</code></p> <p><i>Attribute:</i> <code>PriorityOverride</code></p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>
Time To Live Override	<p>The time-to-live value assigned to all messages that arrive at the destination, regardless of the TimeToLive specified by the message producer.</p> <p>If no JMS template is specified for this destination, then -1 means that the destination will not override the TimeToLive setting. Otherwise, -1 means that the value comes from the template.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSQueueMBean</code></p> <p><i>Attribute:</i> <code>TimeToLiveOverride</code></p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 187-1

Attribute Label	Description	Value Constraints
Time To Deliver Override	<p>The default delay, either in milliseconds or as a schedule, between when a message is produced and when it is made visible on its target destination, regardless of the delivery time specified by the producer and/or connection factory.</p> <p>If no JMS template is specified for this destination, then -1 means that the destination will not override the TimeToDeliver setting. Otherwise, -1 means that the value comes from the template.</p> <p>The Time To Deliver Override can be specified either as a long or as an advanced scheduling syntax (see "Setting Message Delivery Times" in <i>Programming JMS</i> for details.)</p> <p><i>Note:</i> Changing the Time To Deliver Override only affects future message delivery, it does not affect message delivery of already produced messages.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSQueueMBean</p> <p><i>Attribute:</i> TimeToDeliverOverride</p>	<p><i>Default:</i> "-1"</p> <p><i>Dynamic:</i> yes</p>

Table 187-1

Attribute Label	Description	Value Constraints
Delivery Mode Override	<p>The delivery mode assigned to all messages that arrive at the destination regardless of the DeliveryMode specified by the message producer.</p> <p>If no JMS template is specified for this destination, then No-Delivery means the DeliveryMode will not be overridden. Otherwise, No-Delivery means that the value comes from the template.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSQueueMBean</p> <p><i>Attribute:</i> DeliveryModeOverride</p>	<p><i>Default:</i> No-Delivery</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ Persistent■ Non-Persistent■ No-Delivery <p><i>Dynamic:</i> yes</p>

JMS Queue --> Configuration --> Redelivery

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines the message redelivery configuration attributes for a JMS queue. After defining a JMS server, you can configure its queues. You can configure one or more queues for each JMS server.

Note: You configure queues explicitly or by configuring a JMS template that can be used to define multiple queues with similar attribute settings, as described in [“JMS Template Tasks” on page 232-18](#).

Tasks

[“Creating a JMS Queue” on page 232-14](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 188-1

Attribute Label	Description	Value Constraints
Redelivery Delay Override	<p>The delay, in milliseconds, before rolled back or recovered messages are redelivered, regardless of the RedeliveryDelay specified by the consumer and/or connection factory. Redelivered queue messages are put back into their originating destination; redelivered topic messages are put back into their originating subscription.</p> <p>The default value (-1) specifies that the destination will not override the RedeliveryDelay setting.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> Changing the Redelivery Delay Override only affects future rollbacks and recovers, it does not affect rollbacks and recovers that have already occurred.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSQueueMBean</p> <p><i>Attribute:</i> RedeliveryDelayOverride</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 188-1

Attribute Label	Description	Value Constraints
Redelivery Limit	<p>The number of redelivery tries a message can have before it is moved to the Error Destination specified on this page. This setting overrides any redelivery limit set by the message sender. If the redelivery limit is configured, but no error destination is configured, then persistent and non-persistent messages are simply dropped (deleted) when they reach their redelivery limit.</p> <p>The default value (-1) specifies that the destination will not override the message sender's redelivery limit setting.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; previously sent messages continue to use their original redelivery limit.</p> <p><i>Note:</i> The number of times a message has been redelivered is not persisted. This means that after a restart, the number of delivery attempts on each message is reset to zero.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSQueueMBean</code></p> <p><i>Attribute:</i> <code>RedeliveryLimit</code></p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 188-1

Attribute Label	Description	Value Constraints
Error Destination	<p>The name of the target destination (queue or topic) for messages that have reached their redelivery limit. If no error destination is configured on the local JMS server, then such messages are simply dropped. If a message has expired and the Expiration Policy is set to Redirect, then the message is moved to the specified error destination.</p> <p>If this destination has a template, the none option indicates that the error destination comes from the template. If this destination has no template, none indicates that there is no error destination configured.</p> <p><i>Note:</i> Configured destination quotas do not apply to expired or redelivery limit messages that are redirected to an error destination. Such messages are still moved into an error destination even if that destination has reached its quotas.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSQueueMBean</p> <p><i>Attribute:</i> ErrorDestination</p>	<p><i>Minimum:</i> -1</p> <p><i>Default:</i> null</p> <p><i>Dynamic:</i> yes</p>

JMS Queue --> Configuration --> Thresholds & Quotas

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines the bytes/messages threshold and quota configuration attributes for a JMS queue. After defining a JMS server, you can configure its queues. You can configure one or more queues for each JMS server.

Note: You configure queues explicitly or by configuring a JMS template that can be used to define multiple queues with similar attribute settings, as described in [“JMS Template Tasks” on page 232-18](#).

Tasks

[“Creating a JMS Queue” on page 232-14](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 189-1

Attribute Label	Description	Value Constraints
Bytes Maximum	<p>The maximum bytes quota (total amount of bytes) that can be stored in this destination. The default value of -1 specifies that there is no WebLogic-imposed limit on the number of bytes that can be stored in the destination. However, excessive bytes volume can cause memory saturation, so this value should correspond to the total amount of available system memory relative to the rest of your application load.</p> <p><i>Note:</i> Configured quotas do not apply to expired or error (redelivery limit) messages that are redirected to an error destination. Such messages are still moved into an error destination even if that destination has reached its quotas.</p> <p>Range of Values: >= BytesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> If a JMS template is used for distributed destination members, then this setting applies only to those specific members and not the distributed destination set as a whole.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSQueueMBean</code></p> <p><i>Attribute:</i> BytesMaximum</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 189-1

Attribute Label	Description	Value Constraints
Bytes Threshold High	<p>The upper threshold value that triggers events based on the number of bytes stored in the destination. If the number of bytes exceeds this threshold, the triggered events are:</p> <ul style="list-style-type: none"> ■ Log Messages - A message is logged on the server indicating a high threshold condition. ■ Bytes Paging - If bytes paging is enabled (and a paging store has been configured), then destination-level bytes paging is started. ■ Flow Control - If flow control is enabled, the destination becomes armed and instructs producers to begin decreasing their message flow. <p>A value of -1 specifies that bytes paging, flow control, and threshold log messages are disabled for the destination. However, if a JMS template is specified for this destination, then -1 implies that the value will come from the template.</p> <p>Range of Values: <= BytesMaximum; >BytesThresholdLow</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> Bytes paging cannot be dynamically disabled by resetting the value to -1. To dynamically disable paging, you could set the value to a very large number, so that paging would not be triggered.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSQueueMBean</p> <p><i>Attribute:</i> BytesThresholdHigh</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 189-1

Attribute Label	Description	Value Constraints
Bytes Threshold Low	<p>The lower threshold value that triggers events based on the number of bytes stored in the destination. If the number of bytes falls below this threshold, the triggered events are:</p> <ul style="list-style-type: none">■ Log Messages - A message is logged on the server indicating that the threshold condition has cleared.■ Bytes Paging - If bytes paging is enabled, paging is stopped (if paging is occurring).■ Flow Control - If flow control is enabled, the destination becomes disarmed and instructs producers to begin increasing their message flow. <p>A value of -1 specifies that bytes paging, flow control, and threshold log messages are disabled for the destination. However, if a JMS template is specified for this destination, then -1 implies that the value will come from the template.</p> <p>Range of Values: < BytesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSQueueMBean</p> <p><i>Attribute:</i> BytesThresholdLow</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 189-1

Attribute Label	Description	Value Constraints
Bytes Paging Enabled	<p>Specifies whether bytes paging is enabled on the destination for temporarily swapping message bodies out from memory when the destination's bytes load reaches a specified threshold.</p> <ul style="list-style-type: none">■ default - If a JMS template is specified, then this value inherits the template's Bytes Paging value. If no JMS template is configured for the destination, then the Default value is equivalent to false.■ false - Bytes paging is explicitly disabled for this destination.■ true - If a paging store has been configured for the JMS server, and both the BytesThresholdLow and BytesThresholdHigh attribute values are greater than -1, then bytes paging is enabled for this destination. <p><i>MBean:</i> <code>weblogic.management.configuration.JMSQueueMBean</code></p> <p><i>Attribute:</i> <code>BytesPagingEnabled</code></p>	<p><i>Default:</i> default</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ default■ false■ true

Table 189-1

Attribute Label	Description	Value Constraints
Messages Maximum	<p>The maximum message quota (total amount of messages) that can be stored in this destination. The default value of -1 specifies that there is no WebLogic-imposed limit on the number of messages that can be stored in the destination. However, excessive message volume can cause memory saturation, so this value should correspond to the total amount of available system memory relative to the rest of your application load.</p> <p><i>Note:</i> Configured quotas do not apply to expired or error (redelivery limit) messages that are redirected to an error destination. Such messages are still moved into an error destination even if that destination has reached its quotas.</p> <p>Range of Values: >= MessagesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> If a JMS template is used for distributed destination members, then this setting applies only to those specific members and not the distributed destination set as a whole.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSQueueMBean</p> <p><i>Attribute:</i> MessagesMaximum</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 189-1

Attribute Label	Description	Value Constraints
Messages Threshold High	<p>The upper threshold value that triggers events based on the number of messages stored in the destination. If the number of messages exceeds this threshold, the triggered events are:</p> <ul style="list-style-type: none">■ Log Messages - A message is logged on the server indicating a high threshold condition.■ Messages Paging - If messages paging is enabled (and a paging store has been configured), then destination-level messages paging is started.■ Flow Control - If flow control is enabled, the destination becomes armed and instructs producers to begin decreasing their message flow. <p>A value of -1 specifies that messages paging, flow control, and threshold log messages are disabled for the destination. However, if a JMS template is specified for this destination, then -1 implies that the value will come from the template.</p> <p>Range of Values: <= MessagesMaximum; >MessagesThresholdLow</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> Messages paging cannot be dynamically disabled by resetting the value to -1. To dynamically disable paging, you could set the value to a very large number, so that paging would not be triggered.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSQueueMBean</p> <p><i>Attribute:</i> MessagesThresholdHigh</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 189-1

Attribute Label	Description	Value Constraints
Messages Threshold Low	<p>The lower threshold value that triggers events based on the number of messages stored in the destination. If the number of messages falls below this threshold, the triggered events are:</p> <ul style="list-style-type: none">■ Log Messages - A message is logged on the server indicating that the threshold condition has cleared.■ Messages Paging - If messages paging is enabled, paging is stopped (if paging is occurring).■ Flow Control - If flow control is enabled, the destination becomes disarmed and instructs producers to begin increasing their message flow. <p>A value of -1 specifies that messages paging, flow control, and threshold log messages are disabled for the destination. However, if a JMS template is specified for this destination, then -1 implies that the value will come from the template.</p> <p>Range of Values: < MessagesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSQueueMBean</p> <p><i>Attribute:</i> MessagesThresholdLow</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 189-1

Attribute Label	Description	Value Constraints
Messages Paging Enabled	<p>Specifies whether messages paging is enabled on the destination for temporarily swapping message bodies out from memory when the destination's message load reaches a specified threshold.</p> <ul style="list-style-type: none"> ■ default - If a JMS template is specified, then this value inherits the template's Messages Paging value. If no JMS template is configured for the destination, then the Default value is equivalent to false. ■ false - Messages paging is explicitly disabled for this destination. ■ true - If a paging store has been configured for the JMS server, and both the MessagesThresholdLow and MessagesThresholdHigh attribute values are greater than -1, then messages paging is enabled for this destination. <p><i>MBean:</i> weblogic.management.configuration.JMSQueueMBean</p> <p><i>Attribute:</i> MessagesPagingEnabled</p>	<p><i>Default:</i> default</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ default ■ false ■ true

Table 189-1

Attribute Label	Description	Value Constraints
Maximum Message Size	<p>The maximum allowable size for messages to be accepted from producers on this queue. The message size includes the message body, user-defined properties, and the user-defined JMS header fields: <code>JMSCorrelationID</code> and <code>JMSType</code>.</p> <p>The maximum message size is checked before messages are sent to a queue. Therefore, if a message exceeds the specified allowable size, it does not count against the queue's configured quota (bytes/messages maximum) or its upper thresholds, since the message is immediately rejected.</p> <p>When a producer sends a message that exceeds the specified maximum message size for the queue, it will receive a <code>JMSResourceAllocation</code> exception.</p> <p>The maximum message size is only enforced for the initial production of a message. Messages that are redirected to an error destination or forwarded to a member of a distributed queue are not checked for size. For example, if a queue and its corresponding error destination are configured with a maximum message size of 128K and 64K, respectively, a message of 96K could be redirected to the error destination (even though it exceeds the 64K maximum), but a producer could not directly send the 96K message to the error destination.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSQueueMBean</code></p> <p><i>Attribute:</i> <code>MaximumMessageSize</code></p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 2147483647</p> <p><i>Dynamic:</i> yes</p>





JMS Queue --> Monitoring

This page enables you to monitor all of the active JMS destinations (queues and topics) in your domain, as well as all of the active durable subscribers. To view all active destinations, click the Monitor all Active JMS Destinations link. To view all active durable subscribers, click the Monitor all Durable Subscribers link.

- For more information on monitoring JMS destinations, see [“Monitoring Active JMS Destinations” on page 234-3](#).
- For more information on monitoring durable subscribers, see [“Monitoring Durable Subscribers for Topics” on page 234-5](#).



JMS Queue --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to supply optional information about this configuration.

Tasks

[“Creating a JMS Queue” on page 232-14](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 191-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.JMSQueueMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

JMS Server --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines general configuration attributes for an instance of a JMS server. A JMS server manages connections and message requests on behalf of clients. You must first configure a JMS server before you can configure any destinations or consumers.

Tasks

“Configuring a JMS Server” on page 232-5

“Targeting and Deploying a JMS Server” on page 232-7

Related Topics

“[Managing WebLogic JMS](#)” in *Programming WebLogic JMS*

Attributes

Table 192-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this JMS server configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSServerMBean</p> <p><i>Attribute:</i> Name</p>	
Store	<p>The persistent store (either a disk-based file or a JDBC-accessible database) for this JMS server, which will be used as a physical repository for storing persistent message data.</p> <p>In order to select a store, first configure either a JMS file store or JMS JDBC store. The selected store cannot be the same as the selected paging store, or the same store used by any other JMS server.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSServerMBean</p> <p><i>Attribute:</i> Store</p>	

Table 192-1

Attribute Label	Description	Value Constraints
Paging Store	<p>The name of the paging store for this JMS server, which is used exclusively for paging out non-persistent messages for the JMS server and its destinations.</p> <p>In order to select a paging store, first configure a JMS file store to be used only as a paging store. The selected paging store cannot be the same as the selected non-paging store, or the same store used by any other JMS server.</p> <p>It is best to use a JMS file store rather than a JMS JDBC store, as the JDBC store will perform poorly in comparison without any real benefit.</p> <p>Note: Message paging is not enabled by default. However, a message paging store will be automatically created when either bytes paging or messages paging is enabled on the JMS server or it's destinations without preconfiguring a message paging store.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSServerMBean</code></p> <p><i>Attribute:</i> <code>PagingStore</code></p>	

Table 192-1

Attribute Label	Description	Value Constraints
Temporary Template	<p>The name of an existing JMS template to use when creating all temporary queues and topics for this JMS server. Specifying a value for this field allows JMS applications to create temporary destinations.</p> <p>The attribute values for a temporary destination are derived from the specified JMS template. If provided as part of the JMS template, the Store attribute values are ignored because temporary destinations do not support persistent messaging.</p> <p><i>Note:</i> If this attribute is set to None, attempts to create a temporary destination will fail.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSServerMBean</code></p> <p><i>Attribute:</i> <code>TemporaryTemplate</code></p>	<i>Default:</i> null

Table 192-1

Attribute Label	Description	Value Constraints
Expiration Scan Interval	<p>The amount of time, in seconds, that the JMS server pauses between its cycles of scanning its destinations for expired messages to process (according to the specified Expiration Policy on the destinations). The default value is 30 seconds. A value of 0 indicates that active scanning is disabled; messages still expire, but are cleaned up less quickly.</p> <p>Setting this value to a very large value also effectively disables active scanning for expired messages from the system. Users will not receive expired messages, and any expired messages that are discovered are removed from the system. However, expired messages sitting in idle destinations (i.e., an inactive queue or disconnected durable subscriber) will not be removed and will continue to consume system resources.</p> <p>The scanning and processing cycle for expired messages occurs as follows:</p> <ul style="list-style-type: none">■ After the specified waiting period, the JMS server devotes a separate thread to scan all of its local destinations for expired messages.■ After the scanning is completed, all discovered expired messages are processed according to the specified Expiration Policy on the destinations (Discard, Log, or Redirect).■ The entire process repeats after another specified waiting period. <p>Note: Since a new scan will not start until the current one is finished and the specified waiting period ends, an expired message could still remain in the system for the maximum scan waiting period plus the amount of time it takes to perform the scan and processing.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSServerMBean</p> <p><i>Attribute:</i> ExpirationScanInterval</p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 30</p> <p><i>Dynamic:</i> yes</p>



JMS Server --> Configuration --> Thresholds & Quotas

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines threshold and quota attributes for an instance of a JMS server. A JMS server manages connections and message requests on behalf of clients. You must first configure a JMS server before you can configure any destinations or consumers.

Tasks

[“Configuring a JMS Server” on page 232-5](#)

[“Targeting and Deploying a JMS Server” on page 232-7](#)

Related Topics

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

Attributes

Table 193-1

Attribute Label	Description	Value Constraints
Bytes Maximum	<p>The maximum bytes quota (total amount of bytes) that can be stored in this JMS server. The default value of <i>-1</i> specifies that there is no WebLogic-imposed limit on the number of bytes that can be stored. However, excessive bytes volume can cause memory saturation, so this value should correspond to the total amount of available system memory relative to the rest of your application load.</p> <p>This attribute is dynamically configurable.</p> <p>Range of Values: >= BytesThresholdHigh</p> <p><i>MBean:</i> weblogic.management.configuration.JMSServerMBean</p> <p><i>Attribute:</i> BytesMaximum</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 193-1

Attribute Label	Description	Value Constraints
Bytes Threshold High	<p>The upper threshold value that triggers events based on the number of bytes stored in the JMS server. If the number of bytes exceeds this threshold, the triggered events are:</p> <ul style="list-style-type: none"> ■ Log Messages - A message is logged on the server indicating a high threshold condition. ■ Bytes Paging - If bytes paging is enabled (and a paging store has been configured), then server bytes paging is started. ■ Flow Control - If flow control is enabled, the JMS server becomes armed and instructs producers to begin decreasing their message flow. <p>This attribute is dynamically configurable. A value of -1 specifies that server bytes paging, flow control, and threshold log messages are disabled for the JMS server.</p> <p>Range of Values: <= BytesMaximum; >BytesThresholdLow</p> <p><i>Note:</i> Bytes paging cannot be dynamically disabled by resetting the BytesThresholdHigh to -1. To disable paging, you could set the BytesThresholdHigh to a very large number, so that paging would not be triggered.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSServerMBean</p> <p><i>Attribute:</i> BytesThresholdHigh</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 193-1

Attribute Label	Description	Value Constraints
Bytes Threshold Low	<p>The lower threshold value that triggers events based on the number of bytes stored in the JMS server. If the number of bytes falls below this threshold, the triggered events are:</p> <ul style="list-style-type: none">■ Log Messages - A message is logged on the server indicating that the threshold condition has cleared.■ Bytes Paging - If bytes paging is enabled, paging is stopped (if paging is occurring).■ Flow Control - If flow control is enabled, the JMS server becomes disarmed and instructs producers to begin increasing their message flow. <p>This attribute is dynamically configurable. A value of -1 specifies that server bytes paging, flow control, and threshold log messages are disabled for the JMS server.</p> <p>Range of Values: < BytesThresholdHigh</p> <p><i>MBean:</i> weblogic.management.configuration.JMSServerMBean</p> <p><i>Attribute:</i> BytesThresholdLow</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 193-1

Attribute Label	Description	Value Constraints
Bytes Paging Enabled	<p>Specifies whether bytes paging is enabled on this JMS server for temporarily swapping message bodies out from memory when the JMS servers's byte load reaches a specified threshold.</p> <ul style="list-style-type: none">■ false - Server bytes paging is explicitly disabled.■ true - If a paging store has been configured, and both the BytesThresholdLow and BytesThresholdHigh values are greater than -1, then server bytes paging is enabled. <p>If either the BytesThresholdLow or BytesThresholdHigh attribute is undefined, or defined as -1, then server bytes paging is implicitly disabled--even though this flag is set to true.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSServerMBean</p> <p><i>Attribute:</i> BytesPagingEnabled</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false

Table 193-1

Attribute Label	Description	Value Constraints
Messages Maximum	<p>The maximum message quota (total amount of messages) that can be stored in this JMS server. The default value of <code>-1</code> specifies that there is no WebLogic-imposed limit on the number of messages that can be stored. However, excessive message volume can cause memory saturation, so this value should correspond to the total amount of available system memory relative to the rest of your application load.</p> <p>This attribute is dynamically configurable.</p> <p>Range of Values: <code>>= MessagesThresholdHigh</code></p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSServerMBean</code></p> <p><i>Attribute:</i> <code>MessagesMaximum</code></p>	<p><i>Minimum:</i> <code>-1</code></p> <p><i>Maximum:</i> <code>9223372036854775807</code></p> <p><i>Default:</i> <code>-1</code></p> <p><i>Dynamic:</i> <code>yes</code></p>

Table 193-1

Attribute Label	Description	Value Constraints
Messages Threshold High	<p>The upper threshold value that triggers events based on the number of messages stored in the JMS server. If the number of messages exceeds this threshold, the triggered events are:</p> <ul style="list-style-type: none">■ Log Messages - A message is logged on the server indicating a high threshold condition.■ Bytes Paging - If messages paging is enabled (and a paging store has been configured), then server messages paging is started.■ Flow Control - If flow control is enabled, the JMS server becomes armed and instructs producers to begin decreasing their message flow. <p>This attribute is dynamically configurable. A value of -1 specifies that server messages paging, flow control, and threshold log messages are disabled for the JMS server.</p> <p>Range of Values: <= MessagesMaximum; >MessagesThresholdLow</p> <p><i>Note:</i> Messages paging cannot be dynamically disabled by resetting the MessagesThresholdHigh to -1. To disable paging, you could set the MessagesThresholdHigh to a very large number, so that paging would not be triggered.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSServerMBean</p> <p><i>Attribute:</i> MessagesThresholdHigh</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 193-1

Attribute Label	Description	Value Constraints
Messages Threshold Low	<p>The lower threshold value that triggers events based on the number of messages stored in the JMS server. If the number of messages falls below this threshold, the triggered events are:</p> <ul style="list-style-type: none">■ Log Messages - A message is logged on the server indicating that the threshold condition has cleared.■ Bytes Paging - If messages paging is enabled, paging is stopped (if paging is occurring).■ Flow Control - If flow control is enabled, the JMS server becomes disarmed and instructs producers to begin increasing their message flow. <p>This attribute is dynamically configurable. A value of -1 specifies that server messages paging, flow control, and threshold log messages are disabled for the JMS server.</p> <p>Range of Values: < MessagesThresholdHigh</p> <p><i>MBean:</i> weblogic.management.configuration.JMSServerMBean</p> <p><i>Attribute:</i> MessagesThresholdLow</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 193-1

Attribute Label	Description	Value Constraints
Messages Paging Enabled	<p>Specifies whether messages paging is enabled on this JMS server for temporarily swapping message bodies out from memory when the JMS servers's message load reaches a specified threshold.</p> <ul style="list-style-type: none">■ false - Server bytes paging is explicitly disabled.■ true - If a paging store has been configured, and both the MessagesThresholdLow and MessagesThresholdHigh values are greater than -1, then server messages paging is enabled. <p>If either the MessagesThresholdLow or MessagesThresholdHigh attribute is undefined, or defined as -1, then server messages paging is implicitly disabled--even though this flag is set to true.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSServerMBean</p> <p><i>Attribute:</i> MessagesPagingEnabled</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false

Table 193-1

Attribute Label	Description	Value Constraints
Maximum Message Size	<p>The maximum size of a message that will be accepted from producers on this JMS server. The message size includes the message body, any user-defined properties, and the user-defined JMS header fields: <code>JMSCorrelationID</code> and <code>JMSType</code>. Producers sending messages that exceed the configured maximum message size for the JMS server will receive a <code>ResourceAllocationException</code>.</p> <p>The maximum message size is only enforced for the initial production of a message. Messages that are redirected to an error destination or forwarded to a member of a distributed destination are not checked for size. For instance, if a destination and its corresponding error destination are configured with a maximum message size of 128K bytes and 64K bytes, respectively, a message of 96K bytes could be redirected to the error destination (even though it exceeds the 64K byte maximum), but a producer could not directly send the 96K byte message to the error destination.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSServerMBean</code></p> <p><i>Attribute:</i> <code>MaximumMessageSize</code></p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 2147483647</p> <p><i>Dynamic:</i> yes</p>

Table 193-1

Attribute Label	Description	Value Constraints
Blocking Send Policy	<p>The JMS server's policy on whether to deliver smaller messages before larger ones when a destination has exceeded its message quota. This can occur when multiple senders are competing for space on the same JMS server.</p> <p>This policy is defined only for the JMS server; it cannot be set on individual destinations. Valid expiration policies are:</p> <ul style="list-style-type: none">■ FIFO - The first in, first out policy indicates that all send requests for the same destination are queued up one behind the other until space is available. No send request is permitted to successfully complete if there is another send request waiting for space before it. When space is limited, the FIFO policy prevents the starvation of larger requests because smaller requests cannot continuously use the remaining available space. Smaller requests are delayed, though not starved, until the larger request can be completed. When space does become available, requests are considered in the order in which they were made.■ Preemptive - Indicates that a send operation can preempt other blocking send operations if space is available. That is, if there is sufficient space for the current request, then that space is used even if there are other requests waiting for space. When space is limited, the Preemptive policy can result in the starvation of larger requests. For example, if there is insufficient available space for a large request, then it is queued up behind other existing requests. When space does become available, all requests are considered in the order in which they were originally made.	<p><i>Default:</i> FIFO</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ FIFO■ Preemptive <p><i>Dynamic:</i> yes</p>



JMS Server --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to supply optional information about this JMS server configuration.

Tasks

“Configuring a JMS Server” on page 232-5

“Targeting and Deploying a JMS Server” on page 232-7

Related Topics

“[Managing WebLogic JMS](#)” in *Programming WebLogic JMS*

Attributes

Table 194-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.JMSServerMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

JMS Server --> Monitoring

This page enable you to monitor all active JMS servers in your domain, as well as all the active JMS destinations (queues and topics) and JMS session pools.

To view all active servers, click the Monitor all Active JMS Servers link. To view all active destinations, click the Monitor all Active JMS Destinations link. To view all active session pools, click the Monitor all Active JMS Session Pools Runtimes link.

- For more information on monitoring JMS servers, see [“Monitoring Active JMS Servers” on page 234-3](#).
- For more information on monitoring JMS destinations, see [“Monitoring Active JMS Destinations” on page 234-3](#).
- For more information on monitoring JMS session pools, see [“Monitoring Active JMS Session Pools” on page 234-4](#).



JMS Server --> Target and Deploy

[Tasks](#) [Related Topics](#)

Overview

This tab enables you to target a single WebLogic Server instance or a migratable server target on which to deploy a JMS server. When a target server boots, the JMS server boots as well. If no target server is specified, the JMS server will not boot.

Migratable targets define a set of WebLogic Server instances in a cluster that can potentially host an "exactly-once" service, such as JMS. When a migratable target server boots, the JMS server boots as well on the *user-preferred* server in the cluster. However, a JMS server and all of its destinations can migrate to another server within the cluster in response to a server failure, or due to a scheduled migration or system maintenance.

Tasks

[“Configuring a JMS Server” on page 232-5](#)

[“Targeting and Deploying a JMS Server” on page 232-7](#)

[“Server --> Control --> JMS Migration Config.” on page 481-1](#)

Related Topics

[“Managing WebLogic JMS” in *Programming WebLogic JMS*](#)

[“Migrating a Pinned Service to a Target Server Instance”](#)

[“Migrating When the Currently Active Host is Unavailable”](#)



Active JMS Servers

This page displays the active JMS servers defined in your domain. A JMS server manages connections and message requests on behalf of JMS clients. You must first configure a JMS server before you can configure any destinations or session pools.

Monitoring information for a JMS server is available only after you have targeted it to either a WebLogic Server instance or a migratable target on the JMS --> Servers --> Target and Deploy tab.

- For more information on monitoring all active JMS destinations, see [“Active JMS Destinations” on page 164-1](#).
- For more information on monitoring all active JMS session pools, see [“Active JMS Session Pools” on page 201-1](#).

Active JMS Server Information

Name — Name of the JMS server.

Connections — Current number of active JMS connections to this WebLogic Server instance. Clicking the number takes you to the Active JMS Connections page, as described in [“Active JMS Connections” on page 160-1](#).

Active JMS Destinations — Current number of active JMS destinations running on the JMS server. Clicking the number takes you to the Active JMS Destinations page, as described in [“Active JMS Destinations” on page 164-1](#).

Destinations High — Greatest number of instantiated destinations for the JMS server since the last time the server was booted.

Destinations Total — Total number of instantiated destinations for the JMS server.

Bytes Current — Current number of bytes stored across all destinations for the JMS server. This does not include the pending bytes.

Bytes High — Greatest number of bytes stored across all destinations for the JMS server since the last time the server was booted.

Bytes Pending — Current number of bytes sent and received that are pending as the result of an uncommitted transaction or an acknowledgement. Pending bytes are over and above the current number of bytes.

Bytes Received Count — Total number of bytes received across all destinations for the JMS server since the last time the server was booted.

Bytes Threshold Time — Amount of time spent in a bytes threshold condition since the last time the server was booted.

Messages — Current number of messages stored across all destinations for the JMS server. This does not include pending messages.

Messages High — Greatest number of messages stored across all destinations for the JMS server since the last time the server was booted.

Messages Pending — Current number of messages sent and received that are pending as the result of an uncommitted transaction or an acknowledgement across all destinations for the JMS server. Pending messages are over and above the current number of messages.

Messages Received — Total number of messages received for all destinations stored across the JMS server since the last time the server was booted.

Messages Threshold Time — Amount of time spent in a messages threshold condition since the last time the server was booted.

Session Pools — Current number of session pools defined for the JMS server. Clicking the number takes you to the Active JMS Session Pools page, as described in [“Active JMS Session Pools” on page 201-1](#).

Session Pools High — Greatest number of session pools instantiated on the JMS server since the last time the server was booted.

Session Pools Total — Total number of instantiated sessions pools on the JMS server since the last time the server was booted.

JMS Server

A JMS server manages connections and message requests on behalf of JMS clients.

To configure a new JMS server, click the Configure a new JMS Server link.

- For more information on creating a JMS server, see [“Configuring a JMS Server” on page 232-5](#).
- For more information on monitoring a JMS server, see [“Monitoring Active JMS Servers” on page 234-3](#).



JMS Session Pool --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines general configuration attributes for a JMS session pool, which enable an application to process messages concurrently. Once you have defined a JMS server, optionally, you can configure one or more session pools for each JMS server.

Note: Session pools are now used rarely, as they are not a required part of the J2EE specification, do not support JTA user transactions, and are largely superseded by message-driven beans (MDBs), which are a required part of the J2EE specification.

Tasks

[“Creating a JMS Session Pool” on page 232-29](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 199-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this JMS session pool configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSSessionPoolMBean</p> <p><i>Attribute:</i> Name</p>	
Connection Factory	<p>The JNDI name of the connection factory for the session pool.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSSessionPoolMBean</p> <p><i>Attribute:</i> ConnectionFactory</p>	<i>Default:</i> null
Listener Class	<p>The name of the listener class for the session pool, which is used to receive and process messages concurrently.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSSessionPoolMBean</p> <p><i>Attribute:</i> ListenerClass</p>	

Table 199-1

Attribute Label	Description	Value Constraints
Acknowledge Mode	<p>The acknowledge mode used by non-transacted sessions within the session pool.</p> <p>For transacted sessions, messages are acknowledged automatically when the session is committed and this field is ignored.</p> <p>This attribute is not dynamically configurable.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSSessionPoolMBean</p> <p><i>Attribute:</i> AcknowledgeMode</p>	<p><i>Default:</i> Auto</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ Auto■ Client■ Dups-Ok■ None
Sessions Maximum	<p>The maximum number of concurrent sessions allowed for the session pool.</p> <p>A value of -1 indicates that there is no maximum.</p> <p>This attribute is dynamically configurable; however, it does not take effect until the session pool is restarted.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSSessionPoolMBean</p> <p><i>Attribute:</i> SessionsMaximum</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>
Transacted	<p>Indicates whether or not the session pool creates transacted sessions.</p> <p>This attribute is not dynamically configurable.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSSessionPoolMBean</p> <p><i>Attribute:</i> Transacted</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false



JMS Session Pool --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to supply optional information about this JMS session pool configuration.

Tasks

[“Creating a JMS Session Pool”](#) on page 232-29

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 200-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.JMSSessionPoolMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

Active JMS Session Pools

This page displays all the active JMS session pools defined for a JMS server. Session pools enable an application to process messages concurrently. Once you have defined a JMS server, you can configure one or more session pools for each server, and then create connection consumers for those session pools.

Monitoring information for a JMS session pool is available only after you have targeted the JMS server to either an independent WebLogic Server instance or a migratable server target on the JMS --> Servers --> Target and Deploy tab.

- For more information on creating a JMS session pool, see [“Creating a JMS Session Pool” on page 232-29](#).
- For more information on creating a JMS connection consumer, see [“Creating a JMS Connection Consumer” on page 232-30](#).

Active JMS Session Pools Information

Name — Name of the session pool.

Server — Name of the associated JMS server.

Consumers — Current number of connection consumers in the session pool.

Consumers High — Greatest number of simultaneous connection consumers in the session pool since the last time the server was booted.

Consumers Total — Total number of connection consumers since the last time the server was booted.



JMS Session Pool

Session pools enable an application to process messages concurrently. Once you have defined a JMS server, you can configure one or more session pools for each JMS server.

Note: Session pools are now used rarely, as they are not a required part of the J2EE specification, do not support JTA user transactions, and are largely superseded by message-driven beans (MDBs), which are a required part of the J2EE specification.

To create a JMS session pool, click the [Configure a new JMS Session Pool](#) link.

- For more information on creating a JMS session pool, see [“Creating a JMS Session Pool” on page 232-29](#).
- For more information on monitoring a JMS server, see [“Monitoring Active JMS Session Pools” on page 234-4](#).



Active JMS Sessions

Use this page to monitor session information for a JMS connection. A session defines a serial order for both the messages produced and the messages consumed, and can create multiple message producers and message consumers. The same thread can be used for producing and consuming messages.

- For more information, see [“Active JMS Connections” on page 160-1](#).
- For more information, see [“Fundamentals of WebLogic JMS”](#) in *Programming WebLogic JMS*.

Active JMS Sessions Information

Consumers — Current number of consumers for the session.

Consumers High — Greatest number of simultaneous consumers for the session since the last time the server was booted.

Consumers Total — Total number of consumers for the session since the last time the server was booted.

Producers — Current number of consumers for the session.

Producers High — Greatest number of simultaneous producers for the session since the last time the server was booted.

Producers Total — Total number of producers for the session since the last time the server was booted.

Acknowledge Mode — Indicates the acknowledge mode for the session as one of the following: AUTO, CLIENT, DUPS_OK, NO.

Messages Received — Total number of messages received by the session since the last time the server was booted.

Messages Pending — Current number of messages pending by the session that are pending as the result of an uncommitted transaction or an acknowledgement.

Messages Sent — Total number of messages sent by the session since the last time the server was booted.

Bytes Pending — Current number of bytes pending by the session that are pending as the result of an uncommitted transaction or an acknowledgement.

Bytes Sent — Total number of bytes sent by the session since the last time the server was booted.

Bytes Received — Total number of bytes received by the session since the last time the server was booted.

JMS Store

A persistent JMS store is a physical repository for storing persistent message data and durable subscribers. It can be either a disk-based file store or a JDBC-accessible database. A JMS file store can also be used for the paging of messages to disk when memory has been exhausted.

To create a JMS file store, click the [Configure a new JMS File Store](#) link. To create a JMS JDBC store, click the [Configure a new JMS JDBC Store](#) link.

- For more information on creating a JMS File Store, see [“Creating a JMS File Store” on page 232-23](#).
- For more information on creating a paging store, see [“Configuring a Paging Store” on page 232-24](#).
- For more information on creating a JMS JDBC Store, see [“Creating a JMS JDBC Store” on page 232-26](#).



JMS Template --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines general configuration attributes for a JMS template, which provides an efficient means of defining multiple destinations with similar attribute settings:

The general configuration attributes are inherited by the destinations that use them. However, the Name attribute is not inherited by the destination; this name is valid for the JMS template only. You must explicitly define a unique name for all destinations.

Tasks

[“Creating a JMS Template” on page 232-19](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 205-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this JMS template configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTemplateMBean</code></p> <p><i>Attribute:</i> <code>Name</code></p>	
Destination Keys	<p>The sort ordering for messages that arrive on destinations based on this template.</p> <p>The keys are ordered from most significant to least significant. If more than one key is specified, a key based on the <code>JMSMessageID</code> property can only be the last key in the list.</p> <p><i>Note:</i> If <code>JMSMessageID</code> is not defined in the key, it is implicitly assumed to be the last key and is set as "ascending" (first-in, first-out) for the sort order.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTemplateMBean</code></p> <p><i>Attribute:</i> <code>DestinationKey</code></p>	

JMS Template --> Configuration --> Override

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines configuration override attributes for a JMS template, which provides an efficient means of defining multiple destinations with similar attribute settings.

The configurable override attributes for a JMS template are the same as those configured for a destination. However, if the destination that is using a JMS template specifies an override value for an attribute, the override value is used.

Tasks

[“Creating a JMS Template” on page 232-19](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 206-1

Attribute Label	Description	Value Constraints
Priority Override	<p>The priority assigned to all messages that arrive at the destination, regardless of the Priority specified by the message producer.</p> <p>The default value (-1) specifies that the destination will not override the Priority setting.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> PriorityOverride</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>
Time To Live Override	<p>The time-to-live value assigned to all messages that arrive at the destination, regardless of the TimeToLive specified by the message producer.</p> <p>The default value (-1) specifies that the destination will not override the TimeToLive setting.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> TimeToLiveOverride</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 206-1

Attribute Label	Description	Value Constraints
Time To Deliver Override	<p>The default delay, either in milliseconds or as a schedule, between when a message is produced and when it is made visible on its target destination, regardless of the TimeToDeliver specified by the producer and/or the connection factory.</p> <p>The default value (-1) specifies that the destination will not override the TimeToDeliver setting.</p> <p>The Time To Deliver Override can be specified either as a long or as an advanced scheduling syntax (see "Setting Message Delivery Times" in <i>Programming JMS</i> for details.)</p> <p><i>Note:</i> Changing the Time To Deliver Override only affects future message delivery, it does not affect message delivery of already produced messages.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> TimeToDeliverOverride</p>	<p><i>Default:</i> "-1"</p> <p><i>Dynamic:</i> yes</p>
Delivery Mode Override	<p>The delivery mode assigned to all messages that arrive at the destination regardless of the DeliveryMode specified by the message producer.</p> <p>A value of No-Delivery specifies that the DeliveryMode will not be overridden.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> DeliveryModeOverride</p>	<p><i>Default:</i> No-Delivery</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ Persistent■ Non-Persistent■ No-Delivery <p><i>Dynamic:</i> yes</p>



JMS Template --> Configuration --> Redelivery

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines configuration redelivery attributes for a JMS template, which provides an efficient means of defining multiple destinations with similar attribute settings.

The configurable redelivery attributes for a JMS template are the same as those configured for a destination. However, if the destination that is using a JMS template specifies a message redelivery value for an attribute, that redelivery value is used.

Tasks

[“Creating a JMS Template” on page 232-19](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 207-1

Attribute Label	Description	Value Constraints
Redelivery Delay Override	<p>The delay, in milliseconds, before rolled back or recovered messages are redelivered, regardless of the RedeliveryDelay specified by the consumer and/or connection factory. Redelivered queue messages are put back into their originating destination; redelivered topic messages are put back into their originating subscription.</p> <p>The default value (-1) specifies that the destination will not override the RedeliveryDelay setting.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> Changing the Redelivery Delay Override only affects future rollbacks and recovers, it does not affect rollbacks and recovers that have already occurred.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> RedeliveryDelayOverride</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 207-1

Attribute Label	Description	Value Constraints
Redelivery Limit	<p>The number of redelivery tries a message can have before it is moved to the Error Destination specified on this page. This setting overrides any redelivery limit set by the message sender. If the redelivery limit is configured, but no error destination is configured, then persistent and non-persistent messages are simply dropped (deleted) when they reach their redelivery limit.</p> <p>The default value (-1) specifies that the destination will not override the message sender's redelivery limit setting.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; previously sent messages continue to use their original redelivery limit.</p> <p><i>Note:</i> The number of times a message has been redelivered is not persisted. This means that after a restart, the number of delivery attempts on each message is reset to zero.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTemplateMBean</code></p> <p><i>Attribute:</i> <code>RedeliveryLimit</code></p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 207-1

Attribute Label	Description	Value Constraints
Error Destination	<p>The name of the target destination (queue or topic) for messages that have reached their redelivery limit. If no error destination is configured on the local JMS server, then such messages are simply dropped. If a message has expired and the Expiration Policy is set to Redirect, then the message is moved to the specified error destination.</p> <p><i>Note:</i> Configured destination quotas do not apply to expired or redelivery limit messages that are redirected to an error destination. Such messages are still moved into an error destination even if that destination has reached its quotas.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTemplateMBean</code></p> <p><i>Attribute:</i> <code>ErrorDestination</code></p>	<p><i>Minimum:</i> -1</p> <p><i>Default:</i> null</p> <p><i>Dynamic:</i> yes</p>

JMS Template --> Configuration --> Thresholds & Quotas

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This page allows you to define the maximum message/byte quotas, the upper and lower message/byte thresholds, the maximum allowable message size, and whether message and/or byte paging is enabled for destinations created from this JMS template. The configurable threshold and quota attributes for a JMS template are the same as those configured for a destination. However, if a destination that is using a JMS template specifies a non-default value for a Threshold & Quota attribute, that value is used instead. See also the JMS Server quota, threshold, and paging configuration settings.

Tasks

[“Creating a JMS Template” on page 232-19](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 208-1

Attribute Label	Description	Value Constraints
Bytes Maximum	<p>The maximum bytes quota (total amount of bytes) that can be stored in a topic or queue destination. The default value of -1 specifies that there is no WebLogic-imposed limit on the number of bytes that can be stored in a destination. However, excessive bytes volume can cause memory saturation, so this value should correspond to the total amount of available system memory relative to the rest of your application load.</p> <p><i>Note:</i> Configured quotas do not apply to expired or error (redelivery limit) messages that are redirected to an error destination. Such messages are still moved into an error destination even if that destination has reached its quotas.</p> <p>Range of Values: >= BytesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> If a JMS template is used for distributed destination members, then this setting applies only to those specific members and not the distributed destination set as a whole.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTemplateMBean</code></p> <p><i>Attribute:</i> BytesMaximum</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 208-1

Attribute Label	Description	Value Constraints
Bytes Threshold High	<p>The upper threshold value that triggers events based on the number of bytes stored in a topic or queue destination. If the number of bytes exceeds this threshold, the triggered events are:</p> <ul style="list-style-type: none"> ■ Log Messages - A message is logged on the server indicating a high threshold condition. ■ Bytes Paging - If bytes paging is enabled (and a paging store has been configured), then destination-level bytes paging is started. ■ Flow Control - If flow control is enabled, the destination becomes armed and instructs producers to begin decreasing their message flow. <p>A value of -1 specifies that bytes paging, flow control, and threshold log messages are disabled for the destination -- unless the destination setting overrides this template.</p> <p>Range of Values: <= BytesMaximum; >BytesThresholdLow</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> Bytes paging cannot be dynamically disabled by resetting the value to -1. To dynamically disable paging, you could set the value to a very large number, so that paging would not be triggered.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> BytesThresholdHigh</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 208-1

Attribute Label	Description	Value Constraints
Bytes Threshold Low	<p>The lower threshold value that triggers events based on the number of bytes stored in a topic or queue destination. If the number of bytes falls below this threshold, the triggered events are:</p> <ul style="list-style-type: none">■ Log Messages - A message is logged on the server indicating that the threshold condition has cleared.■ Bytes Paging - If bytes paging is enabled, paging is stopped (if paging is occurring).■ Flow Control - If flow control is enabled, the destination becomes disarmed and instructs producers to begin increasing their message flow. <p>A value of -1 specifies that bytes paging, flow control, and threshold log messages are disabled for the destination -- unless the destination setting overrides this template.</p> <p>Range of Values: < BytesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> BytesThresholdLow</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 208-1

Attribute Label	Description	Value Constraints
Bytes Paging Enabled	<p>Specifies whether bytes paging is enabled for a topic or queue destination. Paging allows for temporarily swapping message bodies out from memory when a destination's bytes load reaches a specified threshold.</p> <ul style="list-style-type: none"> ■ If the check box is cleared (false), bytes paging is disabled for the template's destinations -- unless the destination setting overrides the template. ■ If the check box is selected, and if a paging store has been configured for the JMS Server, and both the BytesThresholdLow and BytesThresholdHigh attribute values are greater than -1, then bytes paging is enabled for the template's destinations -- unless the destination setting overrides the template. <p>Note: If no value is defined, then this setting defaults to false and bytes paging is disabled for the template's destinations -- unless the destination setting overrides the template.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> BytesPagingEnabled</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ true ■ false

Table 208-1

Attribute Label	Description	Value Constraints
Messages Maximum	<p>The maximum message quota (total amount of messages) that can be stored in a topic or queue destination. The default value of -1 specifies that there is no WebLogic-imposed limit on the number of messages that can be stored in the destination. However, excessive message volume can cause memory saturation, so this value should correspond to the total amount of available system memory relative to the rest of your application load.</p> <p><i>Note:</i> Configured quotas do not apply to expired or error (redelivery limit) messages that are redirected to an error destination. Such messages are still moved into an error destination even if that destination has reached its quotas.</p> <p>Range of Values: >= MessagesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> If a JMS template is used for distributed destination members, then this setting applies only to those specific members and not the distributed destination set as a whole.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> MessagesMaximum</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 208-1

Attribute Label	Description	Value Constraints
Messages Threshold High	<p>The upper threshold value that triggers events based on the number of messages stored in a topic or queue destination. If the number of messages exceeds this threshold, the triggered events are:</p> <ul style="list-style-type: none">■ Log Messages - A message is logged on the server indicating a high threshold condition.■ Messages Paging - If messages paging is enabled (and a paging store has been configured), then destination-level messages paging is started.■ Flow Control - If flow control is enabled, the destination becomes armed and instructs producers to begin decreasing their message flow. <p>A value of -1 specifies that messages paging, flow control, and threshold log messages are disabled for the destination -- unless the destination setting overrides this template.</p> <p>Range of Values: <= MessagesMaximum; >MessagesThresholdLow</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> Messages paging cannot be dynamically disabled by resetting the value to -1. To dynamically disable paging, you could set the value to a very large number, so that paging would not be triggered.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> MessagesThresholdHigh</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 208-1

Attribute Label	Description	Value Constraints
Messages Threshold Low	<p>The lower threshold value that triggers events based on the number of messages stored in a topic or queue destination. If the number of messages falls below this threshold, the triggered events are:</p> <ul style="list-style-type: none">■ Log Messages - A message is logged on the server indicating that the threshold condition has cleared.■ Messages Paging - If messages paging is enabled, paging is stopped (if paging is occurring).■ Flow Control - If flow control is enabled, the destination becomes disarmed and instructs producers to begin increasing their message flow. <p>A value of -1 specifies that messages paging, flow control, and threshold log messages are disabled for the destination -- unless the destination setting overrides this template.</p> <p>Range of Values: < MessagesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> MessagesThresholdLow</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 208-1

Attribute Label	Description	Value Constraints
Messages Paging Enabled	<p>Specifies whether messages paging is enabled for a topic or queue destination. Paging allows for temporarily swapping message bodies out from memory when a destination's message load reaches a specified threshold.</p> <ul style="list-style-type: none"> ■ If the check box is cleared (false), messages paging is disabled for the template's destinations -- unless the destination setting overrides this template. ■ If the check box is selected (true), and if a paging store has been configured for the JMS Server, and both the MessagesThresholdLow and MessagesThresholdHigh attribute values are greater than -1, then messages paging is enabled for the template's destinations -- unless the destination setting overrides the template. <p>Note: If no value is defined, this setting defaults to false and messages paging is disabled for the template's destinations -- unless the destination setting overrides this template.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTemplateMBean</code></p> <p><i>Attribute:</i> <code>MessagesPagingEnabled</code></p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ true ■ false

Table 208-1

Attribute Label	Description	Value Constraints
Maximum Message Size	<p>The maximum allowable size for messages to be accepted from producers on this destination. The message size includes the message body, user-defined properties, and the user-defined JMS header fields: <code>JMSCorrelationID</code> and <code>JMSType</code>.</p> <p>The maximum message size is checked before messages are sent to a destination. Therefore, if a message exceeds the specified allowable size, it does not count against the destination's configured quota (bytes/messages maximum) or its upper thresholds, since the message is immediately rejected.</p> <p>When a producer sends a message that exceeds the specified maximum message size for the destination, it will receive a <code>JMS ResourceAllocation</code> exception.</p> <p>The maximum message size is only enforced for the initial production of a message. Messages that are redirected to an error destination or forwarded to a member of a distributed destination are not checked for size. For example, if a destination and its corresponding error destination are configured with a maximum message size of 128K and 64K, respectively, a message of 96K could be redirected to the error destination (even though it exceeds the 64K maximum), but a producer could not directly send the 96K message to the error destination.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTemplateMBean</code></p> <p><i>Attribute:</i> <code>MaximumMessageSize</code></p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 2147483647</p> <p><i>Dynamic:</i> yes</p>





JMS Template --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to supply optional information about this JMS template configuration.

Tasks

[“Creating a JMS Template” on page 232-19](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 209-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

JMS Template

A JMS template provides an efficient means of defining multiple destinations (queues and topics) with similar attribute settings. This way, you do not need to re-enter every attribute setting each time you define a new destination; you can use the JMS template and override any setting to which you want to assign a new value. Also, you can modify shared attribute settings dynamically simply by modifying the template. In addition, a JMS template must be defined and assigned to a JMS server if you want to take advantage of the JMS Temporary Destination (queues and topics) feature.

To create a JMS template, click the Configure a new JMS Template link.

- For more information on creating a JMS template, see [“Creating a JMS Template” on page 232-19](#).



JMS Topic --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines general configuration attributes for a JMS topic. After defining a JMS server, you can configure its topics. You can configure one or more topics for each JMS server.

Note: You configure destinations explicitly or by configuring a JMS template that can be used to define multiple destinations with similar attribute settings, as described in [“JMS Template Tasks” on page 232-18](#).

Tasks

[“Creating a JMS Topic” on page 232-16](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 211-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTopicMBean</p> <p><i>Attribute:</i> Name</p>	
JNDI Name	<p>The JNDI name used to look up this topic destination within the JNDI namespace. If not specified, the topic destination name is not advertised through the JNDI namespace and cannot be looked up and used.</p> <p>This attribute is not dynamically configurable.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTopicMBean</p> <p><i>Attribute:</i> JNDIName</p>	
Replicate JNDI Name In Cluster	<p>Specifies whether the JNDI name for this topic destination (if specified) is replicated across the cluster. If this option is not selected, then the JNDI name for the topic (if specified) is only visible on the server hosting this topic.</p> <p>This attribute is not dynamically configurable.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTopicMBean</p> <p><i>Attribute:</i> JNDINameReplicated</p>	<p><i>Default:</i> true</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ false■ true

Table 211-1

Attribute Label	Description	Value Constraints
Enable Store	<p>Specifies whether this topic destination supports persistent messaging by using the JMS store specified by the JMS server.</p> <ul style="list-style-type: none"> ■ default - The topic destination uses the JMS store defined for the JMS server--if one is defined--and supports persistent messaging. However, if a JMS store is not defined for the JMS server, then persistent messages are automatically downgraded to non-persistent. ■ false - The topic destination does not support persistent messaging. ■ true - The topic destination does support persistent messaging. However, if a JMS store is not defined for the JMS server, then the configuration will fail and the JMS server will not boot. <p>This attribute is not dynamically configurable.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTopicMBean</code></p> <p><i>Attribute:</i> <code>StoreEnabled</code></p>	<p><i>Default:</i> default</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ default ■ false ■ true

Table 211-1

Attribute Label	Description	Value Constraints
Template	<p>The JMS template from which this topic destination is derived.</p> <p>If a JMS template is specified, the topic destination attributes that are set to their default values will inherit their values from the JMS template at run time. However, if this attribute is not defined, then the attributes for the topic destination must be specified as part of the destination.</p> <p>The Template attribute setting per topic destination is static. The JMS template's attributes, however, can be modified dynamically.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTopicMBean</code></p> <p><i>Attribute:</i> <code>Template</code></p>	
Destination Keys	<p>The sort ordering for messages that arrive on this topic destination.</p> <p>The keys are ordered from most significant to least significant. If more than one key is specified, a key based on the <code>JMSMessageID</code> property can only be the last key in the list.</p> <p><i>Note:</i> If <code>JMSMessageID</code> is not defined in the key, it is implicitly assumed to be the last key and is set as "ascending" (first-in, first-out) for the sort order.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTopicMBean</code></p> <p><i>Attribute:</i> <code>DestinationKey</code></p>	

JMS Topic --> Configuration --> Multicast

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines multicast configuration attributes for a topic destination. After defining a JMS server, you can configure its topics. You can configure one or more topics for each JMS server.

Note: You configure destinations explicitly or by configuring a JMS template that can be used to define multiple destinations with similar attribute settings, as described in [“JMS Template Tasks” on page 232-18](#).

Tasks

[“Creating a JMS Topic” on page 232-16](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 212-1

Attribute Label	Description	Value Constraints
Multicast Address	<p>The IP address used for multicasting by the topic destination. This address is used to transmit messages to multicast consumers.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTopicMBean</code></p> <p><i>Attribute:</i> <code>MulticastAddress</code></p>	<p><i>Configurable:</i> yes</p>
Multicast TTL	<p>The number of network hops that a multicast message is allowed to travel.</p> <p>This is the Time-To-Live value used for multicasting, which specifies the number of routers that the message can traverse enroute to the consumers. A value of 1 indicates that the message will not traverse any routers and is limited to one subnet.</p> <p>This value is independent of the <code>JMSEExpirationTime</code> value.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTopicMBean</code></p> <p><i>Attribute:</i> <code>MulticastTTL</code></p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 255</p> <p><i>Default:</i> 1</p> <p><i>Configurable:</i> yes</p> <p><i>Required:</i> yes</p>
Multicast Port	<p>The IP port used for multicasting for the topic. This port is used to transmit messages to multicast consumers.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTopicMBean</code></p> <p><i>Attribute:</i> <code>MulticastPort</code></p>	<p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 65535</p> <p><i>Default:</i> 6001</p> <p><i>Configurable:</i> yes</p> <p><i>Required:</i> yes</p>

JMS Topic --> Configuration --> Overrides

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines override configuration attributes for a topic destination. After defining a JMS server, you can configure its topics. You can configure one or more topics for each JMS server.

Note: You configure destinations explicitly or by configuring a JMS template that can be used to define multiple destinations with similar attribute settings, as described in [“JMS Template Tasks” on page 232-18](#).

Tasks

[“Creating a JMS Topic” on page 232-16](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 213-1

Attribute Label	Description	Value Constraints
Priority Override	<p>The priority assigned to all messages that arrive at the destination, regardless of the Priority specified by the message producer.</p> <p>If no JMS template is specified for this destination, then -1 means that the destination will not override the Priority setting. Otherwise, -1 means that the value comes from the template.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTopicMBean</p> <p><i>Attribute:</i> PriorityOverride</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>
Time To Live Override	<p>The time-to-live value assigned to all messages that arrive at the destination, regardless of the TimeToLive specified by the message producer.</p> <p>If no JMS template is specified for this destination, then -1 means that the destination will not override the TimeToLive setting. Otherwise, -1 means that the value comes from the template.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTopicMBean</p> <p><i>Attribute:</i> TimeToLiveOverride</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 213-1

Attribute Label	Description	Value Constraints
Time To Deliver Override	<p>The default delay, either in milliseconds or as a schedule, between when a message is produced and when it is made visible on its target destination, regardless of the delivery time specified by the producer and/or connection factory.</p> <p>If no JMS template is specified for this destination, then -1 means that the destination will not override the TimeToDeliver setting. Otherwise, -1 means that the value comes from the template.</p> <p>The Time To Deliver Override can be specified either as a long or as an advanced scheduling syntax (see "Setting Message Delivery Times" in <i>Programming JMS</i> for details.)</p> <p><i>Note:</i> Changing the Time To Deliver Override only affects future message delivery, it does not affect message delivery of already produced messages.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTopicMBean</p> <p><i>Attribute:</i> TimeToDeliverOverride</p>	<p><i>Default:</i> "-1"</p> <p><i>Dynamic:</i> yes</p>

Table 213-1

Attribute Label	Description	Value Constraints
Delivery Mode Override	<p>The delivery mode assigned to all messages that arrive at the destination regardless of the <code>DeliveryMode</code> specified by the message producer.</p> <p>If no JMS template is specified for this destination, then No-Delivery means the <code>DeliveryMode</code> will not be overridden. Otherwise, No-Delivery means that the value comes from the template.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTopicMBean</code></p> <p><i>Attribute:</i> <code>DeliveryModeOverride</code></p>	<p><i>Default:</i> No-Delivery</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ Persistent■ Non-Persistent■ No-Delivery <p><i>Dynamic:</i> yes</p>

JMS Topic --> Configuration --> Redelivery

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines message redelivery configuration attributes for a topic destination. After defining a JMS server, you can configure its topics. You can configure one or more topics for each JMS server.

Note: You configure destinations explicitly or by configuring a JMS template that can be used to define multiple destinations with similar attribute settings, as described in [“JMS Template Tasks” on page 232-18](#).

Tasks

[“Creating a JMS Topic” on page 232-16](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 214-1

Attribute Label	Description	Value Constraints
Redelivery Delay Override	<p>The delay, in milliseconds, before rolled back or recovered messages are redelivered, regardless of the RedeliveryDelay specified by the consumer and/or connection factory. Redelivered queue messages are put back into their originating destination; redelivered topic messages are put back into their originating subscription.</p> <p>The default value (-1) specifies that the destination will not override the RedeliveryDelay setting.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> Changing the Redelivery Delay Override only affects future rollbacks and recovers, it does not affect rollbacks and recovers that have already occurred.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTopicMBean</p> <p><i>Attribute:</i> RedeliveryDelayOverride</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 214-1

Attribute Label	Description	Value Constraints
Redelivery Limit	<p>The number of redelivery tries a message can have before it is moved to the Error Destination specified on this page. This setting overrides any redelivery limit set by the message sender. If the redelivery limit is configured, but no error destination is configured, then persistent and non-persistent messages are simply dropped (deleted) when they reach their redelivery limit.</p> <p>The default value (-1) specifies that the destination will not override the message sender's redelivery limit setting.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; previously sent messages continue to use their original redelivery limit.</p> <p><i>Note:</i> The number of times a message has been redelivered is not persisted. This means that after a restart, the number of delivery attempts on each message is reset to zero.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTopicMBean</code></p> <p><i>Attribute:</i> <code>RedeliveryLimit</code></p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 214-1

Attribute Label	Description	Value Constraints
Error Destination	<p>The name of the target destination (queue or topic) for messages that have reached their redelivery limit. If no error destination is configured on the local JMS server, then such messages are simply dropped. If a message has expired and the Expiration Policy is set to Redirect, then the message is moved to the specified error destination.</p> <p>If this destination has a template, the none option indicates that the error destination comes from the template. If this destination has no template, none indicates that there is no error destination configured.</p> <p><i>Note:</i> Configured destination quotas do not apply to expired or redelivery limit messages that are redirected to an error destination. Such messages are still moved into an error destination even if that destination has reached its quotas.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTopicMBean</p> <p><i>Attribute:</i> ErrorDestination</p>	<p><i>Minimum:</i> -1</p> <p><i>Default:</i> null</p> <p><i>Dynamic:</i> yes</p>

JMS Topic --> Configuration --> Thresholds & Quotas

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines the bytes/messages threshold and quota configuration attributes for a JMS topic. After defining a JMS server, you can configure its topics. You can configure one or more topics for each JMS server.

Note: You configure destinations explicitly or by configuring a JMS template that can be used to define multiple destinations with similar attribute settings, as described in [“JMS Template Tasks” on page 232-18](#).

Tasks

[“Creating a JMS Topic” on page 232-16](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 215-1

Attribute Label	Description	Value Constraints
Bytes Maximum	<p>The maximum bytes quota (total amount of bytes) that can be stored in this destination. The default value of -1 specifies that there is no WebLogic-imposed limit on the number of bytes that can be stored in the destination. However, excessive bytes volume can cause memory saturation, so this value should correspond to the total amount of available system memory relative to the rest of your application load.</p> <p><i>Note:</i> Configured quotas do not apply to expired or error (redelivery limit) messages that are redirected to an error destination. Such messages are still moved into an error destination even if that destination has reached its quotas.</p> <p>Range of Values: >= BytesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> If a JMS template is used for distributed destination members, then this setting applies only to those specific members and not the distributed destination set as a whole.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTopicMBean</p> <p><i>Attribute:</i> BytesMaximum</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 215-1

Attribute Label	Description	Value Constraints
Bytes Threshold High	<p>The upper threshold value that triggers events based on the number of bytes stored in this destination. If the number of bytes exceeds this threshold, the triggered events are:</p> <ul style="list-style-type: none"> ■ Log Messages - A message is logged on the server indicating a high threshold condition. ■ Bytes Paging - If bytes paging is enabled (and a paging store has been configured), then destination-level bytes paging is started. ■ Flow Control - If flow control is enabled, the destination becomes armed and instructs producers to begin decreasing their message flow. <p>A value of -1 specifies that bytes paging, flow control, and threshold log messages are disabled for the destination. However, if a JMS template is specified for this destination, then -1 implies that the value will come from the template.</p> <p>Range of Values: <= BytesMaximum; >BytesThresholdLow</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> Bytes paging cannot be dynamically disabled by resetting the value to -1. To dynamically disable paging, you could set the value to a very large number, so that paging would not be triggered.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTopicMBean</p> <p><i>Attribute:</i> BytesThresholdHigh</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 215-1

Attribute Label	Description	Value Constraints
Bytes Threshold Low	<p>The lower threshold value that triggers events based on the number of bytes stored in the destination. If the number of bytes falls below this threshold, the triggered events are:</p> <ul style="list-style-type: none">■ Log Messages - A message is logged on the server indicating that the threshold condition has cleared.■ Bytes Paging - If bytes paging is enabled, paging is stopped (if paging is occurring).■ Flow Control - If flow control is enabled, the destination becomes disarmed and instructs producers to begin increasing their message flow. <p>A value of -1 specifies that bytes paging, flow control, and threshold log messages are disabled for the destination. However, if a JMS template is specified for this destination, then -1 implies that the value will come from the template.</p> <p>Range of Values: < BytesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTopicMBean</code></p> <p><i>Attribute:</i> <code>BytesThresholdLow</code></p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 215-1

Attribute Label	Description	Value Constraints
Bytes Paging Enabled	<p>Specifies whether bytes paging is enabled on the destination for temporarily swapping message bodies out from memory when the destination's bytes load reaches a specified threshold.</p> <ul style="list-style-type: none">■ default - If a JMS template is specified, then this value inherits the template's Bytes Paging value. If no JMS template is configured for the destination, then the Default value is equivalent to false.■ false - Bytes paging is explicitly disabled for this destination.■ true - If a paging store has been configured for the JMS server, and both the BytesThresholdLow and BytesThresholdHigh attribute values are greater than -1, then bytes paging is enabled for this destination. <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTopicMBean</code></p> <p><i>Attribute:</i> <code>BytesPagingEnabled</code></p>	<p><i>Default:</i> default</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ default■ false■ true

Table 215-1

Attribute Label	Description	Value Constraints
Messages Maximum	<p>The maximum message quota (total amount of messages) that can be stored in this destination. The default value of -1 specifies that there is no WebLogic-imposed limit on the number of messages that can be stored in the destination. However, excessive message volume can cause memory saturation, so this value should correspond to the total amount of available system memory relative to the rest of your application load.</p> <p><i>Note:</i> Configured quotas do not apply to expired or error (redelivery limit) messages that are redirected to an error destination. Such messages are still moved into an error destination even if that destination has reached its quotas.</p> <p>Range of Values: >= MessagesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> If a JMS template is used for distributed destination members, then this setting applies only to those specific members and not the distributed destination set as a whole.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTopicMBean</p> <p><i>Attribute:</i> MessagesMaximum</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 215-1

Attribute Label	Description	Value Constraints
Messages Threshold High	<p>The upper threshold value that triggers events based on the number of messages stored in the destination. If the number of messages exceeds this threshold, the triggered events are:</p> <ul style="list-style-type: none">■ Log Messages - A message is logged on the server indicating a high threshold condition.■ Messages Paging - If messages paging is enabled (and a paging store has been configured), then destination-level messages paging is started.■ Flow Control - If flow control is enabled, the destination becomes armed and instructs producers to begin decreasing their message flow. <p>A value of -1 specifies that messages paging, flow control, and threshold log messages are disabled for the destination. However, if a JMS template is specified for this destination, then -1 implies that the value will come from the template.</p> <p>Range of Values: <= MessagesMaximum; >MessagesThresholdLow</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>Note:</i> Messages paging cannot be dynamically disabled by resetting the value to -1. To dynamically disable paging, you could set the value to a very large number, so that paging would not be triggered.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTopicMBean</p> <p><i>Attribute:</i> MessagesThresholdHigh</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 215-1

Attribute Label	Description	Value Constraints
Messages Threshold Low	<p>The lower threshold value that triggers events based on the number of messages stored in the destination. If the number of messages falls below this threshold, the triggered events are:</p> <ul style="list-style-type: none">■ Log Messages - A message is logged on the server indicating that the threshold condition has cleared.■ Messages Paging - If messages paging is enabled, paging is stopped (if paging is occurring).■ Flow Control - If flow control is enabled, the destination becomes disarmed and instructs producers to begin increasing their message flow. <p>A value of -1 specifies that messages paging, flow control, and threshold log messages are disabled for the destination. However, if a JMS template is specified for this destination, then -1 implies that the value will come from the template.</p> <p>Range of Values: < MessagesThresholdHigh</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTopicMBean</p> <p><i>Attribute:</i> MessagesThresholdLow</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 215-1

Attribute Label	Description	Value Constraints
Messages Paging Enabled	<p>Specifies whether messages paging is enabled on the destination for temporarily swapping message bodies out from memory when the destination's message load reaches a specified threshold.</p> <ul style="list-style-type: none">■ default - If a JMS template is specified, then this value inherits the template's Messages Paging value. If no JMS template is configured for the destination, then the Default value is equivalent to false.■ false - Messages paging is explicitly disabled for this destination.■ true - If a paging store has been configured for the JMS server, and both the MessagesThresholdLow and MessagesThresholdHigh attribute values are greater than -1, then messages paging is enabled for this destination. <p><i>MBean:</i> weblogic.management.configuration.JMSTopicMBean</p> <p><i>Attribute:</i> MessagesPagingEnabled</p>	<p><i>Default:</i> default</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ default■ false■ true

Table 215-1

Attribute Label	Description	Value Constraints
Maximum Message Size	<p>The maximum allowable size for messages to be accepted from producers on this topic. The message size includes the message body, user-defined properties, and the user-defined JMS header fields: <code>JMSCorrelationID</code> and <code>JMSType</code>.</p> <p>The maximum message size is checked before messages are sent to a topic. Therefore, if a message exceeds the specified allowable size, it does not count against the topic's configured quota (bytes/messages maximum) or its upper thresholds, since the message is immediately rejected.</p> <p>When a producer sends a message that exceeds the specified maximum message size for the topic, it will receive a <code>JMSResourceAllocation</code> exception.</p> <p>The maximum message size is only enforced for the initial production of a message. Messages that are redirected to an error destination or forwarded to a member of a distributed topic are not checked for size. For instance, if a topic and its corresponding error destination are configured with a maximum message size of 128K and 64K, respectively, a message of 96K could be redirected to the error destination (even though it exceeds the 64K maximum), but a producer could not directly send the 96K message to the error destination.</p> <p>This attribute is dynamically configurable, but only incoming messages are impacted; stored messages are not impacted.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTopicMBean</code></p> <p><i>Attribute:</i> <code>MaximumMessageSize</code></p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 2147483647</p> <p><i>Dynamic:</i> yes</p>





JMS Topic --> Monitoring

This page enables you to monitor all of the active JMS destinations (topics and queues) in your domain, as well as all of the active durable subscribers. To view all active destinations, click the Monitor all Active JMS Destinations link. To view all active durable subscribers, click the Monitor all Durable Subscribers link.

- For more information on monitoring JMS destinations, see [“Monitoring Active JMS Destinations” on page 234-3](#).
- For more information on monitoring durable subscribers, see [“Monitoring Durable Subscribers for Topics” on page 234-5](#).



JMS Topic --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to supply optional information about this configuration.

Tasks

[“Creating a JMS Topic” on page 232-16](#)

[“JMS Queue and Topic Destination Tasks” on page 232-14](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 217-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.JMSTopicMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

JMS Distributed Destinations

You can configure multiple physical WebLogic JMS destinations (queues and topics) as members of a single distributed destination set that can be served by multiple WebLogic Server instances within a cluster. Once configured, your producers and consumers are able to send and receive to the distributed destination. WebLogic JMS then distributes the messaging load across all available destination members within the distributed destination. When a destination member becomes unavailable, traffic is then redirected toward other available destination members in the set.

To create a new distributed topic, click the [Configure a new Distributed Topic](#) link. To create a new distributed queue, click the [Configure a new Distributed Queue](#) link.

For more information on creating a new distributed topic, see:

- [“Creating a Distributed Topic and Creating Members Automatically” on page 232-32](#)
- [“Creating a Distributed Topic and Creating Members Automatically” on page 232-32](#)

For more information on creating a new distributed queue, see:

- [“Creating a Distributed Queue and Creating Members Automatically” on page 232-37](#)
- [“Creating a Distributed Queue and Adding Existing Physical Queues as Members Manually” on page 232-40](#)



Foreign JMS Destination --> Configuration

--> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to supply optional information about this foreign JMS destination (topic or queue) configuration.

Tasks

[“Creating a Foreign JMS Destination” on page 232-49](#)

Related Topics

[“Using Foreign JMS Providers with WebLogic Server” white paper on dev2dev.bea.com](#)

Attributes

Table 219-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.ForeignJMSDestinationMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

Foreign JMS Connection Factory --> Configuration --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to supply optional information about this foreign JMS connection factory configuration.

Tasks

[“Creating a Foreign JMS Connection Factory” on page 232-48](#)

Related Topics

[“Using Foreign JMS Providers with WebLogic Server” white paper on dev2dev.bea.com](#)

Attributes

Table 220-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.ForeignJMSConnectionFactoryMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

Foreign JMS Connection Factory

A foreign JMS connection factory contains the JNDI name of the connection factory in the remote JNDI provider, the JNDI name that the connection factory is mapped to in the local WebLogic Server JNDI tree, and an optional user name and password.

To configure a new foreign JMS connection factory, click the [Configure a new Foreign JMS Connection Factory](#) link.

- For more information on creating a Foreign JMS connection factory, see [“Creating a Foreign JMS Connection Factory” on page 232-48](#).



Foreign JMS Destination --> Configuration

--> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines general configuration attributes for an instance of a foreign JMS server. A *foreign* JMS destination represents either a queue or a topic. It contains the destination JNDI name that is looked up on the foreign JNDI provider and the JNDI name that the destination is mapped to on the local WebLogic Server. When the foreign destination is looked up on the local server, a lookup is performed on the remote JNDI directory, and the destination object is returned from that directory.

Tasks

[“Creating a Foreign JMS Destination” on page 232-49](#)

Related Topics

[“Using Foreign JMS Providers with WebLogic Server” white paper on dev2dev.bea.com](#)

Attributes

Table 222-1

Attribute Label	Description	Value Constraints
Name	The name of this foreign JMS destination configuration. WebLogic Server uses an MBean to implement and persist the configuration. <i>MBean:</i> weblogic.management.configuration.ForeignJMSDestinationMBean <i>Attribute:</i> Name	
Local JNDI Name	The name that the remote object will be bound to in the local server's JNDI tree. This is the name that should be used to look up the object on the local server. <i>MBean:</i> weblogic.management.configuration.ForeignJMSDestinationMBean <i>Attribute:</i> LocalJNDIName	<i>Dynamic:</i> yes
Remote JNDI Name	The name of the remote object that will be looked up in the remote JNDI directory. <i>MBean:</i> weblogic.management.configuration.ForeignJMSDestinationMBean <i>Attribute:</i> RemoteJNDIName	<i>Dynamic:</i> yes

Foreign JMS Connection Factory --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines general configuration attributes for an instance of a foreign JMS connection factory. A *foreign* JMS connection factory contains the JNDI name of the connection factory in the remote JNDI provider, the JNDI name that the connection factory is mapped to in the local WebLogic Server JNDI tree, and an optional user name and password.

Tasks

[“Creating a Foreign JMS Connection Factory” on page 232-48](#)

Related Topics

[“Using Foreign JMS Providers with WebLogic Server” white paper on dev2dev.bea.com](#)

Attributes

Table 223-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this foreign JMS connection factory configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> weblogic.management.configuration.ForeignJMSConnectionFactoryMBean</p> <p><i>Attribute:</i> Name</p>	
Local JNDI Name	<p>The name that the remote object will be bound to in the local server's JNDI tree. This is the name that should be used to look up the object on the local server.</p> <p><i>MBean:</i> weblogic.management.configuration.ForeignJMSConnectionFactoryMBean</p> <p><i>Attribute:</i> LocalJNDIName</p>	<i>Dynamic:</i> yes
Remote JNDI Name	<p>The name of the remote object that will be looked up in the remote JNDI directory.</p> <p><i>MBean:</i> weblogic.management.configuration.ForeignJMSConnectionFactoryMBean</p> <p><i>Attribute:</i> RemoteJNDIName</p>	<i>Dynamic:</i> yes

Table 223-1

Attribute Label	Description	Value Constraints
User Name	<p>The user name that will be passed when opening a connection to the remote JMS server represented by this connection factory. If not set, then no user name will be used.</p> <p><i>MBean:</i> weblogic.management.configuration.ForeignJMSConnectionFactoryMBean</p> <p><i>Attribute:</i> Username</p>	<i>Dynamic:</i> yes
Password	<p>The password that will be used in conjunction with the user name specified in the "Username" parameter.</p> <p><i>MBean:</i> weblogic.management.configuration.ForeignJMSConnectionFactoryMBean</p> <p><i>Attribute:</i> Password</p>	<i>Dynamic:</i> yes



Foreign JMS Destination

A *foreign* JMS destination represents either a queue or a topic. It contains the destination JNDI name that is looked up on the foreign JNDI provider and the JNDI name that the destination is mapped to on the local WebLogic Server. When the foreign destination is looked up on the local server, a lookup is performed on the remote JNDI directory, and the destination object is returned from that directory.

To configure a new foreign JMS destination, click the [Configure a new Foreign JMS Destination](#) link.

- For more information on creating a Foreign JMS destination, see [“Creating a Foreign JMS Destination” on page 232-49](#).



Foreign JMS Server --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines general configuration attributes for an instance of a foreign JMS server. A *foreign* JMS server represents a JNDI provider that is outside the WebLogic JMS server. It contains information that allows a local WebLogic Server to reach a remote JNDI provider, thereby allowing for a number of foreign JMS connection factory and destination objects to be defined on one JNDI directory.

Tasks

[“Creating a Foreign JMS Server” on page 232-47](#)

Related Topics

[“Using Foreign JMS Providers with WebLogic Server” white paper on dev2dev.bea.com](#)

Attributes

Table 225-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ForeignJMSServerMBean</code></p> <p><i>Attribute:</i> <code>Name</code></p>	
JNDI Initial Context Factory	<p>The name of the class that must be instantiated to access the JNDI provider. This class name depends on the provider and vendor that are being used.</p> <p>It defaults to <code>weblogic.jndi.WLInitialContextFactory</code>, which is the correct value for WebLogic Server. This value corresponds to the standard JNDI property, <code>java.naming.factory.initial</code>.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ForeignJMSServerMBean</code></p> <p><i>Attribute:</i> <code>InitialContextFactory</code></p>	<p><i>Default:</i> <code>weblogic.jndi.WLInitialContextFactory</code></p> <p><i>Dynamic:</i> <code>yes</code></p>

Table 225-1

Attribute Label	Description	Value Constraints
JNDI Connection URL	<p>The URL that WebLogic Server will use to contact the JNDI provider. The syntax of this URL depends on which JNDI provider is being used. This value corresponds to the standard JNDI property, <code>java.naming.provider.url</code>.</p> <p>For WebLogic JMS, leave this field blank if you are referencing WebLogic JMS objects within the same cluster. When no URL is specified, look-ups will be performed on the JNDI server within the WebLogic Server instance where this connection factory is deployed.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ForeignJMSServerMBean</code></p> <p><i>Attribute:</i> <code>ConnectionURL</code></p>	<i>Dynamic:</i> yes
JNDI Properties	<p>Additional properties that must be set for the particular JNDI provider. These properties will be passed directly to the constructor for the JNDI provider's <code>InitialContext</code> class.</p> <p>This field must be filled in using a <code>name=value<return>name=value</code> format.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ForeignJMSServerMBean</code></p> <p><i>Attribute:</i> <code>JNDIProperties</code></p>	<i>Dynamic:</i> yes



Foreign JMS Server --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to supply optional information about this foreign JMS server configuration.

Tasks

[“Creating a Foreign JMS Server” on page 232-47](#)

Related Topics

[“Using Foreign JMS Providers with WebLogic Server” white paper on dev2dev.bea.com](#)

Attributes

Table 226-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.ForeignJMSServerMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

Foreign JMS Server --> Target and Deploy

[Tasks](#) [Related Topics](#)

Overview

This tab enables you to target an independent WebLogic Server instance or a server cluster on which to deploy a foreign JMS server.

Tasks

[“Creating a Foreign JMS Server” on page 232-47](#)

Related Topics

[“Using Foreign JMS Providers with WebLogic Server” white paper on dev2dev.bea.com](#)



Foreign JMS Server

WebLogic JMS enables you to reference foreign (that is, external) JMS providers within a local WebLogic JNDI tree. Using the Foreign JMS Server node, you can quickly map a foreign JMS provider so that its connection factories and destinations appear in the WebLogic JNDI tree as a local JMS objects. A Foreign JMS Server configuration can also be used to reference remote instances of WebLogic Server in another cluster or domain in the local WebLogic JNDI tree

To configure a new foreign JMS server, click the [Configure a new Foreign JMS Server](#) link.

- For more information on creating a Foreign JMS server, see [“Creating a Foreign JMS Server” on page 232-47](#).



JMS Queue --> Configuration --> Expiration Policy

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines the action that a queue destination should take when an expired message is encountered: discard the message, discard the message and log its removal, or redirect the message to an error destination.

Note: You configure queues explicitly or by configuring a JMS template that can be used to define multiple queues with similar attribute settings, as described in [“JMS Template Tasks” on page 232-18](#).

Tasks

[“Creating a JMS Queue” on page 232-14](#)

[“Handling Expired Messages” on page 235-26](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 229-1

Attribute Label	Description	Value Constraints
Expiration Policy	<p>An optional policy to use when expired messages are found on a queue destination.</p> <p>None - If a JMS template is specified for this queue, then <i>None</i> means that the policy is inherited from the template. Otherwise, <i>None</i> is equivalent to <i>Discard</i>.</p> <p>Discard - Removes expired messages from the messaging system. The removal is not logged and the message is not redirected to another location.</p> <p>Log - Removes expired messages from the system and writes an entry to the server log file indicating that the messages have been removed from the system. The actual information that is logged is defined by the Expiration Logging Policy.</p> <p>Redirect - Moves expired messages from their current location to the Error Destination defined for the destination. The message retains its body, and all of its properties. The message also retains all of its header fields, but with some exceptions:</p> <ul style="list-style-type: none">■ The destination for the message becomes the error destination.■ All property overrides associated with the error destination are applied to the redirected message.■ If there is no Time-To-Live Override value set for the error destination, then the message receives a new Expiration Time of zero (indicating that it will not expire again). <p>It is illegal to use the Redirect policy when there is no valid error destination defined for the destination. Similarly, it is illegal to remove the error destination for a destination using the Redirect policy.</p> <p>Note: The Maximum Message quota is only enforced for sending new messages. It is ignored when moving messages because of the Redirect policy.</p> <p>Administration Console Online Help</p> <p><i>MBean:</i> weblogic.management.configuration.</p>	<p><i>Valid values:</i></p> <ul style="list-style-type: none">■ Discard■ Log■ Redirect <p><i>Dynamic:</i> yes</p>

Table 229-1

Attribute Label	Description	Value Constraints
Expiration Logging Policy	<p>The policy that defines what information about the message is logged when the Expiration Policy on this queue is set to Log. The valid logging policy values are:</p> <ul style="list-style-type: none">■ %header% - All the JMS header fields are logged.■ %properties% - All the user properties are logged.■ JMSDeliveryTime - This WebLogic JMS-specific extended header field is logged.■ JMSRedeliveryLimit - This WebLogic JMS-specific extended header field is logged.■ foo - Any valid JMS header field or user property is logged. <p>When using multiple property values, enter them as a comma-separated list. The %header% and %properies% values are <i>not</i> case sensitive. For example, you can use "%header%, %properties%" for all the JMS header fields and user properties. However, the enumeration of individual JMS header fields and user properties are case sensitive. To enumerate only individual JMS header fields you could use "%header, name, address, city, state, zip".</p> <p>Note: The JMSMessageID field is always logged and cannot be turned off. Therefore, if the Expiration Policy is not defined (i.e., None) or is defined as an empty string, then the output to the log file contains only the JMSMessageID of the message.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSQueueMBean</p> <p><i>Attribute:</i> ExpirationLoggingPolicy</p>	<i>Dynamic:</i> yes





JMS Template --> Configuration --> Expiration Policy

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines the expiration policy that a topic or queue destination should take when an expired message is found: discard the message, discard the message and log its removal, or redirect the message to an error destination.

The configurable expiration policy attributes for a JMS template are the same as those configured for a destination. However, if the destination that is using a JMS template specifies an expiration policy value for an attribute, that expiration policy value is used.

Tasks

[“Creating a JMS Topic” on page 232-16](#)

[“Handling Expired Messages” on page 235-26](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Attribute Label	Description	Value Constraints
Expiration Policy	<p>An optional policy to use when expired messages are found on a topic or queue destination.</p> <p>None and Discard - Expired messages are removed from the messaging system. The removal is not logged and the message is not redirected to another location. If None is defined for a given destination, then expired messages are discarded.</p> <p>Log - Removes expired messages from the system and writes an entry to the server log file indicating that the messages have been removed from the system. The actual information that is logged is defined by the Expiration Logging Policy.</p> <p>Redirect - Moves expired messages from their current location to the Error Destination defined for the destination (on the Redelivery tab). The message retains its body, and all of its properties. The message also retains all of its header fields, but with the following exceptions:</p> <ul style="list-style-type: none"> ■ The destination for the message becomes the error destination. ■ All property overrides associated with the error destination are applied to the redirected message. ■ If there is no Time-To-Live Override value set for the Error Destination, then the message receives a new Expiration Time of zero (indicating that it will not expire again). <p>It is illegal to use the Redirect policy when there is no valid error destination defined for the destination. Similarly, it is illegal to remove the error destination for a destination using the Redirect policy.</p> <p>Note: The Maximum Message quota is only enforced for sending new messages. It is ignored when moving messages because of the Redirect policy.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.JMSTemplateMBean</code></p>	<p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ Discard ■ Log ■ Redirect <p><i>Dynamic:</i> yes</p>

Attribute Label	Description	Value Constraints
Expiration Logging Policy	<p>The policy that defines what information about the message is logged when the Expiration Policy on this destination is set to Log. The valid logging policy values are:</p> <ul style="list-style-type: none"> ■ %header% - All JMS header fields are logged. ■ %properties% - All user-defined properties are logged. ■ JMSDeliveryTime - This WebLogic JMS-specific extended header field is logged. ■ JMSRedeliveryLimit - This WebLogic JMS-specific extended header field is logged. ■ foo - Any valid JMS header field or user-defined property is logged. <p>When using multiple property values, enter them as a comma-separated list. The %header% and %properies% values are <i>not</i> case sensitive. For example, you can use "%header%, %properties%" for all the JMS header fields and user properties. However, the enumeration of individual JMS header fields and user-defined properties are case sensitive. To enumerate only individual JMS header fields you could use "%header, name, address, city, state, zip".</p> <p>Note: The JMSMessageID field is always logged and cannot be turned off. Therefore, if the Expiration Policy is not defined or is defined as an empty string, then the output to the log file contains only the JMSMessageID of the message.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTemplateMBean</p> <p><i>Attribute:</i> ExpirationLoggingPolicy</p>	<i>Dynamic:</i> yes





JMS Topic --> Configuration --> Expiration Policy

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines the action that a topic destination should take when an expired message is found: discard the message, discard the message and log its removal, or redirect the message to an error destination.

Note: You configure queues explicitly or by configuring a JMS template that can be used to define multiple queues with similar attribute settings, as described in [“JMS Template Tasks” on page 232-18](#).

Tasks

[“Creating a JMS Topic” on page 232-16](#)

[“Handling Expired Messages” on page 235-26](#)

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Managing WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Attributes

Table 231-1

Attribute Label	Description	Value Constraints
Expiration Policy	<p>An optional policy to use when expired messages are found on a topic destination.</p> <p>None - If a JMS template is specified for this topic, then <i>None</i> means that the policy is inherited from the template. Otherwise, <i>None</i> is equivalent to <i>Discard</i>.</p> <p>Discard - Removes expired messages from the messaging system. The removal is not logged and the message is not redirected to another location.</p> <p>Log - Removes expired messages from the system and writes an entry to the server log file indicating that the messages have been removed from the system. The actual information that is logged is defined by the Expiration Logging Policy.</p> <p>Redirect - Moves expired messages from their current location to the Error Destination defined for the destination. The message retains its body, and all of its properties. The message also retains all of its header fields, but with some exceptions:</p> <ul style="list-style-type: none">■ The destination for the message becomes the error destination.■ All property overrides associated with the error destination are applied to the redirected message.■ If there is no Time-To-Live Override value set for the error destination, then the message receives a new Expiration Time of zero (indicating that it will not expire again). <p>It is illegal to use the Redirect policy when there is no valid error destination defined for the destination. Similarly, it is illegal to remove the error destination for a destination using the Redirect policy.</p> <p>Note: The Maximum Message quota is only enforced for sending new messages. It is ignored when moving messages because of the Redirect policy.</p> <p>Administration Console Online Help</p> <p><i>MBean:</i> weblogic.management.configuration.</p>	<p><i>Valid values:</i></p> <ul style="list-style-type: none">■ Discard■ Log■ Redirect <p><i>Dynamic:</i> yes</p>

Table 231-1

Attribute Label	Description	Value Constraints
Expiration Logging Policy	<p>The policy that defines what information about the message is logged when the Expiration Policy on this topic is set to Log. The valid logging policy values are:</p> <ul style="list-style-type: none">■ %header% - All the JMS header fields are logged.■ %properties% - All the user properties are logged.■ JMSDeliveryTime - This WebLogic JMS-specific extended header field is logged.■ JMSRedeliveryLimit - This WebLogic JMS-specific extended header field is logged.■ foo - Any valid JMS header field or user property is logged. <p>When using multiple property values, enter them as a comma-separated list. The %header% and %properties% values are <i>not</i> case sensitive. For example, you can use "%header%, %properties%" for all the JMS header fields and user properties. However, the enumeration of individual JMS header fields and user properties are case sensitive. To enumerate only individual JMS header fields you could use "%header, name, address, city, state, zip".</p> <p>Note: The JMSMessageID field is always logged and cannot be turned off. Therefore, if the Expiration Policy is not defined (i.e., null) or is defined as an empty string, then the output to the log file contains only the JMSMessageID of the message.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSTopicMBean</p> <p><i>Attribute:</i> ExpirationLoggingPolicy</p>	<i>Dynamic:</i> yes





1 JMS: Configuring

[“Attributes and Console Screen Reference for JMS” on page 233-1]

The following sections explain how to configure the Java Message Service (JMS) for WebLogic Server:

- “JMS and WebLogic Server” on page 232-2
- “JMS Server Tasks” on page 232-5
- “JMS Connection Factory Tasks” on page 232-8
- “JMS Queue and Topic Destination Tasks” on page 232-14
- “JMS Distributed Destination Tasks” on page 232-31
- “JMS Template Tasks” on page 232-18
- “Destination Keys Tasks” on page 232-21
- “JMS Store Tasks” on page 232-22
- “Session Pools Tasks” on page 232-29
- “Connection Consumers Tasks” on page 232-30
- “Simple Access to Remote or Foreign JMS Providers” on page 232-48

You may also want to refer these WebLogic JMS and Messaging Bridge sections:

- “JMS: Tuning” on page 235-1
- “JMS: Monitoring” on page 234-1
- “Messaging Bridge” on page 289-1

JMS and WebLogic Server

JMS (Java Message Service) is a standard API for accessing enterprise messaging systems. Specifically, WebLogic JMS:

- Enables Java applications sharing a messaging system to exchange messages.
- Simplifies application development by providing a standard interface for creating, sending, and receiving messages.

The following figure illustrates WebLogic JMS messaging.



As illustrated in the figure, WebLogic JMS accepts messages from *producer* applications and delivers them to *consumer* applications.

Configuring WebLogic JMS

Using the Administration Console, you define configuration attributes to:

- Create JMS servers and target a WebLogic Server instance or a Migratable Target where the JMS server will be deployed, as describe in “JMS Server Tasks” on page 232-5.
- Create and/or customize values for JMS servers, connection factories, destinations (queues and topics), JMS templates, destination sort order (using destination keys), persistent stores (file or JDBC), paging stores, session pools, and connection consumers.
- Define message and/or bytes thresholds and quotas, as well as a maximum allowable message size on your JMS servers, destinations, and templates, as

discussed in “JMS Server Tasks” on page 232-5, “JMS Queue and Topic Destination Tasks” on page 232-14, and “JMS Template Tasks” on page 232-18.

- Enable any desired WebLogic JMS features, such as:
 - Server clustering using multiple connection factories, as described in “[Managing JMS](#)” in *Programming WebLogic JMS*.
 - High availability and load balancing for destinations across a cluster by using *distributed* destinations, as described in “JMS Distributed Destination Tasks” on page 232-31.
 - Persistent messages and durable subscribers, as described in “[Developing a WebLogic JMS Application](#)” in *Programming WebLogic JMS*.
 - Paging out message bodies during peak load periods to free up memory, as described in “Paging Out Messages To Free Up Memory” on page 235-8.
 - Controlling message flow during peak load periods, including blocking message producers, as described in “Controlling the Flow of Messages on JMS Servers and Destinations” on page 235-20.
 - Establishing a message expiration policy to ensure that expired messages are cleaned up immediately, as described in “Handling Expired Messages” on page 235-27.
 - Preventing message quota errors by temporarily blocking message producers from sending messages to a destination when it exceeds its specified maximum message quota, as described in “Avoiding Quota Exceptions by Blocking Message Producers” on page 235-24.
 - Concurrent message processing via session pools, as described in “Session Pools Tasks” on page 232-29.
- Messaging bridge for transferring messages (includes transactional messages) between any two JMS providers—including separate implementations of WebLogic JMS, as described in “Messaging Bridge”.

WebLogic JMS provides default values for some configuration attributes; you must provide values for all others. If you specify an invalid value for any configuration attribute, or if you fail to specify a value for an attribute for which a default does not exist, WebLogic Server will not boot JMS when you restart it.

A sample `examplesJMSServer` configuration is provided with the product in the Examples Server. For more information about starting the Examples Server, see “[Starting the Default, Examples, and Pet Store Servers](#)” in the *Installation Guide*.

There are also instructions for manually configuring a basic JMS implementation in the “[Starting WebLogic Server and Configuring JMS](#)” section of *Programming WebLogic JMS*.

When you port WebLogic JMS applications from a previous release of Weblogic Server, the configuration information is automatically converted, as described in “[Porting WebLogic JMS Applications](#)” in *Programming WebLogic JMS*.

Once WebLogic JMS is configured, applications can send and receive messages using the JMS API. For more information about developing WebLogic JMS applications, refer to “[Developing a WebLogic JMS Application](#)” in *Programming WebLogic JMS*.

Note: To assist with your WebLogic JMS configuration planning, *Programming WebLogic JMS* provides [configuration checklists](#) for the attribute requirements and/or options that support various JMS features.

JMS Configuration Naming Rules

Each server within a domain must have a name that is unique for all configuration objects in the domain. Within a domain, each server, machine, cluster, virtual host, and any other resource type must be named uniquely and must not use the same name as the domain. This unique naming rule also applies to all configurable JMS objects, such as JMS servers, stores, templates, connection factories, session pools, and connection consumers.

The one exception to this unique naming rule, however, is for JMS queue and topic destinations on different JMS servers in a domain, as follows:

- Queue destinations *can* use the same name as other queues on different JMS servers; topic destinations can also use the same name as other topics on different JMS servers.
- Queue destinations *cannot* use the same name with topic destinations, nor can queues nor topics use the same name as any other configurable objects.

JMS Server Tasks

A JMS server manages connections and message requests on behalf of clients. Use the Services →JMS →Server node to configure a JMS server and assign it to either an independent WebLogic Server instance or a migratable server target where it will be deployed.

Configuring a JMS Server

You must first configure a JMS server before you can configure any destinations or consumers.

1. Expand the JMS →Servers node. The JMS Servers table displays in the right pane showing all the JMS servers defined in your domain.
2. Click the Configure a new JMS Server text link. A dialog displays in the right pane showing the tabs associated with configuring a new JMS server.
3. On the Configuration General tab, define the general configuration attributes for a JMS server:
 - Enter a name for the JMS server. This name must be unique within the WebLogic Server instance or its cluster. For more information, see “JMS Configuration Naming Rules” on page 232-4.
 - Select an existing persistent store (disk-based file or JDBC-accessible database) for storing persistent messages. To select a store, you must first configure a JMS file store or a JDBC store using the JMS →Stores node. If you do not assign a persistent store for a JMS server, persistent messaging is not supported on that server.
 - Select a paging store where message bodies can be temporarily swapped out from memory when the JMS server’s message load reaches a specified threshold. To select a paging store, you must first configure a JMS file store using the JMS →Stores node. A paging store cannot be the same JMS file store used for storing persistent messages or durable subscribers. If you do not assign a paging store for a JMS server, paging is not supported on that server.

- Select an existing JMS template that is used to create all temporary destinations (temporary queues or topics).
- Specify an Expiration Scan Interval to define how long the JMS server will pause between its cycles of scanning local destinations for expired messages to be processed.

For more information about JMS server general attributes, see [“JMS Server --> Configuration --> General” on page 192-1](#).

4. Click Create to create a JMS 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 Destinations node and a Session Pools node are automatically added under the new server instance by default.
5. On the Configuration Thresholds & Quotas tab, define the following upper and lower message/byte threshold and maximum quota attributes for a JMS server:
 - Specify the maximum bytes or message quota that can be stored in a JMS server.
 - Specify the upper threshold value that trigger events based on the number of bytes or messages stored in a JMS server. Events include message paging, message flow control, and system log messages.
 - Specify the lower threshold value that trigger events based on the number of bytes or messages stored in a JMS server. Events include message paging, message flow control, and system log messages.
 - Indicate whether bytes or messages paging is enabled on a JMS server for temporarily swapping messages out from memory to a paging store when a JMS server’s message load reaches a specified bytes/messages threshold.
 - Specify the maximum size of a message that will be accepted from message producers on a JMS server. The size includes the message body, any user-defined properties, and the user-defined JMS header fields: JMSCorrelationID and JMSType.
 - Select the server’s policy on whether to deliver smaller messages before larger ones when a destination has exceeded its message quota.

For more information about the Thresholds & Quota attributes, see [“JMS Server --> Configuration --> Thresholds & Quotas” on page 193-1](#).

6. Click Apply to save your changes.

7. On the Target and Deploy tab, select an independent WebLogic Server or a migratable server target on which to deploy the JMS server.

For more information see, “Targeting and Deploying a JMS Server” on page 232-7.

8. Click Apply to target the JMS server.

Targeting and Deploying a JMS Server

You can assign a JMS server to either an independent WebLogic Server instance or to a migratable target server where it will be deployed. Whereas, a connection factory or template can be instantiated on multiple WebLogic Server instances simultaneously

1. Under the JMS →Servers node in the left pane, click the node for the JMS server instance that you want to assign. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Target and Deploy tab.
3. Complete one of the following steps for targeting either an independent server or a migratable target server:
 - From the Target list box, select the server target where you want to deploy the JMS server. When a target WebLogic Server boots, the JMS server boots as well. If no target WebLogic Server is specified, the JMS server will not boot. The deployment of a JMS server differs from that of a connection factory or template.
 - From the Migratable Target list box, select the migratable target where you want to deploy the JMS server. Migratable targets define a set of WebLogic Server instances in a cluster that can potentially host an *exactly-once* service, such as JMS. When a migratable target server boots, the JMS server boots as well on the *user-preferred* server in the cluster. However, a JMS server and all of its destinations can migrate to another server within the cluster in response to a server failure or due to a scheduled migration for system maintenance.

For more information on configuring a migratable target for JMS, see “[Managing JMS](#)” in *Programming WebLogic JMS*.

4. Click Apply to save your assignments.

Monitoring a JMS Server

On the JMS Monitoring tab, you can monitor statistics for view run-time information for active JMS servers, destinations, and server session pools.

1. Expand the JMS node.
2. Click the Servers node. The JMS Servers information displays in the right pane showing all the JMS servers defined in your domain.
3. Click the JMS server that you want to monitor from the JMS server list, or from the JMS Servers table displayed in the right pane.
4. Click the Monitoring tab to display the monitoring links for monitoring JMS server data:
 - Monitor all Active JMS Servers — A table displays showing all instances of the JMS server deployed across the WebLogic Server domain.
 - Monitor all Active JMS Destinations — A table displays showing all active JMS destinations for the current domain.
 - Monitor all Active JMS Session Pool Runtimes — A table displays showing all active JMS session pools for the current domain.

For more information about monitoring JMS objects, see “JMS: Monitoring” on page 234-1.

JMS Connection Factory Tasks

Connection factories are objects that enable JMS clients to create JMS connections. A connection factory supports concurrent use, enabling multiple threads to access the object simultaneously. WebLogic JMS provides preconfigured “default connection factories” that can be enabled or disabled on a per-server basis, as described in “Using a Default Connection Factory” on page 232-9. Otherwise, you can configure one or more connection factories to create connections with predefined attributes that better suit your application — as long as each connection factory is uniquely named. WebLogic Server adds them to the JNDI space during startup, and the application then retrieves a connection factory using WebLogic JNDI.

You can establish cluster-wide, transparent access to JMS destinations from any server in the cluster, either by using the default connection factories for each server instance, or by configuring one or more connection factories and targeting them to one or more server instances in the cluster. This way, each connection factory can be deployed on multiple WebLogic Servers. For more information on configuring JMS clustering, see [“Managing JMS”](#) in *Programming WebLogic JMS*.

Using a Default Connection Factory

WebLogic JMS defines two default connection factories, which can be looked up using the following JNDI names:

- `weblogic.jms.ConnectionFactory`
- `weblogic.jms.XAConnectionFactory`

You only need to configure a new connection factory if the pre-configured settings of the default factories are not suitable for your application. The main difference between the pre-configured settings for the default connection factories and a user-defined connection factory is the default value for the “XA Connection Factory Enabled” attribute to enable JTA transactions, as shown in the following table:

Table 232-2 Default Connection Factory Settings for Transacted Sessions (XA)

<i>Default Connection Factory. . .</i>	<i>XAConnectionFactoryEnabled setting is. . .</i>
<code>weblogic.jms.ConnectionFactory</code>	False
<code>weblogic.jms.XAConnectionFactory</code>	True

An XA factory is required for JMS applications to use JTA user-transactions, but is not required for transacted sessions. For more information about using transactions with WebLogic JMS, see [“Using Transactions with WebLogic JMS”](#) in *Programming WebLogic JMS*.

All other default factory configuration attributes are set to the same default values as a user-defined factory. For more information about the XA Connection Factory Enabled attribute, and to see the default values for the other connection factory attributes, see [“Attributes and Console Screen Reference for JMS”](#) on page 233-1.

Another distinction when using the default connection factories is that you have no control over targeting the WebLogic Server instances where the connection factory may be deployed. However, you can enable and/or disable the default connection factories on a per-WebLogic Server basis.

- For more information on enabling or disabling the default connection factories, see “Server --> Services --> JMS” on page 454-1.
- For information on deploying a user-defined connection factory on independent server instances, on specific servers within a cluster, or on an entire cluster, see “Deploying a Connection Factory on Multiple Individual Servers” on page 232-12 and “Deploying a Connection Factory on a Cluster” on page 232-13.

Note: Some connection factory attributes are dynamically configurable. When dynamic attributes are modified at run time, the new values become effective for new connections only, and do not affect the behavior of existing connections.

Configuring a JMS Connection Factory

Using the Services →JMS →Connection Factories node, you can configure one or more connection factories to create connections with predefined attributes.

1. Expand the JMS →Connection Factories node. The JMS Connection Factories table displays in the right pane showing all the connection factories defined in your domain.
2. 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.
3. On the Configuration General tab, define the general configuration attributes for the connection factory.
 - Enter a name for the connection factory. This name must be unique within the WebLogic Server instance or its cluster. For more information, see “JMS Configuration Naming Rules” on page 232-4.
 - Enter a JNDI name for accessing the connection factory within the JNDI namespace.

- Specify a client identifier (client ID) for clients with durable subscribers. For more information about durable subscribers, see “[Developing a WebLogic JMS Application](#)” in *Programming WebLogic JMS*.
- Define the default message delivery attributes: Priority, Time To Live, Time To Deliver, and the Delivery Mode (Persistent or Non-Persistent).
- Specify the maximum number of outstanding messages that may exist for an asynchronous session and the overrun policy (that is, the action to be taken, for multicast sessions, when this maximum is reached).
- Indicate whether the `close()` method is allowed to be called from the `onMessage()` method.
- Choose whether all messages are acknowledged or only previously received messages are acknowledged.
- For distributed destinations, determine whether non-anonymous producers created through a connection factory are load balanced on a per-call basis and whether server affinity is used when load balancing consumers or producers in a distributed destination.

For more information about general connection factory attributes, see “[JMS Connection Factory --> Configuration --> General](#)” on page 155-1.

4. 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.
5. On the Configuration Transactions tab, define a value for the transaction time-out attribute and use the `XAConnectionFactoryEnabled` field to indicate whether a transaction queue or topic connection factory is returned and whether the connection factory creates sessions that are JTA aware.

For more information about connection factory transaction attributes, see “[JMS Connection Factory --> Configuration --> Transactions](#)” on page 156-1.

6. Click Apply to save your changes.
7. On the Configuration Flow Control tab, define values that instruct a message producer to adjust its message flow. Specifically, the producer receives attributes that limit its flow within a minimum and maximum range. As conditions worsen, the producer moves toward the minimum; as conditions improve; the producer

moves toward the maximum. Use the Send Timeout attribute to specify the maximum time that a producers will wait for sufficient quota on a JMS server and destination to accommodate the message that it is attempting to send.

For more information about connection factory flow control attributes, see [“JMS Connection Factory --> Configuration --> Flow Control” on page 154-1](#).

8. Click Apply to save your changes.
9. On the Target and Deploy tab, assign the connection factory to a WebLogic Server instance or to a server cluster. Targets enable you to limit the set of servers, groups, and/or clusters on which a connection factory may be deployed.

For more information see, “Deploying a Connection Factory on Multiple Individual Servers” on page 232-12 and “Deploying a Connection Factory on a Cluster” on page 232-13.

10. Click Apply to save your changes.

Deploying a Connection Factory on Multiple Individual Servers

You can establish cluster-wide, transparent access to JMS destinations from any server in a domain by deploying one or more connection factories on multiple WebLogic Server instances simultaneously.

1. Expand the JMS → Connection Factories node, and then select the connection factory that you want to deploy. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Target and Deploy tab. All the server instances in the domain are listed in the Independent Servers box. A check mark is displayed next to each server already targeted to the connection factory.
3. To deploy the connection factory on one or more servers in the list, select the check mark displayed next to each server name.
4. Connection Factories can also be undeployed from individual servers whenever necessary by clearing the check mark displayed next to each server name.
5. Click Apply to save your assignments.

Deploying a Connection Factory on a Cluster

In a clustered environment, you can establish cluster-wide, transparent access to JMS destinations from any server in a cluster by deploying a connection factory on all server instances in the cluster, or even to specific servers within the cluster.

1. Expand the JMS → Connection Factories node, and then select the connection factory that you want to deploy. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Target and Deploy tab. All the clusters configured in the domain are listed in the Clusters box. A check mark is displayed next to each cluster name already targeted to the connection factory.
3. The Connection factory can be targeted to either all the servers in a cluster or to selected servers within a cluster:
 - To deploy the connection factory on all the servers in a cluster, click the “All servers in the cluster” button under the cluster name.
 - To deploy the connection factory on only selected servers within a cluster, click the “Some servers in the cluster” button under the cluster name. Then target one or more servers in the cluster by selecting the check mark displayed next to each server name.
4. Connection Factories can also be undeployed from an entire cluster or from selected servers within a cluster whenever necessary, by doing the following:
 - To undeploy a connection factory from all the servers in a cluster, clear the check box next to the cluster name.
 - To undeploy a connection factory from only selected servers in a cluster, click the “Some servers in the cluster” button under the cluster name. Then undeploy one or more servers in the cluster by clearing the check mark displayed next to each server name.
5. Click Apply to save your assignments.

JMS Queue and Topic Destination Tasks

A JMS destination identifies a queue (point-to-point) or topic (publish/subscribe) for a JMS server. After configuring a JMS server, configure one or more queue or topic destinations for each JMS server.

You configure destinations explicitly or by configuring a destination template that can be used to define multiple destinations with similar attribute settings, as described in “JMS Template Tasks” on page 232-18.

You can configure multiple physical destinations as members of a single distributed destination set within a cluster. Therefore, if one WebLogic Server instance within the cluster fails, then the other instances hosting the same distributed destination will continue to provide service to JMS producers and consumers. For more information, see “JMS Distributed Destination Tasks” on page 232-31.

Note: To help manage recovered or rolled back messages, you can also configure a target error destination for messages that have reached their redelivery limit. The error destination must be a destination that is configured on the local JMS server. For more information, see “[Configuring an Error Destination for Undelivered Messages](#)” in *Programming WebLogic JMS*.

Some destination attributes are dynamically configurable. When attributes are modified at run time, only incoming messages are affected; stored messages are not affected.

Creating a JMS Queue

A JMS queue defines a *point-to-point* destination type for a JMS server. After defining a JMS server, configure one or more queue destinations for each JMS server.

1. Expand the JMS →Servers node and select a JMS server instance.
2. Click the Destinations node. The JMS Destinations table displays in the right pane showing all the JMS queues.
3. Click the Configure a new JMS Queue text link. A dialog shows the tabs associated with configuring a new queue.

4. On the Configuration General tab, define the general configuration attributes for the queue.
 - Enter a name for the queue destination. This name must be unique within the WebLogic Server instance or its cluster. For more information, see “JMS Configuration Naming Rules” on page 232-4.
 - Enter a JNDI name for accessing the queue within the JNDI namespace.
 - Indicate whether the JNDI name for the queue (if specified) is replicated across the cluster.
 - Specify whether the queue supports persistent messaging by using the JMS store specified for the JMS server.
 - Select an existing JMS template if you are using one to create queues.
 - Select existing destination keys that will define the sort order for messages arriving on the queue.

For more information about the queue general attributes, see [“JMS Queue --> Configuration --> General” on page 186-1](#).

5. Click Create to create a queue instance with the name you specified in the Name field. The new instance is added under the Destinations node in the left pane.
6. On the Configuration Thresholds & Quotas tab, define the following upper and lower message/byte threshold and maximum quota attributes for the queue:
 - Specify the maximum bytes or message quota that can be stored in a queue.
 - Specify the upper threshold value that trigger events based on the number of bytes or messages stored in a queue. Events include message paging, message flow control, and system log messages.
 - Specify the lower threshold value that trigger events based on the number of bytes or messages stored in a queue. Events include message paging, message flow control, and system log messages.
 - Indicate whether bytes or messages paging is enabled on a queue for temporarily swapping messages out from memory to a paging store when a queue’s message load reaches a specified bytes/messages threshold.
 - Specify the maximum size of a message that will be accepted from message producers on a queue. The size includes the message body, any user-defined properties, and the user-defined JMS header fields: JMSCorrelationID and JMSType.

For more information about the Thresholds & Quota attributes, see [“JMS Queue --> Configuration --> Thresholds & Quotas” on page 189-1.](#)

7. On the Configuration Overrides tab, define the message attributes that can override those specified by a message producer, including the priority, time-to-live, time-to-deliver, and delivery mode.

For more information about the queue override attributes, see [“JMS Queue --> Configuration --> Overrides” on page 187-1.](#)

8. On the Configuration Redelivery tab, define the message redelivery attributes, including redelivery delay override, redelivery limit, and error destination.

For more information about the queue redelivery attributes, see [“JMS Queue --> Configuration --> Redelivery” on page 188-1.](#)

9. On the Configuration Expiration Policy tab, define the message expiration policy to use when expired messages are encountered on the queue.

For more information about the queue expiration policy attributes, see [“JMS Queue --> Configuration --> Expiration Policy” on page 229-1.](#)

10. Click Apply to save any changes you made on these tabs.

Creating a JMS Topic

A JMS topic identifies a *publish/subscribe* destination type for a JMS server. After defining a JMS server, configure one or more topic destinations for each JMS server.

1. Expand the JMS →Servers node and select a JMS server instance.
2. Click the Destinations node. The JMS Destinations table displays in the right pane showing any configured JMS topics.
3. Click the Configure a new JMS Topic text link. A dialog shows the tabs associated with configuring a new topic.
4. On the Configuration General tab, define the general configuration attributes for the topic.
 - Enter a name for the topic destination. This name must be unique within the WebLogic Server instance or its cluster. For more information, see [“JMS Configuration Naming Rules” on page 232-4.](#)

- Enter a JNDI name for accessing the topic within the JNDI namespace.
- Indicate whether the JNDI name for the topic (if specified) is replicated across the cluster.
- Specify whether the topic supports persistent messaging by using the JMS store specified for the JMS server.
- Select an existing JMS template if you are using one to create topics.
- Select existing destination keys that will define the sort order for messages arriving on the topic.

For more information about the topic general attributes, see [“JMS Topic --> Configuration --> General” on page 211-1](#).

5. Click Create to create a topic instance with the name you specified in the Name field. The new instance is added under the Destinations node in the left pane.
6. On the Configuration Thresholds & Quotas tab, define the following upper and lower message/byte threshold and maximum quota attributes for the topic:
 - Specify the maximum bytes or message quota that can be stored in a topic.
 - Specify the upper threshold value that trigger events based on the number of bytes or messages stored in a topic. Events include message paging, message flow control, and system log messages.
 - Specify the lower threshold value that trigger events based on the number of bytes or messages stored in a topic. Events include message paging, message flow control, and system log messages.
 - Indicate whether bytes or messages paging is enabled on a topic for temporarily swapping messages out from memory to a paging store when a topic's message load reaches a specified bytes/messages threshold.
 - Specify the maximum size of a message that will be accepted from message producers on a topic. The size includes the message body, any user-defined properties, and the user-defined JMS header fields: JMSCorrelationID and JMSType.

For more information about the topic Thresholds & Quota attributes, see [“JMS Topic --> Configuration --> Thresholds & Quotas” on page 215-1](#).

7. On the Configuration Overrides tab, define the message attributes that can override those specified by a message producer, including the priority, time-to-live, time-to-deliver, and delivery mode.

For more information about the topic override attributes, see [“JMS Topic --> Configuration --> Overrides” on page 213-1](#).

8. On the Configuration Redelivery tab, define the message redelivery attributes, including redelivery delay override, redelivery limit, and error destination.

For more information about the topic redelivery attributes, see [“JMS Topic --> Configuration --> Redelivery” on page 214-1](#).

9. On the Configuration Expiration Policy tab, define the message expiration policy logging properties to use when expired messages are encountered on the topic.

For more information about the topic expiration policy attributes, see [“JMS Topic --> Configuration --> Expiration Policy” on page 231-1](#).

10. On the Configuration Multicast tab, define the multicast attributes for the topic, including a multicast address, time-to-live (TTL), and port.

For more information about the topic expiration policy attributes, see [“JMS Topic --> Configuration --> Multicast” on page 212-1](#).

11. Click Apply to save any changes you made on these tabs.

JMS Template Tasks

A JMS template provides an efficient means of defining multiple destinations with similar attribute settings. JMS templates offer the following benefits:

- You do not need to re-enter every attribute setting each time you define a new destination; you can use the JMS template and override any setting to which you want to assign a new value.
- You can modify shared attribute settings dynamically simply by modifying the template.

The configurable attributes for a JMS template are the same as those configured for a destination. These configuration attributes are inherited by the destinations that use them, with the following exceptions:

- If the destination that is using a JMS template specifies an override value for an attribute, the override value is used.

- If the destination that is using a JMS template specifies a message redelivery value for an attribute, that redelivery value is used.
- The Name attribute is not inherited by the destination. This name is valid for the JMS template only. You must explicitly define a unique name for all destinations. For more information, see “JMS Configuration Naming Rules” on page 232-4.
- The JNDI Name, Enable Store, and Template attributes are not defined for JMS templates.
- The Multicast attributes are not defined for JMS templates because they apply only to topic destinations.

Any attributes that are not explicitly defined for a destination are assigned default values. If no default value exists, be sure to specify a value within the JMS template or as a destination attribute override. If you do not do so, the configuration information remains incomplete, the WebLogic JMS configuration fails, and the WebLogic JMS does not boot.

Creating a JMS Template

To define the JMS template configuration attributes for destinations, use the JMS → Templates node.

1. Expand the JMS → Templates node. The JMS Templates table displays in the right pane showing all the templates defined in the domain.
2. 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.
3. On the Configuration General tab, define the general configuration attributes for the JMS template.
 - Name of the JMS template. This name must be unique within the WebLogic Server instance or its cluster. For more information, see “JMS Configuration Naming Rules” on page 232-4.
 - Destination keys used to define the sort order for messages arriving on destinations created from this JMS template.

For more information about the topic general attributes, see [“JMS Template --> Configuration --> General” on page 205-1](#).

4. Click Create to create a template instance with the name you specified in the Name field. The new instance is added under the Templates node in the left pane.
5. On the Configuration Thresholds & Quotas tab, define the following upper and lower message/byte threshold and maximum quota attributes for the destinations created from this JMS template:
 - Specify the maximum bytes or message quota that can be stored in a destination.
 - Specify the upper threshold value that trigger events based on the number of bytes or messages stored in a destination. Events include message paging, message flow control, and system log messages.
 - Specify the lower threshold value that trigger events based on the number of bytes or messages stored in a destination. Events include message paging, message flow control, and system log messages.
 - Indicate whether bytes or messages paging is enabled on a destination for temporarily swapping messages out from memory to a paging store when a destination’s message load reaches a specified bytes/messages threshold.
 - Specify the maximum size of a message that will be accepted from message producers on a destination. The size includes the message body, any user-defined properties, and the user-defined JMS header fields: JMSCorrelationID and JMSType.

For more information about the JMS template Thresholds & Quota attributes, see [“JMS Template --> Configuration --> Thresholds & Quotas” on page 208-1](#).

6. On the Configuration Override tab, define the message attributes that can override those specified by a message producer for destinations created from this JMS template, including the priority, time-to-live, time-to-deliver, and delivery mode.

For more information about the JMS template override attributes, see [“JMS Template --> Configuration --> Override” on page 206-1](#).

7. On the Configuration Redelivery tab, define the message redelivery attributes for destinations created from this JMS template, including redelivery delay override, redelivery limit, and error destination.

For more information about the JMS template redelivery attributes, see [“JMS Template --> Configuration --> Redelivery” on page 207-1](#).

8. On the Configuration Expiration Policy tab, define the message expiration policy logging properties to use when expired messages are encountered on destinations created from this JMS template.

For more information about the JMS template expiration policy attributes, see [“JMS Template --> Configuration --> Expiration Policy” on page 230-1](#).

9. Click Apply to save any changes you made on these tabs.

Destination Keys Tasks

Use destination keys to define the sort order for messages that arrive on a specific destination.

Creating a JMS Destination Key

To create a destination key, use the Destination Keys node.

1. Expand the JMS →Destination Keys node. The JMS Destinations Keys table displays in the right pane showing all the destination keys.
2. Click the Create a new JMS Destination Key text link. A dialog shows the tabs associated with configuring a new destination key.
3. On the Configuration General tab, define the general destination key attributes:
 - Enter a name of the destination key.
 - Select a message sort key name or the name of a message header field on which to sort. Message header field keys start with the letters JMS and ignore the key type setting. (For better performance, use message header fields as sorting keys, rather than message properties.)
 - Select the expected property type for the sort key. (This setting is ignored for message header field keys, which have an implied type.)

- Select the direction in which the key will sort messages (ascending or descending).

For more information about the JMS destination key attributes, see [“JMS Destination Key --> Configuration” on page 161-1](#).

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

JMS Store Tasks

WebLogic JMS can store persistent messages in either a JDBC-accessible database or a disk-based file.

The following are some similarities and differences between file stores and JDBC stores.

- Both have the same transaction semantics and guarantees. As with JDBC store writes, file store writes are guaranteed to be persisted to disk and are not simply left in an intermediate (unsafe) cache.
- Both have the same application interface (no difference in application code).
- File stores are generally much faster.
- File stores are easier to configure.
- File stores generate no network traffic; JDBC stores will generate network traffic if the database is on a different machine from the JMS server.
- File stores are much better suited to paging non-persistent messages.
- JDBC stores may make it easier to handle failure recovery since the JDBC interface can access the database from any machine on the same network; with the file store, the disk must be shared or migrated.
- You cannot configure a transaction (XA) connection pool with a JMS JDBC store. For more information, see [“Handling Transactions with JMS JDBC Stores” on page 232-28](#).

JMS stores can increase the amount of memory required during initialization of a WebLogic Server instance as the number of stored messages increases. If initialization fails due to insufficient memory while rebooting WebLogic Server, increase the heap size of the Java Virtual Machine (JVM) proportionally to the number of messages that are currently stored in the JMS store. Then, try rebooting the server again. For more information on setting heap sizes, see “[Tuning WebLogic Server Applications](#)” in the *WebLogic Performance and Tuning Guide*.

JMS File Store Tasks

A JMS file store consists of a disk-based file that is used for storing persistent messages and durable subscribers on a local file system. It is also the recommended store for temporarily paging messages to disk when memory has been exhausted.

Notes: In order for a JMS file store to handle failure recovery, the disk must be shared or migrated. Therefore, it is highly recommended that you implement a hardware solution, such as a dual-ported SCSI disk or Storage Area Network (SAN) to make your file store available from other machines.

Also, you should avoid using a Network File System (NFS) solution to access JMS file stores due to synchronicity limitations and performance issues.

Creating a JMS File Store

A disk-based JMS file store stores persistent messages and durable subscribers in a file-system directory. This directory must exist on your file system, so be sure to create it before completing this tab.

1. Expand the JMS →Stores node. The JMS Stores table displays in the right pane showing all the JMS stores.
2. Click the Create a new JMS File Store text link. A dialog shows the tabs associated with configuring a new file store.
3. On the Configuration tab, define the general file store attributes:
 - Enter a name for the file store. This name must be unique within the WebLogic Server instance or its cluster. For more information, see “JMS Configuration Naming Rules” on page 232-4.

- Select a Synchronous Write policy to determine how this JMS file store writes data to disk. This policy also affects the JMS file store's performance, scalability, and reliability. For more information see, "Improving JMS File Store Performance" on page 235-2.

Note: If the JMS file store is used exclusively for paging out messages to disk, the Synchronous Write Policy is ignored.

- Enter the pathname to the directory on the file system where the JMS file store is kept. (This directory must exist on your system, so be sure to create it before completing this tab.)

For more information about the JMS file store attributes, see ["JMS File Store --> Configuration" on page 181-1.](#)

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

Configuring a Paging Store

A JMS file store is the recommended store type when configuring message paging for the JMS server and destinations. A paging store cannot be the same JMS file store used for storing persistent messages or durable subscribers for a JMS server. Therefore, you need to configure an additional JMS file store to be used exclusively for message paging for each JMS server.

For more information on configuring paging for a JMS server, see ["Paging Out Messages To Free Up Memory" on page 235-8.](#)

1. Expand the JMS →Stores node. The JMS Stores table displays in the right pane showing all the JMS stores.
2. Click the Create a new JMS File Store text link. A dialog shows the tabs associated with configuring a new file store.
3. On the Configuration tab, define the general file store attributes:
 - Enter a name for the file store. This name must be unique within the WebLogic Server instance or its cluster (for example, "JMSPagingStore"). For more information, see "JMS Configuration Naming Rules" on page 232-4.
 - The Synchronous Write policy is ignored when the JMS file store is used exclusively for paging out messages to disk.

- Enter the pathname to the directory on the file system where the JMS file store is kept. (This directory must exist on your system, so be sure to create it before completing this tab.)

For more information about the JMS file store attributes, see [“JMS File Store --> Configuration” on page 181-1](#).

4. Click Create to create a paging store instance with the name you specified in the Name field. The new instance is added under the Stores node in the left pane.
5. Make sure to select this “paging” file store on the JMS Server Configuration General tab.

JMS JDBC Store Tasks

Through the use of JDBC, JMS enables you to store persistent messages in a database, which is accessed through a designated JDBC connection pool. The JMS database can be any database that is accessible through a JDBC driver. WebLogic JMS detects some drivers for the following databases:

- Pointbase
- Microsoft SQL (MSSQL) Server
- Oracle
- Sybase
- Cloudscape
- Informix
- IBM DB2
- Times Ten

The `weblogic/jms/ddl` directory within the `weblogic.jar` file contains JMS DDL files for these databases, which are actually text files containing the SQL commands that create the JMS database tables. To use a different database, simply copy and edit any one of these `.ddl` files.

Note: The JMS samples provided with your WebLogic Server distribution are set up to work with the Pointbase Java database. An evaluation version of Pointbase is included with WebLogic Server and a *demoPool* database is provided.

If your existing JMS JDBC stores somehow become corrupted, you can regenerate them using the `utils.Schema` utility. For more information see, “[JDBC Database Utility](#)” in *Programming WebLogic JMS*.

Creating a JMS JDBC Store

1. Expand the JMS →Stores node. The JMS Stores table displays in the right pane showing all the JMS stores.
2. Click the Create a new JMS JDBC Store text link. A dialog shows the tabs associated with configuring a new JDBC store.
3. On the Configuration tab, define the general JMS JDBC store attributes:
 - Enter a name for the JDBC-accessible database store. This name must be unique within the WebLogic Server instance or its cluster. For more information, see “JMS Configuration Naming Rules” on page 232-4.
 - Select an existing JDBC connection pool that will be used to access the JMS JDBC store. For more information see,
 - Enter the prefix name that will be prepended to the JMS table names in this JMS JDBC store for use with multiple instances.

For more information about the JMS JDBC store attributes, see “[JMS JDBC Store --> Configuration](#)” on page 183-1.

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

Using Prefixes with JMS JDBC Stores

The JMS database contains two system tables that are generated automatically and are used internally by JMS:

- `<prefix>JMSStore`
- `<prefix>JMSState`

When a JMS JDBC store is used, WebLogic Server JMS may spend a significant amount of time scanning all database tables to find its own tables. If, however, the name of the store includes a unique prefix that includes the schema name, this search time can be reduced, thus improving boot performance.

For this reason we recommend adding a unique prefix to the store name when configuring a JMS JDBC store to identify JMS tables in the store. The prefix may be any string, but in many databases, the user name is used as the schema name. A prefix is prepended to table names when the DBMS requires fully qualified names, or when you must differentiate between JMS tables for two WebLogic Servers, enabling multiple tables to be stored on a single DBMS.

Warning: No two JMS stores should be allowed to use the same database tables, as this will result in data corruption.

Specify the prefix using the following format, which will result in a valid table name when prepended to the JMS table name:

```
[[[catalog.]schema.]prefix]JMSStore
```

where *catalog* identifies the set of system tables being referenced by the DBMS and *schema* translates to the ID of the table owner. For example, in a production database the JMS administrator could maintain a unique table for the Sales department, as follows:

```
[[[Production. ]JMSAdmin. ]Sales]JMSStore
```

Note: For some DBMS vendors, such as Oracle, there is no catalog to set or choose, so this format simplifies to `[[schema.]prefix]`. For more information, refer to your DBMS documentation for instructions on how to write and use a fully-qualified table name.

Recommended JDBC Connection Pool Settings for JMS JDBC Stores

The following settings are recommended when using a JDBC connection pool for JMS JDBC stores.

Automatic Reconnection to Failed Databases

WebLogic Server provides robust JDBC connection pools that can automatically reconnect to failed databases after they come back online, without requiring you to restart WebLogic Server. To take advantage of this capability, and make your use of JMS JDBC stores more robust, configure the following attributes on the JDBC connection pool associated with the JMS JDBC store:

```
TestConnectionsOnReserve="true"
TestTableName="[ [catalog.]schema.]prefix]JMSState"
```

Required Setting for WebLogic Type 4 JDBC DB2 Drivers

For connection pools used as a JMS JDBC store that use the WebLogic Type 4 JDBC driver for DB2, the `BatchPerformanceWorkaround` property must be set to “true” due to internal JMS batching requirements.

For more information, see the [“Performance Workaround for Batch Inserts and Updates”](#) section in the *WebLogic Type 4 JDBC Drivers* documentation.

Handling Transactions with JMS JDBC Stores

You cannot configure a transaction (XA) JDBC connection pool or JDBC `TxDataSource` to use with a JMS JDBC store. JMS must use a JDBC connection pool that uses a non-`TxDataSource` with a non-`XAResource` driver (you cannot use an XA driver or a JTS driver, or implement the “Emulate Two-Phase Commit for non-XA Driver” option). WebLogic JMS does the XA support above the JDBC driver.

This is because WebLogic JMS is its own resource manager. That is, JMS itself implements the `XAResource` and handles the transactions without depending on the database (even when the messages are stored in the database). This means that whenever you are using JMS and a database (even if it is the same database where the JMS messages are stored), then it is two-phase commit transaction. For more information about using transactions with WebLogic JMS, see [“Using Transactions with WebLogic JMS”](#) in *Programming WebLogic JMS*.

From a performance perspective, you may boost your performance if the JDBC connection pool used for the database work exists on the same WebLogic Server as the JMS queue—the transaction will still be two-phase, but it will be handled with less network overhead. Another performance boost might be achieved by using JMS file stores rather than JMS JDBC stores.

Session Pools Tasks

Server session pools enable an application to process messages concurrently. After you define a JMS server, optionally, configure one or more session pools for each JMS server. Some session pool attributes are dynamically configurable, but the new values do not take effect until the session pools are restarted.

Note: Session pools are now used rarely, as they are not a required part of the J2EE specification, do not support JTA user transactions, and are largely superseded by message-driven beans (MDBs), which are a required part of the J2EE specification.

For more information about creating session pools, see [“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Creating a JMS Session Pool

Use the Session Pools node in the Administration Console and define the following configuration attributes:

1. Expand the JMS →Servers node.
2. Expand a JMS server instance under Servers.
3. Click the Session Pools node. The Session Pools table displays in the right pane showing all the session pools.
4. Click the Configure a new JMS Session Pool text link. A dialog shows the tabs associated with configuring a new session pool.
5. On the Configuration tab, define the general session pool attributes:
 - Enter a name for the server session pool. This name must be unique within the WebLogic Server instance or its cluster. For more information, see “JMS Configuration Naming Rules” on page 232-4.
 - Enter the connection factory name with which the server session pool is associated and is used to create sessions.

- Enter the message listener class used to receive and process messages concurrently.
- Select the acknowledge mode used by non-transacted sessions within this JMS session pool.
- Specify the maximum number of concurrent sessions.
- Select whether the session pool creates transacted sessions.

For more information about the session pool attributes, see [“JMS Session Pool --> Configuration” on page 199-1](#).

6. 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.

Connection Consumers Tasks

Connection consumers are queues (Point-To-Point) or topics (Pub/Sub) that retrieve server sessions and process messages. After you define a session pool, configure one or more connection consumers for each session pool.

For more information about creating connection consumers, see [“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

Creating a JMS Connection Consumer

To configure connection consumers, use the Session Pools node to define the following configuration attributes:

1. Expand the JMS →Servers node.
2. Expand a JMS server instance under Servers.
3. Expand the Session Pools node, and then click a session pool instance.
4. Click the Consumers node. The JMS Consumers table displays in the right pane showing all the connection consumers.

5. Click the Configure a new JMS Connection Consumer text link. A dialog shows the tabs associated with configuring a new connection consumer.
6. On the Configuration tab, define the general connection consumer attributes:
 - Enter a name for the connection consumer. This name must be unique within the WebLogic Server instance or its cluster. For more information, see “JMS Configuration Naming Rules” on page 232-4.
 - Specify the maximum number of messages that can be accumulated by the connection consumer.
 - Specify the JMS selector expression used to filter messages. For information about defining selectors, see [Developing a WebLogic JMS Application](#) in *Programming WebLogic JMS*.
 - Specify the destination on which the connection consumer will listen.

For more information about the connection consumer attributes, see [“JMS Connection Consumer --> Configuration” on page 151-1](#).

7. Click Create to create a connection consumer instance with the name you specified in the Name field. The new instance is added under the Consumers node in the left pane.

JMS Distributed Destination Tasks

A *distributed* destination is a set of physical destinations (queues or topics) that are called under a single JNDI name so they appear to be a single, logical destination to a client, when the members of the set are actually distributed across multiple servers within a cluster, with each destination member belonging to a separate JMS server.

By enabling you to configure multiple physical queues and topics as members of a distributed destination, WebLogic JMS supports high availability and load balancing of the JMS destinations within a cluster. For more information about using a distributed destination with your applications, see [“Using Distributed Destinations”](#) in *Programming WebLogic JMS*.

Guidelines for Configuring Distributed Destinations

You configure distributed JMS destinations through the Services → JMS → Distributed Destinations node. To facilitate the configuration process, these instructions are divided into procedures that address the following scenarios:

- New implementations of WebLogic JMS with no physical destinations *or* existing configurations of WebLogic JMS that do not require previously configured destinations to be part of a distributed destination:
 - [Creating a Distributed Topic and Creating Members Automatically](#)
 - [Creating a Distributed Queue and Creating Members Automatically](#)
 - Existing implementations of WebLogic JMS that require previously configured destinations to be members of a distributed destination set:
 - [Creating a Distributed Topic and Adding Existing Physical Topics as Members Manually](#)
 - [Creating a Distributed Queue and Adding Existing Physical Queues as Members Manually](#)
- Note:** The default Load Balancing Enabled and Server Affinity Enabled attributes for tuning a distributed destination configuration can be modified on the JMS connection factory through the Administration Console. For more information, see [“Configuring Message Load Balancing Across a Distributed Destination” on page 235-6](#) and [“Configuring Server Affinity For a Distributed Destination” on page 235-7](#).

When a distributed topic or queue destination is created, a corresponding JMS template is automatically created with default attribute values for the distributed destination members. The new template will appear under the JMS Templates node with the same name as the distributed destination. The thresholds, quotas, and other attributes for the distributed destination members can be reset using this template.

Creating a Distributed Topic and Creating Members Automatically

Follow these steps to configure a distributed topic and automatically create its topic members on JMS servers that are part of a WebLogic Server cluster (for high availability) or on an single WebLogic Server instance that is not part of a cluster.

1. Expand the JMS → Distributed Destinations node.
2. Click the Configure a new Distributed Topic link in the right pane. A dialog shows the tabs associated with configuring a new distributed topic.
3. On the Configuration General tab, define the general configuration attributes for the distributed topic.
 - Enter a name for the distributed topic.
 - Enter a JNDI name for accessing the distributed topic within the JNDI namespace. Applications use the JNDI Name to look up the distributed topic.

Note: A distributed topic that does not have a JNDI Name can be referenced by passing the Name of the distributed topic to
`javax.jms.TopicSession.createTopic()`.
 - Define how producers will distribute their messages across the members of a distributed topic. The valid values are Round-Robin and Random, as explained in “Configuring Message Load Balancing Across a Distributed Destination” on page 235-6.

For more information about the distributed topic general attributes, see [“JMS Distributed Topic --> Configuration --> General” on page 174-1](#).

4. Click Create to create a distributed topic instance with the name you specified in the Name field. The new instance is added under the Distributed Destinations node in the left pane.
5. On the Thresholds & Quotas tab, define the following message/byte thresholds and maximum quota attributes for all distributed topic members:
 - Specify the maximum bytes or message quota that can be stored in a distributed topic.

- Specify the upper threshold value that trigger events based on the number of bytes or messages stored in a distributed topic. Events include message paging, message flow control, and system log messages.
- Specify the lower threshold value that trigger events based on the number of bytes or messages stored in a distributed topic. Events include message paging, message flow control, and system log messages.
- Indicate whether bytes or messages paging is enabled on a distributed topic for temporarily swapping messages out from memory to a paging store when a distributed topic's message load reaches a specified bytes/messages threshold.
- Specify the maximum size of a message that will be accepted from message producers on a distributed topic. The size includes the message body, any user-defined properties, and the user-defined JMS header fields: JMSCorrelationID and JMSType.

For more information about these attributes, see [“JMS Distributed Topic --> Configuration --> Thresholds and Quotas”](#) on page 176-1.

6. Click Apply to save any changes you made on this tab.
7. On the Auto Deploy tab, indicate the WebLogic Server instances where you want the distributed topic members to be automatically created.
8. Click the Create members on the selected Servers (and JMS Servers) text link. An auto deploy dialog prompts you to select one of the following options:
 - Select a cluster at which to target the distributed topic, and then click Next.

or

 - Accept the None option to bypass this dialog so you can select an individual server or servers that are part of the cluster. (In this case, skip to Step 10).
9. If you selected a cluster, do the following to select WebLogic Server instances within the cluster:
 - a. All WebLogic Server instances that are members of the cluster and that are not already hosting a distributed topic are listed and are selected by default. To exclude a server instance from hosting the distributed topic, clear the corresponding check box.
 - b. Click Next to proceed to the next dialog.

- c. Skip to Step 11 to select the JMS servers that are available on the selected WebLogic Servers for creating distributed topic members.
10. If you selected None on the Cluster dialog in Step 8, select a single WebLogic Server instance in the domain:
 - a. From the list box, select an individual server where you want to create a distributed topic member.
 - b. Click Next to proceed to the next dialog.
11. All JMS servers that are deployed on the selected WebLogic Server instances, and that are not already hosting a distributed topic, are listed and are selected by default. To exclude a JMS server from hosting the distributed topic member, clear the corresponding check box.

If there are no existing distributed topic members on the selected JMS servers, one new JMS Topic will be created on each JMS server and added as a member of the distributed topic.
12. Click Next to proceed to the final Auto Deploy dialog.
13. Click Apply to save your Auto Deploy selections.
14. Click the Configuration →Members tab to view the topic members that were automatically created for the new distributed topic.
15. Expand the JMS →Templates node to view the JMS template that was automatically created with the same name as the distributed topic.

Creating a Distributed Topic and Adding Existing Physical Topics as Members Manually

For existing implementations of WebLogic JMS that require previously configured destinations to be members of a distributed destination set, follow these steps to configure a distributed topic and manually add your existing physical topics as members:

1. Expand the JMS →Distributed Destinations node.

2. Click the Configure a new Distributed Topic link in the right pane. A dialog shows the tabs associated with configuring a new distributed topic.
3. On the Configuration General tab, define the general configuration attributes for the distributed topic.
 - Enter a name for the distributed topic.
 - Enter a JNDI name for accessing the distributed topic within the JNDI namespace. Applications use the JNDI Name to look up the distributed topic.

Note: A distributed topic that does not have a JNDI Name can be referenced by passing the Name of the distributed topic to
`javax.jms.TopicSession.createTopic()`.
 - Define how producers will distribute their messages across the members of a distributed topic. The valid values are Round-Robin and Random as defined in “Configuring Message Load Balancing Across a Distributed Destination” on page 235-6.

For more information about the distributed topic general attributes, see “[JMS Distributed Topic --> Configuration --> General](#)” on page 174-1.

4. Click Create to create a distributed topic instance with the name you specified in the Name field. The new instance is added under the Distributed Destinations node in the left pane.
5. On the Thresholds & Quotas tab, define the following upper and lower message/byte threshold and maximum quota attributes for all distributed topic members:
 - Specify the maximum bytes or message quota that can be stored in a distributed topic.
 - Specify the upper threshold value that trigger events based on the number of bytes or messages stored in a distributed topic. Events include message paging, message flow control, and system log messages.
 - Specify the lower threshold value that trigger events based on the number of bytes or messages stored in a distributed topic. Events include message paging, message flow control, and system log messages.
 - Indicate whether bytes or messages paging is enabled on a distributed topic for temporarily swapping messages out from memory to a paging store when a distributed topic’s message load reaches a specified bytes/messages threshold.

- Specify the maximum size of a message that will be accepted from message producers on a distributed topic. The size includes the message body, any user-defined properties, and the user-defined JMS header fields: JMSCorrelationID and JMSType.

If a distributed topic member's underlying physical topic already uses a JMS Template with configured thresholds and quotas, these attributes will not apply to that topic member. For more information about these attributes, see “JMS Distributed Topic --> Configuration --> Thresholds and Quotas” on page 176-1.

6. Click Apply to Apply to save any changes you made on this tab.

Note: If you want to automatically create topic members on JMS servers that are part of a WebLogic Server cluster (for high availability) or on a single WebLogic Server instance that is not part of a cluster, see “Creating a Distributed Topic and Creating Members Automatically” on page 232-33.

7. On the Configuration →Members tab, create distributed topic members for your existing physical topics.
8. Click the Configure a new Distributed Topic Member link in the right pane. A Configuration dialog shows the tabs associated with configuring a new distributed topic member.
9. On the Configuration tab, define the general configuration attributes for the distributed topic.
 - Uniquely identify the distributed topic member within a WebLogic Server domain.
 - Select the underlying physical topic that is associated with the distributed topic member.
 - Define the weight (that is, a measure of ability to handle message load) of the topic member with respect to other topic members in the distributed destination. For more information about load balancing distributed destinations, see “[Developing a WebLogic JMS Application](#)” in *Programming WebLogic JMS*.

For more information about distributed topic member attributes, see “[JMS Distributed Topic Member --> Configuration](#)” on page 178-1.

10. Click Create to create the new distributed topic member. The new member is added to the Distributed Topic table.

11. If necessary, repeat steps 8–10 to continue adding topic members to the distributed topic.
12. Expand the JMS →Templates node to view the JMS template that was automatically created with the same name as the distributed topic.

Creating a Distributed Queue and Creating Members Automatically

Follow these steps to configure a distributed queue and automatically create its queue members on JMS servers that are part of a WebLogic Server cluster (for high availability) or on an single WebLogic Server instance that is not part of a cluster.

1. Expand the JMS →Distributed Destinations node.
2. Click the Configure a new Distributed Queue link in the right pane. A Configuration dialog shows the tabs associated with configuring a new distributed queue.
3. On the Configuration General tab, define the general configuration attributes for the distributed topic.
 - Enter a name for the distributed queue.
 - Enter a JNDI name for accessing the distributed queue within the JNDI namespace. Applications use the JNDI Name to look up the distributed queue.

Note: A distributed queue that does not have a JNDI Name can be referenced by passing the Name of the distributed topic to
`javax.jms.QueueSession.createQueue()`.
 - Define how producers will distribute their messages across the members of a distributed queue. The valid values are Round-Robin and Random, as explained in “Configuring Message Load Balancing Across a Distributed Destination” on page 235-6.
 - Define the amount of time, in seconds, that a distributed queue member with messages, but which has no consumers, will wait before forwarding its messages to other queue members that do have consumers.

For more information about the distributed queue general attributes, see “[JMS Distributed Queue --> Configuration --> General](#)” on page 167-1.

4. Click Create to create a distributed queue instance with the name you specified in the Name field. The new instance is added under the Distributed Destinations node in the left pane.
5. On the Thresholds & Quotas tab, define the following upper and lower message/byte threshold and maximum quotas for all distributed queue members:
 - Specify the maximum bytes or message quota that can be stored in a distributed queue.
 - Specify the upper threshold value that trigger events based on the number of bytes or messages stored in a distributed queue. Events include message paging, message flow control, and system log messages.
 - Specify the lower threshold value that trigger events based on the number of bytes or messages stored in a distributed queue. Events include message paging, message flow control, and system log messages.
 - Indicate whether bytes or messages paging is enabled on a distributed topic for temporarily swapping messages out from memory to a paging store when a distributed queue's message load reaches a specified bytes/messages threshold.
 - Specify the maximum size of a message that will be accepted from message producers on a distributed queue. The size includes the message body, any user-defined properties, and the user-defined JMS header fields: JMSCorrelationID and JMSType.

If a distributed queue member's underlying physical queue already uses a JMS Template with configured thresholds and quotas, these attributes will not apply to that queue member. For more information about these attributes, see "JMS Distributed Topic --> Configuration --> Thresholds and Quotas" on page 176-1.

6. Click Apply to save any changes you made on this tab.
7. On the Auto Deploy tab, indicate the WebLogic Server instances where you want the distributed queue members to be automatically created.
8. Click the Create members on the selected Servers (and JMS Servers) text link. A dialog prompts you to select one of the following options:
 - Select a cluster at which to target the distributed queue, and then click Next.

or

- Accept the None option to bypass this dialog so you can select an individual server that is not in a cluster. (In this case, skip to Step 10).
9. If you selected a cluster, do the following to select WebLogic Server instances within the cluster:
 - a. All servers that are members of the cluster, and which are not already hosting a distributed queue, are listed and are selected by default. To exclude a server from hosting the distributed queue, clear the corresponding check box.
 - b. Click Next to proceed to the next dialog.
 - c. Skip to Step 11 to select the JMS servers that are available on the selected WebLogic Servers for creating distributed queue members.
 10. If you selected None on the Cluster dialog in Step 8, select a single WebLogic Server instance in the domain:
 - a. From the list box, select an individual server where you want to create the distributed queue member.
 - b. Click Next to proceed to the next dialog.
 11. All JMS servers that are deployed on the selected WebLogic Servers, and which are not already hosting a distributed queue, are listed and are selected by default. To exclude a JMS server from hosting the distributed queue member, clear the corresponding check box.

If there are no existing distributed queue members on the selected JMS servers, one new JMS Queue will be created on each JMS server and added as a member of the distributed queue.
 12. Click Next to proceed to the final Auto Deploy dialog.
 13. Click Apply to save your Auto Deploy selections.
 14. Click the Configuration →Members tab to view the queue members that were automatically created for the new distributed queue.
 15. Expand the JMS →Templates node to view the JMS template that was automatically created with the same name as the distributed queue.

Creating a Distributed Queue and Adding Existing Physical Queues as Members Manually

For existing implementations of WebLogic JMS that require previously configured destinations to be members of a distributed destination set, follow these steps to configure a distributed queue and manually add your existing physical queues as members.

1. Expand the JMS → Distributed Destinations node.
2. Click the Configure a new Distributed Queue link in the right pane. A Configuration dialog shows the tabs associated with configuring a new distributed queue.
3. On the Configuration General tab, define the general configuration attributes for the distributed topic.

- Enter a name for the distributed queue.
- Enter a JNDI name for accessing the distributed queue within the JNDI namespace. Applications use the JNDI Name to look up the distributed queue.

Note: A distributed queue that does not have a JNDI Name can be referenced by passing the Name of the distributed topic to:

```
javax.jms.QueueSession.createQueue()
```

- Define how producers will distribute their messages across the members of a distributed queue. The valid values are Round-Robin and Random, as explained in “Configuring Message Load Balancing Across a Distributed Destination” on page 235-6.
- Define the amount of time, in seconds, that a distributed queue member with messages, but which has no consumers, will wait before forwarding its messages to other queue members that do have consumers.

For more information about the distributed queue general attributes, see [“JMS Distributed Queue --> Configuration --> General” on page 167-1](#).

4. Click Create to create a distributed queue instance with the name you specified in the Name field. The new instance is added under the Distributed Destinations node in the left pane.
5. On the Thresholds & Quotas tab, define the following upper and lower message/byte threshold and maximum quotas for all distributed queue members:

- Specify the maximum bytes or message quota that can be stored in a distributed queue.
- Specify the upper threshold value that trigger events based on the number of bytes or messages stored in a distributed queue. Events include message paging, message flow control, and system log messages.
- Specify the lower threshold value that trigger events based on the number of bytes or messages stored in a distributed queue. Events include message paging, message flow control, and system log messages.
- Indicate whether bytes or messages paging is enabled on a distributed topic for temporarily swapping messages out from memory to a paging store when a queue's message load reaches a specified bytes/messages threshold.
- Specify the maximum size of a message that will be accepted from message producers on a distributed queue. The size includes the message body, any user-defined properties, and the user-defined JMS header fields: JMSCorrelationID and JMSType.

If a distributed queue member's underlying physical queue already has a JMS Template with configured thresholds and quotas, these attributes will not apply to that queue member. For more information about these attributes, see “JMS Distributed Queue --> Configuration --> Thresholds and Quotas” on page 169-1.

6. Click Apply to save any changes you made on this tab.

Note: If you want to automatically create queue members on JMS servers that are part of a WebLogic Server cluster (for high availability) or on a single WebLogic Server instance that is not part of a cluster, see “Creating a Distributed Queue and Creating Members Automatically” on page 232-38.

7. Click the Configuration →Members tab to define the queue members for the distributed queue.
8. Click the Configure a new Distributed Queue Member text link in the right pane. A Configuration dialog shows the tabs associated with configuring a new distributed queue member.
9. On the Configuration tab, define the general configuration attributes for the distributed queue.
 - Uniquely identify the distributed queue member within a WebLogic Server domain.

- Select the underlying physical queue that is associated with the distributed queue member.
- Define the weight (that is, a measure of ability to handle message load) of the queue member with respect to other topic members in the distributed destination. For more information about load balancing for distributed destinations, see “[Developing a WebLogic JMS Application](#)” in *Programming WebLogic JMS*.

For more information about distributed queue member attributes, see “[JMS Distributed Queue Member --> Configuration](#)” on page 171-1.

10. Click Create to create the new distributed queue member. The new member is added to the Distributed Queue table.
11. Repeat steps 8–10 to continue adding members to the distributed queue.
12. Expand the JMS →Templates node to view the JMS template that was automatically created with the same name as the distributed queue.

Creating a JMS Distributed Queue Member

Follow these steps to add an existing physical queue as a member of distributed queue.

1. Expand the JMS →Distributed Destinations node. The Distributed Destinations table displays in the right pane showing all the distributed queues and topics.
2. Click the distributed queue that you want to add a member to. The Distributed Queue table shows all the distributed queue members that belong to the distributed queue.
3. Click the Configure a new Distributed Queue Member text link. A dialog shows the Configuration tab for configuring a new distributed queue member.
4. Define the configuration attributes for the distributed queue.
 - Uniquely identify the distributed queue member within a WebLogic Server domain.
 - Select the underlying physical queue that is associated with the distributed queue member.

- Define the weight (that is, a measure of ability to handle message load) of the queue member with respect to other topic members in the distributed destination. For more information about load balancing for distributed destinations, see [“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*.

For more information about distributed queue member attributes, see [“JMS Distributed Queue Member --> Configuration”](#) on page 171-1.

5. Click Create to create a distributed queue member with the name you specified in the Name field. The new member is added to the Distributed Queue Member table in the right pane.
6. Click Apply to save any changes you made.

Deleting a JMS Distributed Queue Member

Follow these steps to delete a member of distributed queue, and if necessary, to delete the member’s underlying physical queue.

Note: If you need to delete the entire distributed queue, then follow the directions for [“Deleting a Distributed Destination”](#) on page 232-46.

1. Expand the JMS →Distributed Destinations node. The Distributed Destinations table displays in the right pane showing all the distributed queues and topics.
2. Click the distributed queue that you want to delete a member from. The Distributed Queue table shows all the distributed queue members that belong to the distributed queue.
3. Click the Delete icon in the row of the distributed queue member that you want to delete. A dialog prompts you to confirm your deletion request.
4. If you want to also remove the underlying physical queue, select the Also Delete check box.
5. Click Remove to delete the distributed queue member (and the underlying physical queue if selected).
6. The Distributed Queue table redisplay in the right pane. The distributed queue member is deleted from the Distributed Queue table.

Creating a JMS Distributed Topic Member

Follow these steps to add an existing physical queue as a member of distributed queue.

1. Expand the JMS → Distributed Destinations node. The Distributed Destinations table displays in the right pane showing all the distributed queues and topics.
2. Click the distributed topic that you want to add a member to. The Distributed Topic table shows all the distributed topic members that belong to the distributed topic.
3. Click the Configure a new Distributed Topic Member text link. A dialog shows the Configuration tab for configuring a new distributed topic member.
4. Define the general configuration attributes for the distributed topic.
 - Uniquely identify the distributed topic member within a WebLogic Server domain.
 - Select the underlying physical topic that is associated with the distributed topic member.
 - Define the weight (that is, a measure of ability to handle message load) of the topic member with respect to other topic members in the distributed destination. For more information about load balancing distributed destinations, see [“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*.

For more information about distributed topic member attributes, see [“JMS Distributed Topic Member --> Configuration”](#) on page 178-1.

5. Click Create to create a distributed topic member with the name you specified in the Name field. The new member is added to the Distributed Topic table in the right pane.
6. Click Apply to save any changes you made.

Deleting a JMS Distributed Topic Member

Follow these steps to delete a member of distributed topic, and if necessary, to delete the member’s underlying physical topic.

Note: If you need to delete the entire distributed topic, then follow the directions for “Deleting a Distributed Destination” on page 232-46.

1. Expand the JMS node.
2. Expand the Distributed Destinations node. The Distributed Destinations table displays in the right pane showing all the distributed queues and topics.
3. Click the distributed topic that you want to delete a member from. The Distributed Topic table shows all the distributed topic members that belong to the distributed topic.
4. Click the Delete icon in the row of the distributed topic member that you want to delete. A dialog prompts you to confirm your deletion request.
5. If you want to also remove the underlying physical queue, select the Also Delete check box.
6. Click Remove to delete the distributed topic member (and the underlying physical topic if selected).
7. The Distributed Topic table redisplay in the right pane. The distributed topic member is deleted from the Distributed Topic table.

Deleting a Distributed Destination

If you need to delete an entire distributed destination, you must remove it in the following sequence:

1. Delete *all* the members of the distributed queue or distributed topic, as explained in the following sections:
 - “Deleting a JMS Distributed Queue Member” on page 232-44
 - “Deleting a JMS Distributed Topic Member” on page 232-45
2. Delete the distributed destination itself by expanding the JMS → Distributed Destinations node, and then clicking the trash icon next to the distributed destination that you want to delete.

Note: You can only delete a distributed destination when *all* its members have been properly deleted.

3. You can delete the JMS template that is associated with the distributed destination. However, make sure that this template is not being used by other JMS servers or destinations. To delete a JMS Template, expand the JMS → Templates node, and then click the trash icon in the row of the JMS template that you want to delete.

Monitoring a Distributed Destinations

When monitoring distributed destinations, you may see proxy topic members or system subscriptions, which are automatically created for the topic or queue members. For more information see, “Monitoring Distributed Destination System Subscriptions and Proxy Topic Members” on page 234-6.

Simple Access to Remote or Foreign JMS Providers

WebLogic JMS enables you to reference foreign (that is, third-party) JMS providers within a local WebLogic Server JNDI tree. Using the Foreign JMS Server node, you can quickly map a foreign JMS provider so that its associated connection factories and destinations appear in the WebLogic JNDI tree as local JMS objects. A Foreign JMS Server configuration can also be used to reference remote instances of WebLogic Server in another cluster or domain in the local WebLogic JNDI tree.

Note: In order to use the Foreign JMS Providers feature to reference remote WebLogic Server clusters or domains, you must have a clustered JMS license, which allows a connection factory and a destination to be on different server instances. If you do not have a valid clustered JMS license, contact your BEA sales representative.

The following sections provide more information on how the Foreign JMS Server node works, configuration instructions, and a sample configuration for accessing a remote MQSeries JNDI provider.

- “How WebLogic JMS Accesses Foreign JMS Providers” on page 232-48
- “Creating a Foreign JMS Server” on page 232-49
- “Creating a Foreign JMS Connection Factory” on page 232-50
- “Creating a Foreign JMS Destination” on page 232-51
- “Sample Configuration for MQSeries JNDI” on page 232-53

How WebLogic JMS Accesses Foreign JMS Providers

When a foreign JMS server is deployed, it creates local connection factory and destination objects in WebLogic Server JNDI. Then when a foreign JMS connection factory or destination object is looked up on the local server, that object performs the actual lookup on the remote JNDI directory, and the foreign object is returned from that directory.

This method makes it easier to configure multiple WebLogic Messaging Bridge destinations, since the foreign JMS server moves the JNDI Initial Context Factory and Connection URL configuration details outside of your Messaging Bridge destination configurations. You need only provide the foreign Connection Factory and Destination JNDI name for each object.

For more information on configuring a Messaging Bridge, see “Messaging Bridge Configuration Tasks” on page 289-3.

The ease-of-configuration concept also applies to configuring WebLogic Servlets, EJBs, and Message-Driven Beans (MDBs) with WebLogic JMS. For example, the `weblogic-ejb-jar.xml` file in the MDB can have a local JNDI name, and you can use the foreign JMS server to control where the MDB receives messages from. For example, you can deploy the MDB in one environment to talk to one JMS destination and server, and you can deploy the same `weblogic-ejb-jar.xml` file to a different server and have it talk to a different JMS destination without having to unpack and edit the `weblogic-ejb-jar.xml` file.

Creating a Foreign JMS Server

A *Foreign JMS Server* represents a JNDI provider that is outside the WebLogic JMS server. It contains information that allows a local WebLogic Server instance to reach a remote JNDI provider, thereby allowing for a number of foreign JMS connection factory and destination objects to be defined on one JNDI directory.

After defining a foreign JMS server, you can configure connection factory and destination objects. You can configure one or more connection factories and destinations (queues or topics) for each foreign JMS server.

To configure a foreign JMS server, follow these steps:

1. Expand the JMS node in the navigation tree, and then click the Foreign JMS Servers node.
2. Click the Configure a new Foreign JMSServer text link in the right pane. A dialog shows the tabs associated with configuring a new Foreign JMS server.
3. On the Configuration General tab, enter values in the Name, JNDI Initial Context Factory, JNDI Connection URL, and JNDI Properties attribute fields.

For a detailed description of the general Foreign JMS Server attributes, see “Foreign JMS Server --> Configuration --> General” on page 225-1.

Note: The JNDI Properties values may be a `name=value` list of properties, separated by commas.

4. Click Create to create a foreign JMS server instance with the name you specified in the Name field. The new instance is added under the Foreign JMS Servers node in the navigation tree, and a Foreign JMS Connection Factories node and Foreign JMS Destinations node are automatically added under the new foreign JMS server instance.
 5. On the Targets tab, select a standalone WebLogic Server instance or a cluster on which to deploy the foreign JMS server.
 - Servers tab—On the Available list, select the WebLogic Server instance where you want to deploy the foreign JMS server object.
 - Clusters tab—On the Targets list, select a WebLogic Server cluster in order to deploy the foreign JMS server object on every node in the cluster.
- Note:** The Clusters tab is available only when the JMS server is part of a clustered WebLogic Server environment.
6. Click Apply to target the foreign JMS server.

Continue by configuring a connection factory and destination objects. You can configure one or more connection factories and destinations (queues or topics) for each foreign JMS server.

Creating a Foreign JMS Connection Factory

A *Foreign JMS Connection Factory* contains the JNDI name of the connection factory in the remote JNDI provider, the JNDI name that the connection factory is mapped to in the local WebLogic Server JNDI tree, and an optional user name and password.

The foreign JMS connection factory creates non-replicated JNDI objects on each WebLogic Server instance that the parent foreign JMS server is targeted to. (To create the JNDI object on every node in a cluster, target the foreign JMS server to the cluster.)

To configure a foreign JMS connection factory, follow these steps:

1. Expand the Foreign JMS Servers node, and then expand the Foreign JMS Connection Factories node.
2. Click the Configure a new Foreign JMSConnection Factory text link. A dialog displays in the right pane showing the tabs associated with configuring a new foreign JMS connection factory.
3. On the Configuration General tab, enter values in the Name, Local JNDI Name, Remote JNDI Name, User Name, and Password attribute fields.

For a detailed description of the general Foreign JMS Connection Factory attributes, see “Foreign JMS Connection Factory --> Configuration --> General” on page 223-1.

Note: The user name and password are only used when the Foreign JMS Connection Factory is used inside a *resource-reference* in an EJB or a servlet, and when the *Container* mode of authentication is used.

4. Click Create to create a foreign JMS connection factory instance with the name you specified in the Name field. The new instance is added under the Foreign JMS Connection Factory node in the navigation tree.

Continue by configuring the destination objects. You can configure one or more destinations (queues or topics) for each foreign JMS server.

Creating a Foreign JMS Destination

A *Foreign JMS Destination* represents either a queue or a topic. It contains the destination JNDI name that is looked up on the foreign JNDI provider and the JNDI name that the destination is mapped to on the local WebLogic Server. When the foreign destination is looked up on the local server, a lookup is performed on the remote JNDI directory, and the destination object is returned from that directory.

To configure a foreign JMS destination, follow these steps:

1. Expand the Foreign JMS Servers node, and then expand the Foreign JMS Destinations node.
2. Click the Configure a new Foreign JMSDestination text link. A dialog displays in the right pane showing the tabs associated with configuring a foreign destination.

3. On the Configuration General tab, enter values in the Name, Local JNDI Name and Remote JNDI Name attribute fields.

For a detailed description of the general Foreign JMS Destination attributes, see “Foreign JMS Destination --> Configuration --> General” on page 222-1.

4. Click Create to create a foreign JMS destination instance with the name you specified in the Name field. The new instance is added under the Foreign JMS Destinations node in the navigation tree.

Sample Configuration for MQSeries JNDI

The following table provides a possible a sample configuration when accessing a remote MQSeries JNDI provider.

Table 232-3 Sample MQSeries Configuration

Foreign JMS Object	Attribute Names	Sample Configuration Data
Foreign JMS Server	Name	MQJNDI
	JNDI Initial Context Factory	com.sun.jndi.fscontext.RefFSContextFactory
	JNDI Connection URL	file:/MQJNDI/
	JNDI Properties	(If necessary, enter a comma-separated name=value list of properties.)
Foreign JMS Connection Factory	Name	MQ_QCF
	Local JNDI Name	mqseries.QCF
	Remote JNDI Name	QCF
	Username	weblogic_jms
	Password	weblogic_jms
Foreign JMS Destination 1	Name	MQ_QUEUE1
	Local JNDI Name	mqseries.QUEUE1
	Remote JNDI Name	QUEUE_1
Foreign JMS Destination 2	Name	MQ_QUEUE2
	Local JNDI Name	mqseries.QUEUE2
	Remote JNDI Name	QUEUE_2

For detailed information about foreign server, connection factory, and attributes, and the valid and default values for them, refer to the following sections:

- “Foreign JMS Server --> Configuration --> General” on page 225-1
- “Foreign JMS Connection Factory --> Configuration --> General” on page 223-1
- “Foreign JMS Destination --> Configuration --> General” on page 222-1

Attributes and Console Screen Reference for JMS

For information about an Administration Console screen and the attributes you can configure, select one of the links in the following categories:

- [Connection Factories](#)
- [Stores](#)
- [JMS Servers](#)
- [Destinations](#)
- [Session Pools](#)
- [Connection Consumers](#)
- [Templates](#)
- [Destination Keys](#)
- [Distributed Destinations](#)
- [Foreign JMS Servers](#)
- [Monitoring JMS Connections and Sessions](#)

Connection Factories

[“JMS Connection Factory” on page 159-1](#)

[“JMS Connection Factory --> Configuration --> General” on page 155-1](#)

[“JMS Connection Factory --> Configuration --> Transactions” on page 156-1](#)

[“JMS Connection Factory --> Configuration --> Flow Control” on page 154-1](#)

[“JMS Connection Factory --> Target and Deploy” on page 158-1](#)

“JMS Connection Factory --> Notes” on page 157-1

Stores

“JMS Store” on page 204-1

“JMS File Store --> Configuration” on page 181-1

“JMS File Store --> Notes” on page 182-1

“JMS JDBC Store --> Configuration” on page 183-1

“JMS JDBC Store --> Notes” on page 184-1

JMS Servers

“JMS Server” on page 198-1

“JMS Server --> Configuration --> General” on page 192-1

“JMS Server --> Configuration --> Thresholds & Quotas” on page 193-1

“JMS Server --> Target and Deploy” on page 196-1

“JMS Server --> Notes” on page 194-1

“JMS Server --> Monitoring” on page 195-1

“Active JMS Servers” on page 197-1

Destinations

“JMS Destination” on page 165-1

“Active JMS Destinations” on page 164-1

Topics

- “JMS Topic --> Configuration --> General” on page 211-1
- “JMS Topic --> Configuration --> Thresholds & Quotas” on page 215-1
- “JMS Topic --> Configuration --> Overrides” on page 213-1
- “JMS Topic --> Configuration --> Redelivery” on page 214-1
- “JMS Topic --> Configuration --> Expiration Policy” on page 231-1
- “JMS Topic --> Configuration --> Multicast” on page 212-1
- “JMS Topic --> Notes” on page 217-1
- “JMS Topic --> Monitoring” on page 216-1
- “Durable Subscribers” on page 180-1

Queues

- “JMS Queue --> Configuration --> General” on page 186-1
- “JMS Queue --> Configuration --> Thresholds & Quotas” on page 189-1
- “JMS Queue --> Configuration --> Overrides” on page 187-1
- “JMS Queue --> Configuration --> Redelivery” on page 188-1
- “JMS Queue --> Configuration --> Expiration Policy” on page 229-1
- “JMS Queue --> Notes” on page 191-1
- “JMS Queue --> Monitoring” on page 190-1

Session Pools

- “JMS Session Pool” on page 202-1
- “JMS Session Pool --> Configuration” on page 199-1
- “JMS Session Pool --> Notes” on page 200-1

“Active JMS Session Pools” on page 201-1

Connection Consumers

“JMS Connection Consumer” on page 153-1

“JMS Connection Consumer --> Configuration” on page 151-1

“JMS Connection Consumer --> Notes” on page 152-1

Templates

“JMS Template” on page 210-1

“JMS Template --> Configuration --> General” on page 205-1

“JMS Template --> Configuration --> Thresholds & Quotas” on page 208-1

“JMS Template --> Configuration --> Override” on page 206-1

“JMS Template --> Configuration --> Redelivery” on page 207-1

“JMS Template --> Configuration --> Expiration Policy” on page 230-1

“JMS Template --> Notes” on page 209-1

Destination Keys

“JMS Destination Key” on page 163-1

“JMS Destination Key --> Configuration” on page 161-1

“JMS Destination Key --> Notes” on page 162-1

Distributed Destinations

“JMS Distributed Destinations” on page 218-1

Distributed Queues

“JMS Distributed Queue” on page 172-1

“JMS Distributed Queue --> Auto Deploy” on page 166-1

“JMS Distributed Queue --> Configuration --> Thresholds and Quotas” on page 169-1

“JMS Distributed Queue --> Configuration --> General” on page 167-1

“Distributed Queue --> Configuration --> Members” on page 168-5

“JMS Distributed Queue --> Notes” on page 170-1

“JMS Distributed Queue Member --> Configuration” on page 171-1

Distributed Topics

“JMS Distributed Topic” on page 179-1

“JMS Distributed Topic --> Auto Deploy” on page 173-1

“JMS Distributed Topic --> Configuration --> General” on page 174-1

“JMS Distributed Topic --> Configuration --> Thresholds and Quotas” on page 176-1

“Distributed Topic --> Configuration --> Members” on page 175-5

“JMS Distributed Topic --> Notes” on page 177-1

“JMS Distributed Topic Member --> Configuration” on page 178-1

Foreign JMS Servers

“Foreign JMS Server” on page 228-1

“Foreign JMS Server --> Configuration --> General” on page 225-1

“Foreign JMS Server --> Notes” on page 226-1

“Foreign JMS Server --> Target and Deploy” on page 227-1

Foreign JMS Connection Factories

“Foreign JMS Connection Factory” on page 221-1

“Foreign JMS Connection Factory --> Configuration --> General” on page 223-1

“Foreign JMS Connection Factory --> Configuration --> Notes” on page 220-1

Foreign JMS Destinations

“Foreign JMS Destination” on page 224-1

“Foreign JMS Destination --> Configuration --> General” on page 222-1

“Foreign JMS Destination --> Configuration --> Notes” on page 219-1

Monitoring JMS Connections and Sessions

“Active JMS Connections” on page 160-1

“JMS Pooled Connections” on page 185-1

“Active JMS Sessions” on page 203-1

“Active JMS Consumers” on page 203-1

“Active JMS Producers” on page 203-1

1 JMS: Monitoring

[“Attributes and Console Screen Reference for JMS” on page 233-1]

The following sections explain how to monitor the run-time statistics for your JMS objects from the Administration Console:

- “Monitoring Pooled JMS Connections” on page 234-2
- “Monitoring Active JMS Connections” on page 234-3
- “Monitoring Active JMS Sessions, Consumers, and Producers” on page 234-3
- “Monitoring Active JMS Servers” on page 234-4
- “Monitoring Active JMS Destinations” on page 234-4
- “Monitoring Durable Subscribers for Topics” on page 234-5
- “Monitoring Active JMS Session Pools” on page 234-6
- “Monitoring Distributed Destination System Subscriptions and Proxy Topic Members” on page 234-6

You may also want to refer these WebLogic JMS and Messaging Bridge sections:

- “JMS: Configuring” on page 232-1
- “JMS: Tuning” on page 235-1
- “Messaging Bridge” on page 289-1

Overview

You can monitor statistics for the following JMS objects: JMS servers, connections, pooled connections, destinations, message producers, message consumers, JMS server session pools, and durable subscribers for JMS topics.

JMS statistics continue to increment as long as the server is running. Statistics are reset only when the server is rebooted.

Note: For instructions on monitoring JMS connections to WebLogic Server, refer to “Server --> Monitoring --> JMS” on page 470-1.

Monitoring Pooled JMS Connections

You can monitor statistics on all the active pooled JMS connections on your server. A pooled JMS connection is a session pool used by EJBs and servlets that use a resource-reference element in their EJB or servlet deployment descriptor to define their JMS connection factories.

1. Expand the Servers node.
2. Select the WebLogic Server instance where you want to monitor JMS connections.
3. Select the Monitoring tab.
4. Select the JMS tab. Totals for JMS server and connection statistics are displayed in the JMS dialog.
5. Click the Monitor all Pooled JMS Connections text link. A table displays showing all the pooled JMS connection for the server, as described in “[JMS Pooled Connections](#)” on page 185-1.

Monitoring Active JMS Connections

You can monitor statistics on all the active JMS connections on your server. A JMS connection is an open communication channel to the messaging system.

1. Expand the Servers node.
2. Select the WebLogic Server instance where you want to monitor JMS connections.
3. Select the Monitoring tab in the right pane.
4. Select the JMS tab. Totals for JMS server and connection statistics are displayed in the JMS dialog.
5. Click the Monitor all Active JMS Connections text link. A table displays showing all active JMS connection for the server, as described in [“Active JMS Connections” on page 160-1](#).

Monitoring Active JMS Sessions, Consumers, and Producers

You can monitor statistics on all the active JMS sessions, consumers, and producers on your server. A session defines a serial order for both the messages produced and the messages consumed, and can create multiple message producers and message consumers. The same thread can be used for producing and consuming messages.

1. Expand the Servers node.
2. Select the WebLogic Server instance where you want to monitor JMS connections.
3. Select the Monitoring tab in the right pane.
4. Select the JMS tab. Totals for JMS server and connection statistics are displayed in the JMS dialog.
5. Click the Monitor all Active JMS Connections text link. A table displays showing all active JMS connection for the server.

6. In the Sessions column, click the hyperlinked number for the session that you want to monitor. The “Active JMS Sessions” table displays with statistics for the consumers and producers that are part of the session, as described in [“Active JMS Sessions” on page 203-1](#).
7. To drill down for more detailed information about the session’s consumers and producers, click the hyperlinked number for the specific consumer or producer that you want to monitor, as described in [“Active JMS Consumers” on page 203-1](#) and [“Active JMS Producers” on page 203-1](#).

Monitoring Active JMS Servers

You can monitor statistics on all the active JMS servers defined in your domain. A JMS server manages connections and message requests on behalf of JMS clients.

1. Expand the JMS →Servers node.
2. The JMS Servers information displays in the right pane showing all the JMS servers defined in your domain.
3. Click the JMS server that you want to monitor from the JMS server list, or from the JMS Servers table displayed in the right pane.
4. Click the Monitoring tab to display the monitoring data.
5. Click the Monitor all Active JMS Servers text link in the right pane.
6. A table displays showing all instances of the JMS server deployed across the WebLogic Server domain, as described in [“Active JMS Servers” on page 197-1](#).

Monitoring Active JMS Destinations

You can monitor statistics on all the active destinations currently running on a JMS server. A JMS destinations identify queue (Point-To-Point) or topic (Pub/Sub) destination types for a JMS server

1. Expand the JMS →Servers node.

2. The JMS Servers table displays in the right pane showing all the JMS servers defined in your domain.
3. Click the JMS server that you want to monitor from the JMS server list, or from the JMS Servers table displayed in the right pane.
4. Click the Monitoring tab to display the monitoring data.
5. Click the Monitor all Active JMS Destinations text link. A table displays in the right pane showing all active JMS destinations for the current domain, as described in [“Active JMS Destinations” on page 164-1](#).

Note: When monitoring distributed destinations, you may see proxy topic members or system subscriptions for the topic or queue members. For more information see, “Monitoring Distributed Destination System Subscriptions and Proxy Topic Members” on page 234-6.

Monitoring Durable Subscribers for Topics

You can monitor statistics on all the durable subscribers that are running on your JMS topics. WebLogic JMS stores durable subscribers in a persistent disk-based file store or JDBC-accessible database until the message has been delivered to the subscribers or has expired, even if those subscribers are not active at the time that the message is delivered.

1. Expand the JMS →Servers node.
2. The JMS Servers table displays in the right pane showing all the JMS servers defined in your domain.
3. Click the JMS server that you want to monitor from the JMS server list in the left pane, or from the JMS Servers table displayed in the right pane.
4. Click the Monitoring tab to display the monitoring data.
5. Click the Monitor all Active JMS Destinations text link. A table displays in the right pane showing all active JMS destinations for the current domain, with the Durable Subscriber Runtimes column listing the number of durable subscribers running (if any) for the destination topics listed in the table.
6. To view durable subscriber information for a specific topic, click the icon (or actual number) in the Durable Subscriber Runtimes column for the desired topic.

7. Click the Monitor all Durable Subscribers text link. Durable Subscribers information is displayed in a table in the right pane for all durable subscribers running (if any) for the destination topics listed in the table, as described in [“Durable Subscribers” on page 180-1](#).

Monitoring Active JMS Session Pools

You can monitor statistics on all the active JMS session pools defined for a JMS server. Session pools enable an application to process messages concurrently.

1. Expand the JMS node →Servers node.
2. The JMS Servers table displays in the right pane showing all the JMS servers defined in your domain.
3. Click the JMS server that you want to monitor from the JMS server list, or from the JMS Servers table displayed in the right pane.
4. Click the Monitoring tab to display the monitoring data.
5. Click the Monitor all Active JMS Session Pools Runtime text link. A table displays showing all active JMS session pools for the current domain, as described in [“Active JMS Session Pools” on page 201-1](#).

Monitoring Distributed Destination System Subscriptions and Proxy Topic Members

In certain distributed destination configurations for Weblogic JMS, the distributed destination may automatically create *proxy topic members* or *system subscriptions* between the topic or queue members. If this occurs, system subscriptions and proxy topic members will appear in MBean statistics, as well as in the Administration Console, when monitoring distributed destination members, as described in [“Active JMS Destinations” on page 164-1](#). They may also appear in the durable subscription names and in the consumer counts for the distributed destination members.

The following points describe the behavior of system subscriptions and proxy topic members:

- **Distributed Topic Proxy Members** — A WebLogic Server instance that has a configured JMS connection factory, but which has not been configured to host a local distributed topic member for a remote distributed topic, may automatically create and host a local proxy topic member for the remote distributed topic. This occurs when the first non-durable subscription for the distributed topic is created on the server's connection factory. The dynamically-created proxy topic member resides within a dynamically-created JMS server. Each of the manually-configured distributed topic members will create a system subscription for each dynamically-created proxy topic member. The non-durable consumers are then created on the proxy topic member.
- **Distributed Topic System Subscriptions** — System subscriptions are used to forward messages between configured distributed destination members. For example, when there are *n* members in a distributed topic, each member has at least *n*-1 system subscribers. In addition, for each proxy topic member, there will also be a system subscription on each distributed topic member.
- **Distributed Queue System Subscriptions** — Distributed queue members that have enabled the *Forward Delay* attribute on the distributed queue (by changing the default value of -1 seconds), may also create system subscribers. The system subscribers are used to forward messages from queue members with no consumers to queue members that do have consumers.
- **Durable System Subscriptions** — When a JMS file or JDBC store is configured for a distributed topic member, system subscriptions are created as durable subscribers. They are displayed by name in the Administration Console.

1 JMS: Tuning

[“Attributes and Console Screen Reference for JMS” on page 233-1]

The following sections explain how to get the most out of your applications by implementing the administrative performance tuning features available with WebLogic JMS:

- [“Overview” on page 235-2](#)
- [“Improving JMS File Store Performance” on page 235-2](#)
- [“Tuning Distributed Destinations” on page 235-5](#)
- [“Paging Out Messages To Free Up Memory” on page 235-8](#)
- [“Controlling the Flow of Messages on JMS Servers and Destinations” on page 235-20](#)
- [“Avoiding Quota Exceptions by Blocking Message Producers” on page 235-24](#)
- [“Handling Expired Messages” on page 235-27](#)

You may also want to refer these WebLogic JMS and Messaging Bridge sections:

- [“JMS: Configuring” on page 232-1](#)
- [“JMS: Monitoring” on page 234-1](#)
- [“Messaging Bridge” on page 289-1](#)

Overview

The following sections explain how to get the most out of your applications by implementing the administrative performance tuning features available with WebLogic JMS.

- Improving JMS File Store Performance
- “Tuning Distributed Destinations” on page 235-5
- Paging Out Messages To Free Up Memory
- “Controlling the Flow of Messages on JMS Servers and Destinations” on page 235-20
- “Avoiding Quota Exceptions by Blocking Message Producers” on page 235-24
- “Handling Expired Messages” on page 235-27

Improving JMS File Store Performance

By default, WebLogic JMS file stores guarantee up-to-the-message integrity by using synchronous writes to a disk. Disabling the default Synchronous Writes Policy of “Cache-Flush” improves file store performance, often quite dramatically, but at the expense of possibly losing sent messages or generating duplicate received messages (even if messages are transactional) in the event of an operating system crash or a hardware failure. Simply shutting down an operating system does not generate these failures, as an operating system flushes all outstanding writes during a normal shutdown. Instead, these failures can be emulated by shutting the power off to a busy server.

Note: The Synchronous Write Policy is ignored if the file store is used exclusively for paging non-persistent messages to disk.

Disabling the Default Synchronous Writes Policy

To modify the default Cache-Flush policy:

1. Expand the JMS →Stores node. The JMS Stores table displays in the right pane showing all the JMS stores.
2. Select the JMS file store that you want to modify. A dialog shows the tabs associated with modifying an existing file store.
3. On the Configuration tab, select a Synchronous Write Policy to determine how this JMS file store writes data to disk.
 - Disabled – Transactions are complete as soon as their writes are cached in memory, instead of waiting for the writes to successfully reach the disk. This policy is the fastest, but the least reliable (that is, transactionally safe).
 - Cache-Flush – Transactions cannot complete until all of their writes have been flushed down to disk. The default policy, it is reliable and scales well as the number of simultaneous users increases.
 - Direct-Write – File store writes are written directly to disk for Solaris and Windows operating systems. This policy's speed depends on whether on-disk cache is enabled for operating system.

For more detailed information about the Synchronous Write Policy attributes, see [“JMS File Store --> Configuration” on page 181-1](#).

4. Click Apply to save your change.

Disabling the On-Disk Cache For a Disk Drive on Windows 2000

Although the use of the Direct-Write policy is transactionally reliable on Solaris systems, Windows systems may leave transaction data in the on-disk cache without writing it to disk immediately. This is not considered to be transactionally reliable, since a power failure can cause loss of on-disk cache data — possibly resulting in lost and/or duplicate messages. For reliable writes using Direct-Write on Windows, either disable all write caching for the disk (enabled by default), or use a disk with a battery-backed cache. Some file systems, however, do not allow this value to be changed (for example, a RAID system that has a reliable cache).

1. Open the Control Panel -> System Properties dialog box.
2. Select the Hardware tab.

3. Click the Device Manager button.
4. Expand the Disk Drives node on the Device Manager tree.
5. Double-click the *drive name* that you want to modify.
6. On the Properties dialog box, select the Disk Properties tab.
7. Clear the Write Cache Enabled check box.

Comparing Synchronous Write Policy Settings

The following tables compare the synchronous write policies with respect to reliability, performance, and scalability. Use the following key to interpret the expected results based on your synchronous write policy settings.

- **Disk Cache On/Off:** the on-disk write cache is enabled/disabled
- **1-m Perf:** very few clients
- **m-m Perf:** many clients
- **M-M Perf:** a large amount of concurrent clients
- **Reliability Low/High:** High reliability is needed for exactly-once (transactional) messaging

Table 235-1 Relative Performance (compare within same column)

Policy	Disk Cache	1-m Perf	m-m Perf	M-M Perf	Reliability
Disabled	On	****	****	****	Low (depends on OS cache)
	Off	****	****	****	Low
Cache-Flush	On	*	**	***	High
	Off	*	**	***	High
Direct-Write	On	**	***	**	Medium (High with reliable disk-cache)
	Off	**	*	*	High

Table 235-2 Relative Scalability (compare within same row)

Policy	Disk Cache	1-m Perf	m-m Perf	M-M Perf
Disabled	On	****	****	****
	Off	****	****	****
Cache-Flush	On	*	**	***
	Off	*	**	***
Direct-Write	On	*	**	***
	Off	*	*	*

Tuning Distributed Destinations

The following sections describe how to tune a distributed destination by configuring attributes on a JMS connection factory.

- “Configuring Message Load Balancing Across a Distributed Destination” on page 235-6
- “Configuring Server Affinity For a Distributed Destination” on page 235-7

For more information about configuring a distributed destination, see “JMS Distributed Destination Tasks” on page 232-31.

Configuring Message Load Balancing Across a Distributed Destination

The Load Balancing Enabled attribute on the JMS Connection Factory → Configuration → General tab defines whether non-anonymous producers created through a connection factory are load balanced within a distributed destination on a per-call basis. Applications that use distributed destinations to distribute or balance their producers and consumers across multiple physical destinations, but do not want to make a load balancing decision each time a message is produced, can turn off the Load Balancing Enabled attribute.

To ensure a fair distribution of the messaging load among a distributed destination, the initial physical destination (queue or topic) used by producers is always chosen at random from among the distributed destination members.

To configure load balancing on a connection factory:

1. Expand the JMS node.
2. Click the Connection Factories node. The JMS Connection Factories table displays all the connection factories defined in your domain.
3. Click the connection factory on which you want to establish message load balancing. A dialog displays in the right pane showing the tabs associated with modifying a connection factory.
4. Define the setting of the Load Balancing Enabled attribute on the General tab using the following guidelines:
 - `Load Balancing Enabled = True`
For `QueueSender.send()` methods, non-anonymous producers are load balanced on *every* invocation across the distributed queue members.

For `TopicPublisher.publish()` methods, non-anonymous producers are always pinned to the same physical topic for every invocation, irrespective of the Load Balancing Enabled setting.
 - `Load Balancing Enabled = False`
Producers always produce to the same physical destination until they fail. At that point, a new physical destination is chosen.
5. Click Apply to save your changes.

Note: Depending on your implementation, the setting of the Server Affinity Enabled attribute can affect load balancing preferences for distributed destinations. For more information, see [“How Distributed Destination Load Balancing Is Affected When Using the Server Affinity Enabled Attribute”](#) in *Programming WebLogic JMS*.

Anonymous producers (producers that do not designate a destination when created), are load-balanced each time they switch destinations. If they continue to use the same destination, then the rules for non-anonymous producers apply (as stated previously).

For more information about how message load balancing takes place among the members of a distributed destination, see [“Load Balancing Messages Across a Distributed Destination”](#) in *Programming WebLogic JMS*.

Configuring Server Affinity For a Distributed Destination

The Server Affinity Enabled attribute on the JMS Connection Factory → Configuration → General tab defines whether a WebLogic Server that is load balancing consumers or producers across multiple physical destinations in a distributed destination set, will first attempt to load balance across any other physical destinations that are also running on the same WebLogic Server.

Note: The Server Affinity Enabled attribute does not affect queue browsers. Therefore, a queue browser created on a distributed queue can be pinned to a remote distributed queue member even when Server Affinity is enabled.

To disable server affinity on a connection factory:

1. Expand the JMS node.
2. Click the 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 connection factory on which you want to disable server affinity. A dialog displays in the right pane showing the tabs associated with modifying a connection factory.
4. Define the Server Affinity Enabled attribute in the General tab.

- If the Server Affinity Enabled check box is selected (True), then a WebLogic Server that is load balancing consumers or producers across multiple physical destinations in a distributed destination set, will first attempt to load balance across any other physical destinations that are also running on the same WebLogic Server.
 - If the Server Affinity Enabled check box is not selected (False), then a WebLogic Server will load balance consumers or producers across physical destinations in a distributed destination set and disregard any other physical destinations also running on the same WebLogic Server.
5. Click Apply to save your changes.

For more information about how the Server Affinity Enabled setting affects the load balancing among the members of a distributed destination, see [“How Distributed Destination Load Balancing Is Affected When Using the Server Affinity Enabled Attribute”](#) in *Programming WebLogic JMS*.

Paging Out Messages To Free Up Memory

With the *message paging* feature, you can free up virtual memory during peak message load periods. This feature can greatly benefit applications with large message spaces.

JMS message paging saves memory for both persistent and non-persistent messages, as even persistent messages cache their data in memory. Paged persistent messages continue to be written to the regular backing store (file or database); and paged non-persistent messages are written to the JMS server’s message paging store, which is configured separately.

A paged-out message does not free all of the memory that it consumes. The message header and message properties remain in memory for use with searching, sorting, and filtering.

Note: Messages sent in a transacted session are only eligible for paging *after* the session is committed. Prior to that, the message will only be held in memory; therefore, the heap size of the Java Virtual Machine (JVM) should be appropriately tuned to accommodate the projected peak amount of client load

from all active sessions until they are committed. For more information on tuning your heap size, see “[Tuning Java Virtual Machines \(JVMs\)](#)” in *WebLogic Performance and Tuning*.

Configuring Message Paging

Unless paging is configured and enabled, all messages (even persistent ones) are held in memory. You can configure paging for a JMS server and/or specific destinations. Using the paging attributes on the JMS Server node you can specify a paging store for a JMS server, enable bytes and/or messages paging, and configure bytes/messages high and low thresholds to start and stop paging. Similarly, using the paging attributes on the Destinations node, you can configure bytes/messages paging for all topics and queues configured on a JMS server. The destinations use a dedicated paging store that is configured for the JMS server.

Note: Message paging is not enabled by default. However, a message paging store will be automatically created when either bytes paging or messages paging is enabled on the JMS server or it’s destinations without preconfiguring a message paging store.

Also, if you use JMS templates to configure multiple destinations, you can use the attributes on the Templates node to configure paging quickly on all your destinations. To override a template’s paging configuration for specific destinations, you can enable or disable paging on any destination.

For instructions on configuring a new JMS server, destinations (topics and queues), and templates, see “JMS Server Tasks” on page 232-5, “JMS Queue and Topic Destination Tasks” on page 232-14, and “JMS Template Tasks” on page 232-18.

Note: For performance tuning purposes, you can modify the paging thresholds to any legal value at any time. Once paging is enabled, however, you cannot dynamically disable it by resetting a byte or message threshold back to -1. To prevent paging from occurring, set the byte/message high threshold to a very large number (maximum is $2^{63} - 1$), so that paging is not triggered.

Configuring a Message Paging Store for a JMS Server

Each JMS server must have its own paging store, which is used exclusively for paging out non-persistent messages for the JMS server and its destinations. It's best to use a JMS file store rather than a JMS JDBC store, as the JDBC store will perform poorly in comparison without any real benefit.

You can create a message paging store manually, or one will be created automatically when either bytes paging or messages paging is enabled on the JMS server or its destinations without preconfiguring a message paging store.

To configure a new message paging store:

1. Expand the JMS →Stores node. The right pane shows all the JMS stores.
2. Click the Create a new JMS File Store text link. The right pane shows the tabs associated with configuring a new JMS file store.
3. On the Configuration tab, define the file store attributes:
 - Enter a name for the message paging file store. This name must be unique within the WebLogic Server instance or its cluster.
 - When a file store is used exclusively for paging out messages to disk, the Synchronous Write Policy is ignored.
 - Enter the pathname to the directory on the file system where the file store is kept. (This directory must exist on your system, so be sure to create it before completing this tab.)

For more information about the JMS file store attributes, see [“JMS File Store --> Configuration” on page 181-1](#).

4. Click Create to create a JMS 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.
5. If you have multiple JMS servers in your domain, repeat steps 3-5 for each server instance.

Configuring Message Paging on a JMS Server

To enable and configure paging on an existing JMS server:

1. Click the JMS Servers node. The right pane shows all the servers defined in your domain.
2. Click the server that you want to configure for paging. The right pane shows the tabs associated with configuring the server.
3. On the General tab, use the Paging Store list box to select the JMS file store that you configured for paged messages. Click Apply to save your changes.

For instructions on configuring a paging store, refer to “Configuring a Message Paging Store for a JMS Server” on page 235-10.

4. On the Thresholds & Quotas tab, configure bytes paging:
 - Select the Bytes Paging Enabled check box.
 - In the Bytes Threshold High field, enter an amount that will start bytes paging when the number of bytes on the JMS server exceeds this threshold.
 - In the Bytes Threshold Low field, enter an amount that will stop bytes paging once the number of bytes on the JMS server falls below this threshold.
5. On the Thresholds & Quotas tab, configure messages paging:
 - Select the Messages Paging Enabled check box.
 - In the Messages Threshold High field, enter an amount that will start messages paging when the number of messages on the JMS server exceeds this threshold.
 - In the Messages Threshold Low field, enter an amount that will stop messages paging once the number of messages on the JMS server falls below this threshold.
6. Click Apply to save the new bytes and/or messages paging values.
7. Repeat steps 2–6 to configure paging for additional JMS servers in the domain.

Note: Each JMS server must have its own paging store.
8. After you configure your JMS server (or servers) for paging, do one of the following:
 - If you are not configuring the JMS server’s destinations for paging, reboot WebLogic Server to activate paging.

- If you are configuring the JMS server's destinations for paging, follow refer to either "Configuring Message Paging on a JMS Template" on page 235-12 or "Configuring Message Paging on Destinations" on page 235-13.

Configuring Message Paging on a JMS Template

JMS templates provide an efficient way to define multiple destinations (topics or queues) with similar attribute settings. To configure paging on a template for destinations, do the following:

1. Click the JMS node in the left pane.
2. Click the JMS Templates node. The right pane shows all the templates defined in the domain.
3. Click the template that you want to configure for paging. The right pane shows the tabs associated with configuring the template.
4. On the Thresholds & Quotas tab, configure bytes paging:
 - Select the Bytes Paging Enabled check box.
 - In the Bytes Threshold High field, enter an amount that will start bytes paging when the number of bytes on the JMS server exceeds this threshold.
 - In the Bytes Threshold Low field, enter an amount that will stop bytes paging once the number of bytes on the JMS server falls below this threshold.
5. On the Thresholds & Quotas tab, configure messages paging:
 - Select the Messages Paging Enabled check box.
 - In the Messages Threshold High field, enter an amount that will start messages paging when the number of messages on the JMS server exceeds this threshold.
 - In the Messages Threshold Low field, enter an amount that will stop messages paging once the number of messages on the JMS server falls below this threshold.
6. Click Apply to save the new bytes and/or messages paging values.
7. Repeat steps 3–6 to configure paging for additional JMS templates.

8. After configuring all of your JMS templates for paging, reboot WebLogic Server to activate paging.

Configuring Message Paging on Destinations

Follow these directions if you are configuring paging on destinations without using a JMS template.

1. Under JMS Servers, expand a server instance that is already configured for paging.
2. Click the Destinations node. The right pane shows all of the server's topics and queues.
3. Click the topic or queue that you want to configure for paging. The right pane shows the tabs associated with configuring the topic or queue.
4. On the Thresholds & Quotas tab, configure bytes paging:
 - Select the Bytes Paging Enabled check box.
 - In the Bytes Threshold High field, enter an amount that will start bytes paging when the number of bytes on the JMS server exceeds this threshold.
 - In the Bytes Threshold Low field, enter an amount that will stop bytes paging once the number of bytes on the JMS server falls below this threshold.
5. On the Thresholds & Quotas tab, configure messages paging:
 - Select the Messages Paging Enabled check box.
 - In the Messages Threshold High field, enter an amount that will start messages paging when the number of messages on the JMS server exceeds this threshold.
 - In the Messages Threshold Low field, enter an amount that will stop messages paging once the number of messages on the JMS server falls below this threshold.
6. Click Apply to save the new bytes and/or messages paging values.
7. Repeat steps 3–6 to configure paging for additional JMS destinations.
8. After you configure all your destinations for paging, reboot WebLogic Server to activate paging.

Note: If you use JMS templates to configure your destinations, a destination's explicit Byte/Messages Paging configuration overrides the template's configuration. For more information, refer to “Configuring a Destination to Override Message Paging on a JMS Template” on page 235-14 and to “JMS Template Tasks” on page 232-18.

Configuring a Destination to Override Message Paging on a JMS Template

Follow these directions if you want to override a template's settings and enable or disable paging on a specific destination.

1. Under JMS Servers, expand a server instance that is already configured for paging.
2. Click the Destinations node. The right pane shows all of the server's topics and queues.
3. Click the topic or queue that you want to configure for paging. The right pane shows the topics or queues associated with the server instance.
4. On the Thresholds & Quotas tab, configure the Bytes Paging Enabled and/or Messages Paging Enabled attributes on the destination according to how you want to override the JMS template for the destination.
 - To disable paging for the destination, select False in the Bytes Paging Enabled and/or the Messages Paging Enabled list boxes.
 - To enable paging for the destination, select True in the Bytes Paging Enabled and/or the Messages Paging Enabled list boxes.
5. Click Apply to save the new bytes and/or messages paging values.
6. Repeat steps 2–5 to configure paging for additional JMS destinations on the same server instance.
7. Once all of your destinations are configured for paging, then reboot WebLogic Server to activate paging.

JMS Message Paging Attributes

The following sections briefly describe the message paging attributes available with WebLogic Server JMS.

JMS Server Message Paging Attributes

Table 235-3 describes the message paging attributes that you define when configuring paging on a JMS Server. For detailed information about other JMS Server attributes, and the valid and default values for them, see “JMS Template Tasks” on page 232-18.

Table 235-3 JMS Server Attributes

Attribute	Description
Bytes Paging Enabled	<ul style="list-style-type: none"> ■ If the Bytes Paging Enabled check box is not selected (False), then server bytes paging is explicitly disabled. ■ If the Bytes Paging Enabled check box is selected (True), a paging store has been configured, and both the Bytes Threshold Low and Bytes Threshold High attributes 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 the Bytes Paging Enabled check box is selected (True).
Messages Paging Enabled	<ul style="list-style-type: none"> ■ If the Messages Paging Enabled check box is not selected (False), then server messages paging is explicitly disabled. ■ If the Messages Paging Enabled check box is selected (True), a paging store has been configured, and both the Messages Threshold Low and Messages Threshold High attributes are greater than -1, then server messages paging is enabled. ■ If either the Messages Threshold Low or Messages Threshold High attribute is undefined, or defined as -1, then server paging is implicitly disabled—even though the Messages Paging Enabled check box is selected (True).

Table 235-3 JMS Server Attributes

Attribute	Description
Paging Store	<p>The name of the persistent store where non-persistent messages are paged. A paging store cannot be the same store used for persistent messages or durable subscribers.</p> <p>Two JMS servers cannot use the same paging store; therefore, you must configure a unique paging store for each server.</p>

JMS Template Message Paging Attributes

Table 235-4 describes the message paging attributes that you define when configuring paging on JMS templates for destinations. For detailed information about other JMS template attributes, and the valid and default values for them, see “JMS Template Tasks” on page 232-18.

Table 235-4 JMS Template Attributes

Attribute	Description
Bytes Paging Enabled	<ul style="list-style-type: none"> ■ If the Bytes Paging Enabled check box is not selected (False), then destination-level bytes paging is disabled for the JMS template’s destinations—unless the destination setting overrides the template. ■ If the Bytes Paging Enabled check box is selected (True), a paging store has been configured for the JMS Server, and both the Bytes Threshold Low and Bytes Threshold High attributes are greater than -1, then destination-level bytes paging is enabled for the JMS template’s destinations—unless the destination setting overrides the template. ■ If no value is defined in the JMS Template MBean, then the value defaults to False and bytes paging is disabled for the JMS template’s destinations.

Table 235-4 JMS Template Attributes

Attribute	Description
Messages Paging Enabled	<ul style="list-style-type: none">■ If the Messages Paging Enabled check box is not selected (False), then destination-level messages paging is disabled for the template's destination—unless the destination setting overrides the template.■ If the Messages Paging Enabled check box is selected (True), a paging store has been configured for the JMS Server, and both the Messages Threshold Low and Messages Threshold High attributes are greater than -1, then destination-level messages paging is enabled for this destination—unless the destination setting overrides the template.■ If no value is defined in the JMS Template MBean, then the value defaults to False and messages paging is disabled for the template's destinations.

JMS Destination Message Paging Attributes

Table 235-5 describes the attributes that you define when configuring message paging on destinations. For detailed information about other JMS destination attributes, and valid and default values for them, see “JMS Queue and Topic Destination Tasks” on page 232-14.

Table 235-5 JMS Destination Attributes

Attribute	Description
Bytes Paging Enabled	<ul style="list-style-type: none"> ■ If Bytes Paging Enabled is set to False, then destination-level bytes paging is disabled for this destination. ■ If Bytes Paging Enabled is set to True, a paging store has been configured for the JMS Server, and both the Bytes Threshold Low and Bytes Threshold High attributes are greater than -1, then destination-level bytes paging is enabled for this destination. ■ If Bytes Paging Enabled is set to Default, then this value inherits the template's value—if a template is specified. If no template is configured for the destination, then the Default value is equivalent to False.
Messages Paging Enabled	<ul style="list-style-type: none"> ■ If Messages Paging Enabled is set to False, then destination-level messages paging is disabled for this destination. ■ If Messages Paging Enabled is set to True, a paging store has been configured for the JMS Server, and both the Messages Threshold Low and Messages Threshold High attributes are greater than -1, then destination-level messages paging is enabled for this destination. ■ If Messages Paging Enabled is set to Default, then this value inherits the template's value—if a template is specified. If no template is configured for the destination, then the Default value is equivalent to False.

Note: If server paging is enabled, and destination-level paging is disabled for a given destination, then messages on the destination can still be paged if server paging is triggered. However, when destination-level paging is disabled for a given destination, then the destination's high thresholds will not force the destination to page out messages when they are exceeded.

Message Paging Threshold Attributes

Table 235-6 briefly describes the bytes and messages paging thresholds available with JMS servers, templates, and destinations. For detailed information about other JMS server, template, and destination attributes, and the valid and default values for them, see “JMS Server Tasks” on page 232-5, “JMS Queue and Topic Destination Tasks” on page 232-14, and “JMS Template Tasks” on page 232-18.

Table 235-6 Paging Threshold Attributes

Attribute	Description
Bytes Threshold High	Start paging when the number of bytes exceeds this threshold.
Bytes Threshold Low	Stop paging when the number of bytes falls back below this threshold.
Messages Threshold High	Start paging when the number of messages exceeds this threshold.
Messages Threshold Low	Stop paging when the number of messages falls back below this threshold.

The thresholds are defined for servers, templates, and destinations as follows:

- If either bytes high/low threshold value is not defined (or is defined as -1), then the number of bytes is not used to determine when and what to page.
- If either messages high/low threshold value is not defined (or is defined as -1), then the number of messages is not used to determine when and what to page.
- A server or template/destination must have the Bytes/Messages Paging Enabled attribute set to True in order for paging to take place. If the thresholds are set, but paging is not enabled, messages are still logged on the server indicating threshold conditions.

Related Topics

For more information on tuning Weblogic JMS performance, refer to the following help topics:

- “Controlling the Flow of Messages on JMS Servers and Destinations” on page 235-20
- “Avoiding Quota Exceptions by Blocking Message Producers” on page 235-24
- “Handling Expired Messages” on page 235-27

Controlling the Flow of Messages on JMS Servers and Destinations

With the Flow Control feature, you can direct a JMS server or destination to slow down message producers when it determines that it is becoming overloaded.

The following sections describe how flow control feature works and how to configure flow control on a connection factory.

- “How Flow Control Works” on page 235-20
- “Configuring Flow Control” on page 235-21
- “Flow Control Thresholds” on page 235-23
- “Avoiding Quota Exceptions by Blocking Message Producers” on page 235-24

How Flow Control Works

Specifically, when either a JMS server or its destinations exceeds its specified byte or message threshold, it becomes *armed* and instructs producers to limit their message flow (messages per second).

Producers will limit their production rate based on a set of flow control attributes configured for producers via the JMS connection factory. Starting at a specified *flow maximum* number of messages, a producer evaluates whether the server/destination is still armed at prescribed intervals (for example, every 10 seconds for 60 seconds). If at each interval, the server/destination is still armed, then the producer continues to move its rate down to its prescribed *flow minimum* amount.

As producers slow themselves down, the threshold condition gradually corrects itself until the server/destination is *unarmed*. At this point, a producer is allowed to increase its production rate, but not necessarily to the maximum possible rate. In fact, its message flow continues to be controlled (even though the server/destination is no longer armed) until it reaches its prescribed *flow maximum*, at which point it is no longer flow controlled.

Configuring Flow Control

Producers receive a set of flow control attributes from their session, which receives the attributes from the connection, and which receives the attributes from the connection factory. These attributes allow the producer to adjust its message flow.

Specifically, the producer receives attributes that limit its flow within a minimum and maximum range. As conditions worsen, the producer moves toward the minimum; as conditions improve, the producer moves toward the maximum. Movement toward the minimum and maximum are defined by two additional attributes that specify the rate of movement toward the minimum and maximum. Also, the need for movement toward the minimum and maximum is evaluated at a configured interval.

To configure message flow control on a connection factory, follow these steps:

1. Under the Services →JMS node in the navigation tree, click the Connection Factory node. The right pane shows the connection factories defined in the domain
2. Select the connection factory instance for which you will configure message flow control. The right pane shows the tabs associated with configuring the connection factory.
3. Select the Configuration →Flow Control tab.
4. Define the Flow Control attributes as described in following table:

Table 235-7 Flow Control Attributes

Attribute	Description
Flow Control Enabled	Determines whether a producer can be flow controlled by the JMS server.

Table 235-7 Flow Control Attributes

Attribute	Description
Flow Maximum	<p>The maximum number of messages per second for a producer that is experiencing a threshold condition.</p> <p>If a producer is not currently limiting its flow when a threshold condition is reached, the initial flow limit for that producer is set to Flow Maximum. If a producer is already limiting its flow when a threshold condition is reached (the flow limit is less than Flow Maximum), then the producer will continue at its current flow limit until the next time the flow is evaluated.</p> <p>Once a threshold condition has subsided, the producer is not permitted to ignore its flow limit. If its flow limit is less than the Flow Maximum, then the producer must gradually increase its flow to the Flow Maximum each time the flow is evaluated. When the producer finally reaches the Flow Maximum, it can then ignore its flow limit and send without limiting its flow.</p>
Flow Minimum	<p>The minimum number of messages per second for a producer that is experiencing a threshold condition. This is the lower boundary of a producer's flow limit. That is, WebLogic JMS will not further slow down a producer whose message flow limit is at its Flow Minimum.</p>
Flow Interval	<p>An adjustment period of time, defined in seconds, when a producer adjusts its flow from the Flow Maximum number of messages to the Flow Minimum amount, or vice versa.</p>

Table 235-7 Flow Control Attributes

Attribute	Description
Flow Steps	<p>The number of steps used when a producer is adjusting its flow from the Flow Minimum amount of messages to the Flow Maximum amount, or vice versa. Specifically, the Flow Interval adjustment period is divided into the number of Flow Steps (for example, 60 seconds divided by 6 steps is 10 seconds per step).</p> <p>Also, the movement (that is, the rate of adjustment) is calculated by dividing the difference between the Flow Maximum and the Flow Minimum into steps. At each Flow Step, the flow is adjusted upward or downward, as necessary, based on the current conditions, as follows:</p> <ul style="list-style-type: none">■ The downward movement (the decay) is geometric over the specified period of time (Flow Interval) and according to the specified number of Flow Steps. (For example, 100, 50, 25, 12.5).■ The movement upward is linear. The difference is simply divided by the number of Flow Steps.

5. Click Apply to store new attribute values.

For detailed information about other connection factory attributes, and the valid and default values for them, see “JMS Connection Factory Tasks” on page 232-8.

Flow Control Thresholds

The attributes used for configuring bytes/messages thresholds are defined as part of the JMS server and/or its destination. Table 235-8 defines how the upper and lower thresholds start and stop flow control on a JMS server and/or JMS destination.

Table 235-8 Flow Control Threshold Attributes

Attribute	Description
Bytes/Messages Threshold High	When the number of bytes/messages exceeds this threshold, the JMS server/destination becomes armed and instructs producers to limit their message flow.

Table 235-8 Flow Control Threshold Attributes

Attribute	Description
Bytes/Messages Threshold Low	<p>When the number of bytes/messages falls below this threshold, the JMS server/destination becomes unarmed and instructs producers to begin increasing their message flow.</p> <p>Flow control is still in effect for producers that are below their message flow maximum. Producers can move their rate upward until they reach their flow maximum, at which point they are no longer flow controlled.</p>

For detailed information about other JMS server and JMS destination attributes, and the valid and default values for them, see “JMS Server Tasks” on page 232-5 and “JMS Queue and Topic Destination Tasks” on page 232-14.

Avoiding Quota Exceptions by Blocking Message Producers

The following sections describe two additional “Flow Control” features that can help you to avoid receiving message quota errors by temporarily blocking message producers from sending messages to a destination (queue or topic) when it has exceeded its specified maximum message quota.

- A Send Timeout value can be set for a JMS connection factory to provide message producers the option of waiting a specified length of time (in milliseconds) until space becomes available on a destination that has exceeded its message quota.
- The Blocking Send Policy enables you to define a JMS server’s blocking behavior when multiple producers are competing for space on a destination that has exceeded its message quota.

For instructions on configuring the Blocking Send Policy and Send Timeout features:

- “Defining a Send Timeout on a JMS Connection Factory” on page 235-25.

- “Specifying a Blocking Send Policy on a JMS Server” on page 235-26.

Defining a Send Timeout on a JMS Connection Factory

The Send Timeout feature provides more control over message send operations by giving producers the option of waiting a user-specified length of time until space becomes available on a destination. For example, if a producer makes a request and there is insufficient space, then the producer is blocked until space becomes available, or the operation times out.

Follow these directions to define how long a JMS connection factory will block message requests when a destination exceeds its maximum quota.

1. Under the Services →JMS node in the navigation tree, click the Connection Factory node. The right pane shows the connection factories defined in the domain.
2. Select the connection factory instance for which you will configure a send timeout. The right pane shows the tabs associated with configuring the connection factory.
3. Select the Configuration →Flow Control tab.
4. In the Send Timeout field, enter the amount of time, in milliseconds, a sender will block messages when there is insufficient space on the message destination. Once the specified waiting period ends, one of the following results will occur:
 - If sufficient space becomes available before the timeout period ends, the operation continues.
 - If sufficient space does not become available before the timeout period ends, you receive a “resource allocation” exception.

If you choose not to enable the blocking send policy by setting this value to 0, then you will receive a “resource allocation” exception whenever sufficient space is not available on the destination.

For more information about the Send Timeout attribute, see “JMS Connection Factory --> Configuration --> Flow Control” on page 154-1.

5. Click Apply to save the new value.

Specifying a Blocking Send Policy on a JMS Server

The Blocking Send policies enable you to define the JMS server's blocking behavior on whether to deliver smaller messages before larger ones when multiple message producers are competing for space on a destination that has exceeded its message quota.

Follow these directions to define how a JMS server will block message requests when its destinations are at maximum quota.

1. Under the Services →JMS node in the navigation tree, click the Servers node. The right pane shows the JMS servers defined in the domain.
2. Select the server instance for which you will configure a blocking send policy. The right pane shows the tabs associated with configuring the server.
3. Select the Configuration →Thresholds & Quota tab.
4. From the Blocking Send Policy list box, select one of the following options:
 - FIFO — All send requests for the same destination are queued up one behind the other until space is available. No send request is permitted to complete when there another send request is waiting for space before it.
 - Preemptive — A send operation can preempt other blocking send operations if space is available. That is, if there is sufficient space for the current request, then that space is used even if there are previous requests waiting for space.

For more information about the Blocking Send Policy attribute, see “JMS Server --> Configuration --> Thresholds & Quotas” on page 193-1.

5. Click Apply to save the new value.

Related Topics

For more information on tuning Weblogic JMS performance, refer to the following help topics:

- “Controlling the Flow of Messages on JMS Servers and Destinations” on page 235-20

- “Paging Out Messages To Free Up Memory” on page 235-8
- “Handling Expired Messages” on page 235-27

Handling Expired Messages

The following sections describe two message expiration features, the message Expiration Policy and the Active Expiration of message, which provide more control over how the system searches for expired messages and how it handles them when they are encountered.

In previous releases, WebLogic JMS implemented a passive message expiration policy. Messages expired as they were discovered and were then simply discarded from the system. Since no active searches were made for expired messages, they could accumulate on the system and strain system resources. Active message expiration addresses this issue by ensuring that expired messages are cleaned up immediately. Moreover, expired message auditing gives you the option of tracking expired messages, either by logging when a message expires or by redirecting expired messages to a special destination.

- “Defining a Message Expiration Policy” on page 235-27
- “Enabling Active Message Expiration” on page 235-33

Defining a Message Expiration Policy

Use the message Expiration Policy feature to define an alternate action to take when messages expire. Using the Expiration Policy attribute on the Destinations node, an expiration policy can be set on a per destination basis. The Expiration Policy attribute defines the action that a destination should take when an expired message is encountered: discard the message, discard the message and log its removal, or redirect the message to an error destination.

Also, if you use JMS templates to configure multiple destinations, you can use the Expiration Policy attribute on the Templates node to quickly configure an expiration policy on all your destinations. To override a template’s expiration policy for specific destinations, you can modify the expiration policy on any destination.

For instructions on configuring the Expiration Policy, click one of the following links:

- “Configuring an Expiration Policy On a Topic” on page 235-28
- “Configuring an Expiration Policy On a Queue” on page 235-29
- “Configuring an Expiration Policy On a JMS Template” on page 235-30
- “Defining an Expiration Logging Policy” on page 235-31

Configuring an Expiration Policy On a Topic

Follow these directions if you are configuring an expiration policy on topics without using a JMS template. Expiration policies that are set on specific topics will override the settings defined on a JMS template.

1. Under Services →JMS →Servers in the navigation tree, expand the server instance that you want to configure.
2. Click the Destinations node. The right pane shows all of the server’s topics and queues.
3. Select the topic for which you will configure an expiration policy. The right pane shows the tabs associated with configuring the topic.
4. Click the Configuration →Expiration Policy tab.
5. From the Expiration Policy list box, select an expiration policy option.
 - none and Discard — If this topic has no JMS template, then “none” is equivalent to Discard. Discard means that expired messages are removed from the system. The removal is not logged and the message is not redirected to another location.
 - Log — Removes expired messages and writes an entry to the server log file indicating that the messages were removed from the system. You define the actual information that will be logged in the Expiration Logging Policy field in Step 6.
 - Redirect — Moves expired messages from their current location into the Error Destination defined for the topic.

For more information about the Expiration Policy options for a topic, see “JMS Topic --> Configuration --> Expiration Policy” on page 231-1.

6. If you selected the Log expiration policy in Step 5, use the Expiration Logging Policy field to define what information about the message is logged.

For more information about valid Expiration Logging Policy values, see “Defining an Expiration Logging Policy” on page 235-31.

7. Click Apply to save the new expiration policy values.
8. Repeat steps 3–7 to configure an expiration policy for additional topics.

Configuring an Expiration Policy On a Queue

Follow these directions if you are configuring an expiration policy on queues without using a JMS template. Expiration policies that are set on specific queues will override the settings defined on a JMS template.

1. Under Services → JMS → Servers in the navigation tree, expand the server instance that you want to configure.
2. Click the Destinations node. The right pane shows all of the server’s topics and queues.
3. Select the queue for which you will configure an expiration policy. The right pane shows the tabs associated with configuring the queue.
4. Click the Configuration → Expiration Policy tab.
5. From the Expiration Policy list box, select an expiration policy option.
 - none and Discard — If this queue has no JMS template, then “none” is equivalent to Discard. Discard means that expired messages are removed from the system. The removal is not logged and the message is not redirected to another location.
 - Log — Removes expired messages from the queue and writes an entry to the server log file indicating that the messages were removed from the system. You define the actual information that will be logged in the Expiration Logging Policy field described in Step 6.
 - Redirect — Moves expired messages from the queue and into the Error Destination defined for the queue.

For more information about the Expiration Policy options for a queue, see “JMS Queue --> Configuration --> Expiration Policy” on page 229-1.

6. If you selected the Log expiration policy in Step 5, use the Expiration Logging Policy field to define what information about the message is logged.

For more information about valid Expiration Logging Policy values, see “Defining an Expiration Logging Policy” on page 235-31.

7. Click Apply to save the new expiration policy values.
8. Repeat steps 3–7 to configure an expiration policy for additional queues.

Configuring an Expiration Policy On a JMS Template

Since JMS templates provide an efficient way to define multiple destinations (topics or queues) with similar attribute settings, you can configure a message expiration policy on an existing template (or templates) for your destinations.

1. Expand the Services → JMS → Templates node in the navigation tree. The right pane shows all the JMS templates defined in the domain (if any).
2. Select the template for which you will configure an expiration policy. The right pane shows the tabs associated with configuring the template.
3. Click the Configuration → Expiration Policy tab.
4. In the Expiration Policy list box, select an expiration policy option.
 - none and Discard — Expired messages are removed from the messaging system. The removal is not logged and the message is not redirected to another location. If “none” is defined for a given destination, then expired messages are discarded.
 - Log — Removes expired messages and writes an entry to the server log file indicating that the messages were removed from the system. The actual information that is logged is defined by the Expiration Logging Policy field described in Step 5.
 - Redirect — Moves expired messages from their current location into the Error Destination defined for the destination.

For more information about the Expiration Policy options, see “JMS Template --> Configuration --> Expiration Policy” on page 230-1.

5. If you selected the Log expiration policy in Step 4, use the Expiration Logging Policy field to define what information about the message is logged.

For more information about valid Expiration Logging Policy values, see “Defining an Expiration Logging Policy” on page 235-31.

6. Click Apply to save the new expiration policy values.
7. Repeat steps 2–6 to configure an expiration policy for additional JMS templates.

Defining an Expiration Logging Policy

When the Expiration Policy is set to Log, the Expiration Logging Policy defines what information about the message is logged. Valid values for Expiration Logging Policy properties include `%header%`, `%properties%`, JMS header properties as defined in the JMS specification, the WebLogic JMS-specific extended header fields `JMSDeliveryTime` and `JMSRedeliveryLimit`, and any user-defined property. Each property must be separated by a comma.

The `%header%` value indicates that all header fields should be logged. The `%properties%` value indicates that all user properties should be logged. Neither values are case sensitive. However, the enumeration of individual JMS header fields and user properties are case sensitive.

For example, you could specify one of the following values:

- `JMSPriority, Name, Address, City, State, Zip`
- `%header%, Name, Address, City, State, Zip`
- `JMSCorrelationID, %properties%`

The `JMSMessageID` field is always logged and cannot be turned off. Therefore, if the Expiration Policy is not defined (that is, none) or is defined as an empty string, then the output to the log file contains only the `JMSMessageID` of the message.

Expiration Log Output Format

When an expired message is logged, the text portion of the message (not including timestamps, severity, thread information, security identity, etc.) conforms to the following format:

```
<ExpiredJMSMessage JMSMessageId='$MESSAGEID' >
  <HeaderFields Field1='Value1' [Field2='Value2'] ... ] />
  <UserProperties Property1='Value1' [Property='Value2'] ... ] />
</ExpiredJMSMessage>
```

where `$MESSAGEID` is the exact string returned by `Message.getJMSMessageID()`.

For example:

```
<ExpiredJMSMessage JMSMessageID='ID:P<851839.1022176920343.0' >
  <HeaderFields JMSPriority='7' JMSRedelivered='false' />
  <UserProperties Make='Honda' Model='Civic' Color='White'
    Weight='2680' />
</ExpiredJMSMessage>
```

If no header fields are displayed, the line for header fields is not be displayed. If no user properties are displayed, that line is not be displayed. If there are no header fields and no properties, the closing `</ExpiredJMSMessage>` tag is not necessary as the opening tag can be terminated with a closing bracket (`/>`).

For example:

```
<ExpiredJMSMessage JMSMessageID='ID:N<223476.1022177121567.1' />
```

All values are delimited with double quotes. All string values are limited to 32 characters in length. Requested fields and/or properties that do not exist are not displayed. Requested fields and/or properties that exist but have no value (a null value) are displayed as null (without single quotes). Requested fields and/or properties that are empty strings are displayed as a pair of single quotes with no space between them.

For example:

```
<ExpiredJMSMessage JMSMessageID='ID:N<851839.1022176920344.0' >
  <UserProperties First='Any string longer than 32 char ...'
    Second=null Third='' />
</ExpiredJMSMessage>
```

Enabling Active Message Expiration

Use the Active Expiration feature to define the timeliness in which expired messages are removed from the destination to which they were sent or published. Messages are not necessarily removed from the system at their expiration time, but they are removed within a user-defined number of seconds. The smaller the window, the closer the message removal is to the actual expiration time.

Configuring a JMS Server to Actively Scan Destinations for Expired Messages

Follow these directions to define how often a JMS server will actively scan its destinations for expired messages. The default value is 30 seconds, which means the JMS server waits 30 seconds between each scan interval.

1. Under the Services → JMS node in the navigation tree, click the Servers node. The right pane shows the JMS servers defined in the domain.
2. Select the server instance for which you will configure an active expiration scan interval. The right pane shows the tabs associated with configuring the server.
3. Using the Scan Expiration Interval field on the Configuration → General tab, enter the amount of time, in seconds, that you want the JMS server to pause between its cycles of scanning its destinations for expired messages to process.

To disable active scanning, enter a value of 0 seconds. Expired messages are passively removed from the system as they are discovered.

For more information about the Expiration Scan Interval attribute, see “Configuring a JMS Server to Actively Scan Destinations for Expired Messages” on page 235-33.

4. Click Apply to save the new value.

Related Topics

For more information on tuning Weblogic JMS performance, refer to the following topics:

- “Paging Out Messages To Free Up Memory” on page 235-8

- “Controlling the Flow of Messages on JMS Servers and Destinations” on page 235-20
- “Avoiding Quota Exceptions by Blocking Message Producers” on page 235-24

Transaction Details

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The Transaction Details page shows details for the selected transaction. For transactions that may need to be manually resolved, you can click the Commit or Rollback options at the bottom of the page. These options are restricted by the overall status of the transaction. See [Table 237-1](#) for a list of valid states and permitted transaction resolutions.

Tasks

[“To Manually Resolve a Transaction” on page 237-9](#)

Related Topics

[“Manually Resolving Current \(Inflight\) Transactions” on page 237-6](#)

[“Monitoring Transactions” on page 237-5](#)

Attributes

Transaction Id—The transaction identifier assigned by the Transaction Manager.

Coordinator—The name of the server that acts as the transaction coordinator for the transaction. The server name is qualified by the domain in which the server runs.

Name—The transaction name as specified in the application that created the transaction.

Status—The overall status of the transaction. This status determines the allowable transaction resolution options: commit or rollback. For valid states and related transaction resolution options, see [Table 237-1](#).

Seconds Active—The number of seconds since the transaction was created.

Servers And Status—A list of all servers that participate in the transaction and the status for each server.

Resources—A list of resources that participate in the transaction and the status for each resource. A resource can be any XA-enabled resource registered with the Transaction Manager, such as a JDBC connection pool or a JMS store.

Properties—A list of transaction properties as defined by the application that created the transaction.

Force Local Rollback—Click this option to issue a rollback operation for the specified transaction on each participating resource that is registered on the current server. The transaction will then be removed from the local transaction manager's data structures.

Force Global Rollback—Click this option to issue a local rollback operation at each participating server for the specified transaction. If this option is invoked on a non-coordinating server, the coordinator will be contacted to process the operation. The coordinating server will issue asynchronous requests to each participant server.

Force Local Commit—Click this option to issue a commit operation for the specified transaction on each participating resource that is registered on the current server. The transaction will be removed from the local transaction manager's data structures. If the local server is the coordinator for the transaction, the commit record is released.

Force Global Commit—Click this option to issue a local commit operation at each participating server for the specified transaction. If this option is invoked on a non-coordinating server, the coordinator will be contacted to process the operation. The coordinating server will issue asynchronous requests to each participating server.

1 JTA

[“Attributes and Console Screen Reference for JTA” on page 238-1]

The following sections explain how to configure the Java Transaction API (JTA) attributes in WebLogic Server:

“Configuring Transactions” on page 237-1

“Configuring Domains for Inter-Domain Transactions” on page 237-3

“Monitoring Transactions” on page 237-5

“Transaction Log Files” on page 237-10

“Handling Heuristic Completions” on page 237-14

“Abandoning Transactions” on page 237-15

“Moving a Server to Another Machine” on page 237-16

“Transaction Recovery After a Server Fails” on page 237-16

“Overview of Transaction Management” on page 237-24

Also see “Attributes and Console Screen Reference for JTA” on page 238-1 for information about JTA pages in the Administration Console.

Configuring Transactions

Configuration settings for JTA (transactions) are applicable at the domain level. This means that configuration attribute settings apply to all servers within a domain. Monitoring and logging tasks for JTA are performed at the server level.

You can configure any transaction attributes before starting the server (static configuration) or, with one exception, while the server is running (dynamic configuration). The `TransactionLogFilePrefix` attribute must be set before starting the server.

Configuring JTA

1. In the left pane of the Administration Console, select the domain node (right below the word "console"). The Configuration tab for the domain is displayed by default.
2. Click the JTA tab.
3. Enter values in the Timeout Seconds, Abandon Timeout Seconds, Before Completion Iteration Limit, Max Transactions, Max Unique Name Statistics, and Checkpoint Interval Seconds attribute fields or accept the default values as assigned. For details about JTA attributes, see [“Attributes” on page 67-2](#).
4. Enable or disable the Forget Heuristics attribute as desired.
5. Click Apply to save any changes you made.
6. Ensure that the `TransactionLogFilePrefix` attribute is set appropriately when you configure the server. For more information about setting the logging attribute, see [“Transaction Log Files” on page 237-10](#).

Related Information

- Transaction attributes at [“Attributes” on page 67-2](#)
- [“Monitoring Transactions” on page 237-5](#)
- [“Transaction Log Files” on page 237-10](#)
- [“Configuring JDBC DataSources” on page 141-1](#)
- [“Emulating Two-Phase Commit” on page 141-3](#)
- [“Configuring JDBC Connection Pools” on page 127-1](#)

Configuring Domains for Inter-Domain Transactions

For a transaction manager to manage distributed transactions, the transaction manager must be able to communicate with all participating servers to prepare and then commit or rollback the transactions. This applies to cases when your WebLogic domain acts as the transaction manager or a transaction participant (resource) in a distributed transaction. The following sections describe how to configure your domain to enable inter-domain transactions. To learn more about interoperability between WebLogic domains, see “Enabling Trust Between WebLogic Domains” on page 428-47.

Limitations for Inter-Domain Transactions

Please note the following limitations for inter-domain transactions:

- Inter-domain transactions are not supported if you configure a data source/connection pool pair that includes the following attribute settings:
 - Emulate Two-Phase Commit is selected in the data source.
 - The connection pool uses a non-XA driver to create database connections.
- You cannot manually resolve incomplete transactions on resources from a WebLogic Server domain from WebLogic Server version 7.0 or earlier.

Inter-Domain Transactions for WebLogic Server 8.x and 7.x Domains

To manage or participate in transactions that span multiple WebLogic Server 8.x and 7.x domains (that is, all participating domains run on WebLogic Server 7.x or 8.x, or a combination of 7.x and 8.x), you must set a security credential for all domains to the same value. To set the credential value, follow these steps *for each participating domain*:

1. Start the Administration Console for one of the participating domains.
2. In the left pane, click the Security node to display Security settings for the domain.
3. Click the Configuration tab (if necessary), then the Advanced tab.
4. Clear the Enable Generated Credential check box and click Apply.
5. In the Credential field, enter the new credential, then re-enter it in the Confirm field. Enter the *same* credential for each domain and click Apply. If WebLogic Server 6.x domains will participate in distributed transactions, use the `system` password from the WebLogic Server 6.x domain.
6. Restart the administration server. You may also need to restart all managed servers.
7. Repeat steps 1 through 6 for each domain that participates in inter-domain transactions. In step 5, enter the *same* credential for each domain.

Inter-Domain Transactions Between WebLogic Server 7.x/8.x and WebLogic Server 6.x Domains

To manage transactions that use servers in both WebLogic Server 7.x or 8.x and WebLogic Server 6.x domains, you must do the following:

In all participating WebLogic Server 6.x domains:

- Change the password for the `system` user to the same value in all participating domains on the Security—Users tab in the Administration Console. See [Changing the System Password at `http://e-docs.bea.com/wls/docs61/adminguide/cnfgsec.html#cnfgsec003`](http://e-docs.bea.com/wls/docs61/adminguide/cnfgsec.html#cnfgsec003).

In all participating WebLogic Server 7.x and 8.x domains:

- Set a security credential for all domains to the same value on the Domain—Security—Advanced tab. The credential must match the `system` password in all participating WebLogic Server 6.x domains. For instructions, see [“Inter-Domain Transactions for WebLogic Server 8.x and 7.x Domains” on page 237-3](#).

Monitoring Transactions

In the Administration Console, you can monitor transactions for each server in the domain. Transaction statistics are displayed for a specific server, not the entire domain, even though transaction settings apply to the entire domain.

Viewing Transaction Statistics for a Server

To view transaction statistics for a server, follow these steps:

1. In the Administration Console, click the server node in the left pane and select a server.
2. In the right pane, select the Monitoring tab and then the JTA tab. Totals for transaction statistics are displayed. For information about the specific information displayed, see “Attributes” on page 469-2.
3. Optionally, click the text links at the bottom of the page to view transaction details by server resource or by transaction name, or for all active transactions on the server.

Viewing Transaction Statistics for Named Transactions

To view aggregate statistics about named transactions coordinated by a server, follow these steps:

1. In the Administration Console, click the server node in the left pane and select a server.
2. In the right pane, select the Monitoring tab and then the JTA tab.
3. Click the Monitor All Transactions by Name text link at the bottom of the page. See “Attributes” on page 493-1 for details about the information displayed.

Viewing Transaction Statistics for Server Resources

To view aggregate statistics for each transactional resource accessed on a server, follow these steps:

1. In the Administration Console, click the server node in the left pane and select a server.
2. In the right pane, select the Monitoring tab and then the JTA tab.
3. Click the Monitor All Transactions by Resource text link at the bottom of the page. See “Attributes” on page 494-1 for details about the information displayed.

Viewing Current (Inflight) Transactions for a Server

To view aggregate statistics for each transactional resource accessed on a server, follow these steps:

1. In the Administration Console, click the Servers node in the left pane and select a server.
2. In the right pane, select the Monitoring tab and then the JTA tab.
3. Click the Monitor All Inflight Transactions text link at the bottom of the page. See “Attributes” on page 450-1 for details about the information displayed.

Manually Resolving Current (Inflight) Transactions

In some cases, a transaction may not complete normally due to system or network failures. In such situations there may be locks held on behalf of the pending transaction that are inhibiting the progress of other transactions. After the Abandon Timeout period has elapsed, the WebLogic Server Transaction Manager removes the transaction from its internal data structures and writes a heuristic error to the server log. You can also manually resolve "stuck" transactions.

To manually resolve a transaction, you view current (inflight) transactions for a server from the Server—Monitoring—JTA tab (see “Viewing Current (Inflight) Transactions for a Server” on page 237-6) and then view details about a specific transaction by

clicking the transaction id. You can then force a commit or a rollback, depending on the status of the transaction. Table 237-1 lists the transaction states and manual resolution options for each transaction state.

Table 237-1 Transaction Status Definitions and Manual Resolution Options

Status	Definition	Forced Commit?	Forced Rollback?
Active	The application is processing the transaction. The transaction has not yet reached the two-phase commit processing.		Y
Preparing	Corresponds to the interval between when the transaction manager starts the <code>javax.transaction.Synchronization beforeCompletion()</code> callback processing, through the first phase of the 2PC protocol, and up to the point when all participants have responded, "ready to commit."		Y
Prepared	The interval between when all participants have responded to prepare up to the commit point (commit log record is flushed to disk) or to the initiation of rollback processing.	Y	Y
Committing	The time from when the commit decision is made up to the point when all participants have been informed of the outcome and the <code>javax.transaction.Synchronization afterCompletion()</code> callback processing has completed.	Y	
Committed	The transaction has been committed. It is likely that heuristics exists, otherwise the transaction would have been completed and would not have been displayed in the list of current transactions.	Y	
Rolling Back	This state occurs from the point when rollback processing is initiated up to the point when all participants have been instructed to rollback and the <code>javax.transaction.Synchronization afterCompletion()</code> callback processing has completed.		Y
Rolled Back	The transaction has been rolled back. It is likely that heuristics exists, otherwise the transaction would have been destroyed and would not have been displayed in the list of current transactions.		Y

Table 237-1 Transaction Status Definitions and Manual Resolution Options

Status	Definition	Forced Commit?	Forced Rollback?
Marked Roll Back	The transaction has been marked for rollback, perhaps as a result of a setRollbackOnly operation.		Y
No Transaction			
Unknown	Current status cannot be determined.	Y	Y

Note that it is possible for a transaction to have different states at different servers. For instance, a transaction may have been committed at the coordinating server, but a remote participant may not have received the commit instruction.

Manual Commit and Rollback Options

To manually resolve a transaction, you can choose from the following options. Options are restricted as described in Table 237-1.

- **Force Local Commit**—Each participating resource that is registered on the server is issued a commit operation for the specified transaction and the transaction will be removed from the local transaction manager's data structures. If the local server is the coordinator for the transaction, the commit record is released.
- **Force Global Commit**—A local commit operation is attempted at each participating server for the specified transaction. If this option is invoked on a non-coordinating server, the coordinator will be contacted to process the operation. The coordinating server will issue asynchronous requests to each participant server.
- **Force Local Rollback**—Each participating resource that is registered on the local server is issued a rollback operation for the specified transaction. The transaction will then be removed from the local transaction manager's data structures.
- **Force Global Rollback**—A local rollback operation is attempted at each participating server for the specified transaction. If this option is invoked on a non-coordinating server, the coordinator will be contacted to process the

operation. The coordinating server will issue asynchronous requests to each participant server.

When you select any of these options, WebLogic Server writes entries to the server log.

The difference between the Local and Global options is that Local options act only upon the current server resources (resources on the server that you select in the navigation tree in the left pane of the Administration Console), whereas the Global options attempt to perform the operation across all participating servers. If a Global operation is invoked for a transaction that is not coordinated by the local server then an attempt will be made to contact the coordinator of the transaction in order to perform the operation. If the coordinator cannot be reached, the operation will fail with a `javax.transaction.SystemException`.

In the case where a transaction may have been committed at the coordinating server (*committing* status), but a remote participant did not receive the commit instruction (*prepared* status). You can force a local commit on the remote participant to complete the transaction. In this case it is possible to force a rollback on the remote participant since its transaction state will still be prepared, but the transaction will complete heuristically. If you try to force a global rollback, the operation will fail because the state at the coordinator is committing. You cannot roll back a transaction with the committing status.

To Manually Resolve a Transaction

1. In the Administration Console, click the Servers node in the left pane and select a server.
2. In the right pane, select the Monitoring tab and then the JTA tab.
3. Click the Monitor All Inflight Transactions text link at the bottom of the page.
4. Click a Transaction Id to view details about the transaction. For information about the transaction statistics shown on the page, see “Attributes” on page 236-1.
5. Choose one of the following options: (options are restricted by transaction status; see Table 237-1)
 - Click Force Local Rollback to roll back the transaction on resources on the current server.

- Click Force Global Rollback to roll back the transaction on all resources that participate in the transaction.
- Click Force Local Commit to commit the transaction on resources on the current server.
- Click Force Global Commit to commit the transaction on all resources that participate in the transaction.

See “Manually Resolving Current (Inflight) Transactions” on page 237-6 for more information about manually resolving transactions.

Transaction Log Files

Each server has a transaction log which stores information about committed transactions coordinated by the server that may not have been completed. WebLogic Server uses the transaction log when recovering from system crashes or network failures. You cannot directly view the transaction log—the file is in a binary format.

The transaction log consists of multiple files. Each file is subject to garbage collection by the transaction manager. That is, when none of the records in a transaction log file are needed, the system deletes the file and returns the disk space to the file system. In addition, the system creates a new transaction log file if the previous log file becomes too large or a checkpoint occurs.

Caution: Do not manually delete transaction log files. Deleting transaction log files may cause inconsistencies in your data.

Transaction log files are uniquely named using a pathname prefix, the server name, a four-digit numeric suffix, and a file extension. The pathname prefix determines the storage location for the file. You can specify a value for the `TransactionLogFilePrefix` server attribute using the WebLogic Administration Console. The default `TransactionLogFilePrefix` is the server’s working directory.

You should set the `TransactionLogFilePrefix` so that transaction log files are created on a highly available file system, for example, on a RAID device. To take advantage of the migration capability of the Transaction Recovery Service for servers in a cluster, you must store the transaction log in a location that is available to a server

and its backup servers, preferably on a dual-ported SCSI disk or on a Storage Area Network (SAN). See [“Preparing to Migrate the Transaction Recovery Service” on page 237-21](#) for more information.

On a UNIX system with a server name of `websvr` and with the `TransactionLogFilePrefix` set to `/usr7/applog1/`, you might see the following log files:

```
/usr7/applog1/websvr0000.tlog  
/usr7/applog1/websvr0001.tlog  
/usr7/applog1/websvr0002.tlog
```

Similarly, on a Windows system with the `TransactionLogFilePrefix` set to `C:\weblogic\logA\`, you might see the following log files:

```
C:\weblogic\logA\websvr0000.tlog  
C:\weblogic\logA\websvr0001.tlog  
C:\weblogic\logA\websvr0002.tlog
```

If you notice a large number of transaction log files on your system, this may be an indication of multiple long-running transactions that have not completed. This can be caused by resource manager failures or transactions with especially large timeout values.

If the file system containing the transaction log runs out of space or is inaccessible, `commit()` throws `SystemException`, and the transaction manager places a message in the system error log. No transactions are committed until more space is available.

When migrating a server to another machine, move the transaction log files as well, keeping all the log files for a server together. See [“Moving a Server to Another Machine” on page 237-16](#) for more information.

Setting the Transaction Log File Location (Prefix)

To set the prefix for the transaction log files, which determines the location of the transaction log files, follow these steps:

1. In the Administration Console, click the server node in the left pane and select a server.
2. In the right pane, select the Logging tab and then the JTA tab.

3. Enter a transaction log file prefix (storage location for transaction logs) then click Apply to save the attribute setting. The new transaction log file prefix takes effect after you restart the server.

The default transaction log file prefix is the server's working directory. You can specify a relative path from the server's working directory or an absolute path to another storage location.

Setting the Transaction Log File Write Policy

You can select a transaction log file write policy to change the way WebLogic Server writes transaction log file entries. You can select either of the following options:

- **Cache-Flush**—(the default) Flushes operating system and on-disk caches after each entry to the transaction log. Transactions cannot commit until the commit record is written to stable storage.
- **Direct-Write**—Forces the operating system to write transaction log entries directly to disk with each write. This option is available on Windows, Solaris and HP-UX platforms.

Warning: On Windows, the Direct-Write transaction log file write policy may leave transaction data in the on-disk cache without immediately writing it to disk. This is not transactionally safe because a power failure can cause loss of on-disk cache data. To prevent cache data loss when using the Direct-Write transaction log file write policy on Windows, disable all write caching for the disk (enabled by default) or use a battery backup for the system. See [“Disabling the On-Disk Cache For a Disk Drive on Windows 2000” on page 235-3](#) for instructions.

The transaction log file write policy can affect transaction performance. You should test these options with your system to see which performs better. Direct-Write typically performs as well or better than Cache-Flush, depending on operating system and OS parameter settings, and is available on Windows, HP-UX, and Solaris. Windows systems optimize serial writes to disk such that subsequent writes to a file get faster after the first write to the file. Transaction log file entries are written serially, so this could improve performance. On some UNIX systems, the Cache-Flush option will flush all cached disk writes, not only those for the transaction log file, which could degrade transaction performance.

To set the transaction log file write policy, follow these steps:

1. In the Administration Console, click the server node in the left pane and select a server.
2. In the right pane, select the Logging tab and then the JTA tab.
3. Select a Transaction Log File Write Policy: Cache-Flush (the default) or Direct Write.
4. Click Apply to save the attribute setting. The new transaction log file write policy takes effect after you restart the server.

Heuristic Log Files

When importing transactions from a foreign transaction manager into WebLogic Server, the WebLogic Server transaction manager acts as an XA resource coordinated by the foreign transaction manager. In rare catastrophic situations, such as after the transaction abandon timeout expires or if the XA resources participating in the WebLogic Server imported transaction throw heuristic exceptions, the WebLogic Server transaction manager will make a heuristic decision. That is, the WebLogic Server transaction manager will decide to commit or roll back the transaction without input from the foreign transaction manager. If the WebLogic Server transaction manager makes a heuristic decision, it stores the information of the heuristic decision in the heuristic log files until the foreign transaction manager tells it to forget the transaction.

Heuristic log files are stored with transaction log files and look similar to transaction log files with .heur before the .tlog extension. They use the following format:

```
<TLOG_file_prefix>\<server_name><4-digit number>.heur.tlog
```

On a UNIX system with a server name of websvr, you might see the following heuristic log files:

```
/usr7/applog1/websvr0000.heur.tlog  
/usr7/applog1/websvr0001.heur.tlog  
/usr7/applog1/websvr0002.heur.tlog
```

Similarly, on a Windows system, you might see the following heuristic log files:

```
C:\weblogic\logA\websvr0000.heur.tlog  
C:\weblogic\logA\websvr0001.heur.tlog  
C:\weblogic\logA\websvr0002.heur.tlog
```

Handling Heuristic Completions

A **heuristic completion** (or heuristic decision) occurs when a resource makes a unilateral decision during the completion stage of a distributed transaction to commit or rollback updates. This can leave distributed data in an indeterminate state. Network failures or resource timeouts are possible causes for heuristic completion. In the event of an heuristic completion, one of the following heuristic outcome exceptions may be thrown:

- **HeuristicRollback**—one resource participating in a transaction decided to autonomously rollback its work, even though it agreed to prepare itself and wait for a commit decision. If the Transaction Manager decided to commit the transaction, the resource's heuristic rollback decision was incorrect, and might lead to an inconsistent outcome since other branches of the transaction were committed.
- **HeuristicCommit**—one resource participating in a transaction decided to autonomously commit its work, even though it agreed to prepare itself and wait for a commit decision. If the Transaction Manager decided to rollback the transaction, the resource's heuristic commit decision was incorrect, and might lead to an inconsistent outcome since other branches of the transaction were rolled back.
- **HeuristicMixed**—the Transaction Manager is aware that a transaction resulted in a mixed outcome, where some participating resources committed and some rolled back. The underlying cause was most likely heuristic rollback or heuristic commit decisions made by one or more of the participating resources.
- **HeuristicHazard**—the Transaction Manager is aware that a transaction might have resulted in a mixed outcome, where some participating resources committed and some rolled back. But system or resource failures make it impossible to know for sure whether a Heuristic Mixed outcome definitely occurred. The underlying cause was most likely heuristic rollback or heuristic commit decisions made by one or more of the participating resources.

When an heuristic completion occurs, a message is written to the server log. Refer to your database vendor documentation for instructions on resolving heuristic completions.

Some resource managers save context information for heuristic completions. This information can be helpful in resolving resource manager data inconsistencies. If the `ForgetHeuristics` attribute is selected (set to true) on the JTA panel of the WebLogic Console, this information is removed after an heuristic completion. When using a resource manager that saves context information, you may want to set the `ForgetHeuristics` attribute to false.

Abandoning Transactions

You can choose to abandon incomplete transactions after a specified amount of time. In the two-phase commit process for distributed transactions, the transaction manager coordinates all resource managers involved in a transaction. After all resource managers vote to commit or rollback, the transaction manager notifies the resource managers to act—to either commit or rollback changes. During this second phase of the two-phase commit process, the transaction manager will continue to try to complete the transaction until all resource managers indicate that the transaction is completed. Using the `AbandonTimeoutSeconds` attribute, you can set the maximum time, in seconds, that a transaction manager will persist in attempting to complete a transaction during the second phase of the commit protocol. The default value is 86400 seconds, or 24 hours. After the abandon transaction timer expires, no further attempt is made to resolve the transaction with any resources that are unavailable or unable to acknowledge the transaction outcome. If the transaction is in a prepared state before being abandoned, the transaction manager will roll back the transaction to release any locks held on behalf of the abandoned transaction and will write an heuristic error to the server log.

Related Information

- For instructions on how to set the `AbandonTimeoutSeconds` attribute, see “Configuring Transactions” on page 237-1.
- For information about manually resolving a transaction, see “Manually Resolving Current (Inflight) Transactions” on page 237-6.
- For more information about the two-phase commit process, see [Distributed Transactions and the Two-Phase Commit Protocol](#) in *Programming WebLogic JTA*.

Moving a Server to Another Machine

When an application server is moved to another machine, it must be able to locate the transaction log files on the new disk. For this reason, BEA recommends moving the transaction log files to the new machine before starting the server on the new machine. By doing so, you can ensure that recovery runs properly. When you start WebLogic Server on the new system, the server reads the transaction log files to recover pending transactions, if any. If the pathname is different on the new machine, update the `TransactionLogFilePrefix` attribute with the new path before starting the server. For instructions on how to change the `TransactionLogFilePrefix`, see “Setting the Transaction Log File Location (Prefix)” on page 237-11.

Transaction Recovery After a Server Fails

The WebLogic Server transaction manager is designed to recover from system crashes with minimal user intervention. The transaction manager makes every effort to resolve transaction branches that are prepared by resource managers with a commit or roll back, even after multiple crashes or crashes during recovery.

To facilitate recovery after a crash, WebLogic Server provides the Transaction Recovery Service, which automatically attempts to recover transactions on system startup. The Transaction Recovery Service owns the transaction log for a server. On startup, the Transaction Recovery Service parses all log files for incomplete transactions and completes them as described in [“Transaction Recovery Service Actions After a Crash”](#) on page 237-17.

Because the Transaction Recovery Service is designed to gracefully handle transaction recovery after a crash, BEA recommends that you attempt to restart a crashed server and allow the Transaction Recovery Service to handle incomplete transactions.

If a server crashes and you do not expect to be able to restart it within a reasonable period of time, you may need to take action. Procedures for recovering transactions after a server failure differ based on your WebLogic Server environment. For a non-clustered server, you can manually move the server (with transaction log files) to another system (machine) to recover transactions. See [“Recovering Transactions for a](#)

[Failed Non-Clustered Server](#)” on page 237-18 for more information. For a server in a cluster, you can manually *migrate* the *Transaction Recovery Service* to another server in the same cluster. Migrating the Transaction Recovery Service involves selecting a server with access to the transaction logs to recover transactions, and then migrating the service using the Administration Console or the WebLogic command line interface.

Note: For non-clustered servers, you can only move the entire server to a new system. For clustered servers, you can temporarily migrate the Transaction Recovery Service.

For more information about migrating the Transaction Recovery Service, see [“Recovering Transactions for a Failed Clustered Server”](#) on page 237-19. For more information about clusters, see *Using WebLogic Server Clusters* at [{DOCROOT}/cluster/index.html](#).

Transaction Recovery Service Actions After a Crash

When you restart a server after a crash or when you migrate the Transaction Recovery Service to another (backup) server, the Transaction Recovery Service does the following:

- Complete transactions ready for second phase of two-phase commit
For transactions for which a commit decision has been made but the second phase of the two-phase commit process has not completed (transactions recorded in the transaction log), the Transaction Recovery Service completes the commit process.
- Resolve prepared transactions
For transactions that the transaction manager has prepared with a resource manager (transactions in phase one of the two-phase commit process), the Transaction Recovery Service must call `XAResource.recover()` during crash recovery for each resource manager and eventually resolve (by calling the `commit()`, `rollback()`, or `forget()` method) all transaction IDs returned by `recover()`.
- Report heuristic completions

If a resource manager reports a heuristic exception, the Transaction Recovery Service records the heuristic exception in the server log and calls `forget()` if the `Forget Heuristics` configuration attribute is enabled. If the `Forget Heuristics` configuration attribute is not enabled, refer to your database vendor's documentation for information about resolving heuristic completions. See [“Handling Heuristic Completions” on page 237-14](#) for more information.

The Transaction Recovery Service provides the following benefits:

- Maintains consistency across resources

The Transaction Recovery Service handles transaction recovery in a consistent, predictable manner: For a transaction for which a commit decision has been made but is not yet committed before a crash, and `XAResource.recover()` returns the transaction ID, the Transaction Recovery Service consistently calls `XAResource.commit()`; for a transaction for which a commit decision has not been made before a crash, and `XAResource.recover()` returns its transaction ID, the Transaction Recovery Service consistently calls `XAResource.rollback()`. With consistent, predictable transaction recovery, a transaction manager crash by itself cannot cause a mixed heuristic completion where some branches are committed and some are rolled back.

- Persists in achieving transaction resolution

If a resource manager crashes, the Transaction Recovery Service must eventually call `commit()` or `rollback()` for each prepared transaction until it gets a successful return from `commit()` or `rollback()`. The attempts to resolve the transaction can be limited by setting the `AbandonTimeoutSeconds` configuration attribute. See [“Abandoning Transactions” on page 237-15](#) for more information.

Recovering Transactions for a Failed Non-Clustered Server

To recover transactions for a failed server, follow these steps:

1. Move (or make available) all transaction log files from the failed server to a new server.

2. Set the `TransactionLogFilePrefix` attribute with the path to the transaction log files. For instructions, see “Setting the Transaction Log File Location (Prefix)” on page 237-11.
3. Start the new server. The Transaction Recovery Service searches all transaction log files for incomplete transactions and completes them as described in [“Transaction Recovery Service Actions After a Crash” on page 237-17](#).

When moving transaction logs after a server failure, make all transaction log files available on the new machine before starting the server there. You can accomplish this by storing transaction log files on a dual-ported disk available to both machines. As in the case of a planned migration, update the `TransactionLogFilePrefix` attribute with the new path before starting the server if the pathname is different on the new machine. Ensure that all transaction log files are available on the new machine before the server is started there. Otherwise, transactions in the process of being committed at the time of a crash might not be resolved correctly, resulting in application data inconsistencies.

Note: The Transaction Recovery Service is designed to gracefully handle transaction recovery after a crash. BEA recommends that you attempt to restart a crashed server and allow the Transaction Recovery Service to handle incomplete transactions, rather than move the server to a new machine.

Recovering Transactions for a Failed Clustered Server

When a clustered server crashes, you can manually migrate the Transaction Recovery Service from the crashed server to another server in the same cluster using the Administration Console or the command line interface. The following events occur:

1. The Transaction Recovery Service on the backup server takes ownership of the transaction log from the crashed server.
2. The Transaction Recovery Service searches all transaction log files from the failed server for incomplete transactions and completes them as described in [“Transaction Recovery Service Actions After a Crash” on page 237-17](#).
3. If the Transaction Recovery Service on the backup server successfully completes all incomplete transactions from the failed server, the server releases ownership of the Transaction Recovery Service (including transaction log files) for the failed server so the failed server can reclaim it upon restart.

For instructions to migrate the Transaction Recovery Service using the Administration Console, see “Migrating the Transaction Recovery Service to a Server in the Same Cluster” on page 237-21.

A server can perform transaction recovery for more than one failed server. While recovering transactions for other servers, the backup server continues to process and recover its own transactions. If the backup server fails during recovery, you can migrate the Transaction Recovery Service to yet another server, which will continue the transaction recovery. You can also manually migrate the Transaction Recovery Service back to the original failed server using the Administration Console or the command line interface. See “Manually Migrating the Transaction Recovery Service Back to the Original Server” on page 237-23 for more information.

When a backup server completes transaction recovery for a server, it releases ownership of the Transaction Recovery Service (and transaction logs) for the failed server. When you restart a failed server, it attempts to reclaim ownership of its Transaction Recovery Service. If a backup server is in the process of recovering transactions when you restart the failed server, the backup server stops recovering transactions, performs some internal cleanup, and releases ownership of the Transaction Recovery service so the failed server can reclaim it and start properly. The failed server will then complete its own transaction recovery.

If a backup server still owns the Transaction Recovery Service for a failed server and the backup server is inactive when you attempt to restart the failed server, the failed server will not start because the backup server cannot release ownership of the Transaction Recovery Service. This is also true if the fail back mechanism fails or if the backup server cannot communicate with the Administration Server. You can manually migrate the Transaction Recovery using the Administration Console or the command line interface.

Limitations of Migrating the Transaction Recovery Service

When migrating the Transaction Recovery Service, the following limitations apply:

- You cannot migrate the Transaction Recovery Service to a backup server from a server that is running. You must stop the server before migrating the Transactions Recovery Service.
- The backup server does not accept new transaction work for the failed server. It only processes incomplete transactions.
- The backup server does not process heuristic log files.

- The backup server only processes log records written by WebLogic Server. It does not process log records written by gateway implementations, including WebLogic Tuxedo Connector.

Preparing to Migrate the Transaction Recovery Service

To migrate the Transaction Recovery Service from a failed server in a cluster to another server (backup server) in the same cluster, the backup server must have access to the transaction log files from the failed server. Therefore, you must store transaction log files on persistent storage available to both (or more) servers. BEA recommends that you store transaction log files on a Storage Area Network (SAN) device or a dual-ported disk. Do not use an NFS file system to store transaction log files. Because of the caching scheme in NFS, transaction log files on disk may not always be current. Using transaction log files stored on an NFS device for recovery may cause data corruption.

When migrating the Transaction Recovery Service from a server, you must stop the failing or failed server before actually migrating the Transaction Recovery Service. If the original server is still running, you cannot migrate the Transaction Recovery Service from it.

For detailed instructions to migrate the Transaction Recovery Service, see [Migrating the Transaction Recovery Service to a Server in the Same Cluster](#) in the *Administration Console Online Help* at `{DOCRROOT}/ConsoleHelp/jta.html#jta_trs_migrate`.

Migrating the Transaction Recovery Service to a Server in the Same Cluster

To migrate the Transaction Recovery Service from a failed server in a cluster, follow these steps:

1. Make sure the failed or failing server is not running:
 - a. Click the Servers node in the left pane in the Administration Console to expand it.
 - b. Right-click the failed server and select Start/Stop this server.
 - c. In the right pane of the Administration Console, click Graceful shutdown of this server. If that action fails, retry these steps and select Force shutdown of this server.

2. In the left pane in the Administration Console, select the failed server from which you want to migrate the Transaction Recovery Service. A dialog displays in the right pane with tabs for configuring the server.
3. Select the Control tab, then select the JTA Migration tab. The JTA Migration tab includes the following:
 - Cluster—The name of the cluster to which the server belongs.
 - Current Server—The server that currently owns the Transaction Recovery Service for the selected server. (The selected server name is displayed at the top of the page.)
 - Destination Server—A list of servers in the cluster. This list shows all servers in the cluster or a list of servers that you specify. See [“Constraining the Servers to Which the Transaction Recovery Service can Migrate” on page 237-22](#).
4. In Destination Server, select the server that you want to recover transactions for the failed server.
5. Click Migrate and follow any additional instructions in the right pane.

Note: The Transaction Recovery Service is designed to gracefully handle transaction recovery after a crash. BEA recommends that you attempt to restart a crashed server and allow the Transaction Recovery Service to handle incomplete transactions, rather than migrate the Transaction Recovery Service to another server.

Constraining the Servers to Which the Transaction Recovery Service can Migrate

You may want to limit the choices of the servers to use as a Transaction Recovery Service backup for a server in a cluster. For example, all servers in your cluster may not have access to the transaction log files for a server. You can limit the list of destination servers available on the ~~Server—Control—Migrate~~ JTA tab in the Administration Console by following these instructions:

1. In the Administration Console, click the Servers node in the left pane to expand it.
2. Select the server for which you want to specify Transaction Recovery Service backup servers. A dialog displays in the right pane with tabs for configuring the server.

3. Select the Control tab, then select the JTA Migration Config tab.
4. In the Available list of Constrained Candidate Servers, select the servers you want to use as a Transaction Recovery Service backup and click the right arrow to move the servers to the Chosen list.

Note: You must include the original server in the list of chosen servers so that you can manually migrate the Transaction Recovery Service back to the original server, if need be. The Administration Console enforces this rule.

5. Click Apply to save your changes.

Viewing Current Owner of the Transaction Recovery Service

When you migrate the Transaction Recovery Service to another server in the cluster, the backup server takes ownership of the Transaction Recovery Service until it completes all incomplete transactions. After which, it releases ownership of the Transaction Recovery Service and the original server can reclaim it. You can see the current owner on the ~~Server—Control—Migrate JTA~~ tab in the Administration Console. Follow these instructions:

1. In the Administration Console, click the Servers node in the left pane to expand it.
2. Select the server for which you want to see the owner of the Transaction Recovery Service. A dialog displays in the right pane with tabs for configuring the server.
3. Select the Control tab. If necessary, select the JTA Migration tab. On the JTA Migration tab, the Current Server indicates the current owner of the Transaction Recovery Service.

Manually Migrating the Transaction Recovery Service Back to the Original Server

After completing transaction recovery for a failed server, a backup server releases ownership of the Transaction Recovery Service so that the original server can reclaim it when the server is restarted. If the backup server stops (crashes) for any reason before it completes transaction recovery, the original server cannot reclaim ownership of the Transaction Recovery Service and will not start. You can manually migrate the

Transaction Recovery Service back to the original server by selecting the original server as the Destination Server. The backup server must not be running when you migrate the service back to the original server. Follow the instructions below.

Note: Please note the following:

- A backup server will continue to recover incomplete transactions after you restart it. You will not need to manually migrate the Transaction Recovery Service back to the original server if the backup server completes the transaction recovery.
 - If you restart the original server while the backup server is recovering transactions, the backup server will gracefully release ownership of the Transaction Recovery Service. You do not need to stop the backup server. See “Recovering Transactions for a Failed Clustered Server” on page 237-19.
1. Make sure the backup server is not running. To do this, click the Servers node in the left pane in the Administration Console to expand it, then right-click the backup server and select Stop this server. Follow any additional instructions.
 2. In the Administration Console, click the Servers node in the left pane to expand it.
 3. Select the failed server that you want to migrate the Transaction Recovery Service back to. A dialog displays in the right pane with tabs for configuring the server.
 4. Select the Control tab. If necessary, select the JTA Migration tab.
 5. In Destination Server, select the original server.
 6. Click Migrate and follow any additional instructions in the right pane.

Overview of Transaction Management

You use the Administration Console to access tools for configuring the WebLogic Server features, including the Java Transaction API (JTA). The transaction configuration process involves specifying values for attributes. These attributes define various aspects of the transaction environment:

- Transaction timeouts and limits
- Transaction Manager behavior
- Transaction log file prefix

Settings you make in the Administration Console, including configuration settings for JTA, are persisted in the `config.xml` file for the domain. For information about entries in this file, see the following sections of the *Configuration Reference Guide*:

- [JTA at {DOCROOT}/config_xml/JTA.html](#)
- [JTAMigratableTarget at {DOCROOT}/config_xml/JTAMigratableTarget.html](#)
- [JTARecoveryService at {DOCROOT}/config_xml/JTARecoveryService.html](#)
- [JDBCTxDataSource at {DOCROOT}/config_xml/JDBCTxDataSource.html](#)

Before configuring your transaction environment, you should be familiar with the J2EE components that can participate in transactions, such as EJBs, JDBC, and JMS.

- EJBs (Enterprise JavaBeans) use JTA for transaction support. Several deployment descriptors relate to transaction handling. For more information about programming with EJBs and JTA, see [Programming WebLogic Enterprise JavaBeans](#).
- JDBC (Java Database Connectivity) provides standard interfaces for accessing relational database systems from Java. JTA provides transaction support on connections retrieved using a JDBC driver and transaction data source. For more information about programming with JDBC and JTA, see [Programming WebLogic JDBC](#).
- JMS (Java Messaging Service) uses JTA to support transactions across multiple data resources. WebLogic JMS is an XA-compliant resource manager. For more information about programming with JMS and JTA, see [Programming WebLogic JMS](#).

Attributes and Console Screen Reference for JTA

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

“Domain --> Configuration --> JTA” on page 67-1

“Server --> Monitoring --> JTA” on page 469-1

“Transactions By Name” on page 493-1

“Transactions by Resource” on page 494-1

“Inflight JTA Transactions” on page 450-1

“Transaction Details” on page 236-1

“Server --> Logging --> JTA” on page 448-1

“Server --> Control --> JTA Migration Config.” on page 479-1

“Server --> Control --> JTA Migrate” on page 482-1



1 JNDI

[“Attributes and Console Screen Reference for JNDI” on page 240-1]

Overview of JNDI Management

You use the Administration Console to manage JNDI. The JNDI API enables applications to look up objects—such as Data Sources, EJBs, JMS, and MailSessions—by name. The JNDI tree is represented by the left pane in the Administration Console.

For additional information, see [Programming WebLogic JNDI](#).

What Do JNDI and Naming Services Do?

JNDI provides a common-denominator interface to many existing naming services, such as LDAP (Lightweight Directory Access Protocol) and DNS (Domain Name System). These naming services maintain a set of bindings, which relate names to objects and provide the ability to look up objects by name. JNDI allows the components in distributed applications to locate each other. The WebLogic Server implementation of JNDI supplies methods that:

- Give clients access to the WebLogic name services
- Make objects available in the WebLogic namespace
- Retrieve objects from the WebLogic namespace

For more information on how to use JNDI, see [Programming WebLogic JNDI](#).

Viewing Objects in the JNDI Tree

To view objects in the JNDI tree from the Administration Console, do the following:

1. Right click on the node for your WebLogic Server in the right pane. This opens a pop-up menu.
2. Select View JNDI Tree.
3. Click to select an object.

Note: Object Class, Object Hash Code, and Object To String entries are not available for a Local Stateless Bean deployment.

Loading Objects in the JNDI Tree

Using the Administration Console, you load WebLogic Server J2EE services and components, such as RMI, JMS, EJBs, and JDBC Data Sources, in the JNDI tree.

To load an object in the JNDI tree, do the following:

1. Choose a name under which you want the object to appear in the JNDI tree.
2. Enter that name in the JNDI Name attribute field when you create the object. When the object is loaded, JNDI provides a path to the object.
3. Verify that the object has been loaded by viewing the JNDI tree. For more information on how to view the JNDI tree, see “Viewing Objects in the JNDI Tree” on page 239-2.

For more information on configuring objects, see Table 239-1 Objects in JNDI Tree.

Table 239-1 Objects In JNDI Tree

Service	Bound Object w/Link to Online Help
JDBC DataSource	“JDBC Data Source --> Configuration” on page 129-1 and “JDBC Data Source --> Configuration” on page 136-1
JMS Connection Factory	“JMS Connection Factory --> Configuration --> General” on page 155-1
Mail	“Mail Session --> Configuration” on page 270-1
Deployment Descriptors	“Connector Component --> Configuration --> Descriptor” on page 45-1

Attributes and Console Screen Reference for JNDI

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

“Server --> Services --> Virtual Hosts” on page 462-1

“Virtual Host --> Configuration --> General” on page 534-1

“Virtual Host --> Configuration --> HTTP” on page 535-1

“Virtual Host --> Notes” on page 537-1



Jolt Connection Pool --> Configuration --> Addresses

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Jolt provides connection failure handling by using two lists of Jolt Service Listeners (JSL) addresses for each connection pool: a primary list and a failover list. Jolt provides connection failure handling in the following cases:

- When WebLogic Server is booted: If no JSL defined in the primary address list is accessible at server startup, Jolt uses JSL addresses from the failover list.
- When a Jolt pool loses an active connection: WebLogic Server tries to reconnect by using other addresses from the primary address list. If all addresses in the primary list fail, WebLogic Server tries to reconnect using addresses from the failover list. Lost connections are restarted only when they are needed.

Tasks

“Configuring Connection Failure Handling” on page 249-4

Related Topics

BEA Jolt at <http://e-docs.bea.com>

Attributes

Table 241-1

Attribute Label	Description	Value Constraints
Primary Addresses	Defines a list of primary addresses used to establish a connection between the Jolt connection pool and Tuxedo.	
Failover Addresses	Defines a list of addresses used if connections defined by the primary addresses cannot be established or fail.	

Jolt Connection Pool --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Jolt uses connection pools to enable WebLogic Server clients to connect to BEA Tuxedo domains. WebLogic Server creates the Jolt connection pools at startup and assigns connections to WebLogic Server clients as needed.

Tasks

“Create a Jolt Connection Pool” on page 249-3

Related Topics

[BEA Jolt at http://e-docs.bea.com](http://e-docs.bea.com)

Attributes

Table 242-1

Attribute Label	Description	Value Constraints
Name	The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.	

Table 242-1

Attribute Label	Description	Value Constraints
Minimum Pool Size	The minimum number of connections to be added to the Jolt connection pool when WebLogic Server starts.	<i>Default:</i> 0
Maximum Pool Size	The maximum number of connections that can be made from the Jolt connection pool.	<i>Default:</i> 1
Recv Timeout	The amount of time the client waits to receive a response before timing out.	
Security Context Enabled	Defines state the security context for this connection pool. If selected (set to true), security context is enabled.	<i>Default:</i> false <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false

Jolt Connection Pool --> Configuration --> User

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Jolt establishes a security context in WebLogic Server used to create a security identity in a BEA Tuxedo domain. Each network connection in the Jolt Connection Pool is authenticated through a User identity. Use this tab to provide security propagation information for inter-domain authentication.

Tasks

“Configuring a User Security Identity” on page 249-5

Related Topics

[BEA Jolt at http://e-docs.bea.com](http://e-docs.bea.com)

Attributes

Table 243-1

Attribute Label	Description	Value Constraints
User Name	The user name for this connection pool.	

Table 243-1

Attribute Label	Description	Value Constraints
User Role	The user role for this connection pool.	
User Password	The user password for this connection pool.	<i>Encrypted: yes</i>
Application Password	The application password for this connection pool.	<i>Encrypted: yes</i>

Jolt Connection Pool --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this section to supply optional information about your configuration.

Tasks

Enter free form text notes to describe your Jolt pool or configuration.

Related Topics

[BEA Jolt at http://e-docs.bea.com](http://e-docs.bea.com)

Attributes

Table 244-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration.	<i>Dynamic:</i> yes



Jolt Connection Pool --> Target and Deploy

[Tasks](#) [Related Topics](#)

Overview

Use this tab to assign connectivity information contained in a Jolt connection pool a selected server or cluster.

Tasks

“Assign Jolt Connection Pools to a Server” on page 249-3

“Assign Jolt Connection Pools to a Cluster” on page 249-4

Related Topics

BEA Jolt at <http://e-docs.bea.com>



Jolt Connection Pool

Jolt connection pools allow you to connect WebLogic Server clients to BEA Tuxedo domains. The server creates the Jolt connection pools at startup and assigns connections to WebLogic Server clients as needed.

- For more information on how to configure a Jolt Startup and Shutdown class, see [Create a Jolt Startup & Shutdown Class](#).
- For more information on how to configure a connection, see [Create a Jolt Connection Pool](#).
- For more information on how to select a server, see [Assign Jolt Connection Pools to a Server](#).
- For more information on how to select a cluster, see [Assign Jolt Connection Pools to a Cluster](#).
- For more information on how to configure failover, see [Configuring Connection Failure Handling](#).
- For more information on how to configure security, see [Configuring a User Security Identity](#).
- For more information on how to monitor Jolt connection pools, see [Monitor Active Instances of a Jolt Connection Pool](#).



Jolt Connection Pool --> Monitoring

[Tasks](#) [Related Topics](#)

Overview

From the Jolt Connection Pool—Monitoring tab, you can click the Monitor all Active Pools text link to access a table that lists information about database connections in the connection pool.

Tasks

[“Create a Jolt Connection Pool” on page 249-3](#)

[“Monitor Active Instances of a Jolt Connection Pool” on page 249-6](#)

Related Topics

[BEA Jolt at http://e-docs.bea.com](http://e-docs.bea.com)



Active Jolt Connection Pool

The Active Jolt Connections table shows information for active Jolt connection pools assigned to the server. You can click column headings in the table to sort the information in the table. You can also click [Customize this view](#) to select the columns to display in the table.

- To create a Jolt connection pool, see [“Create a Jolt Connection Pool” on page 249-3](#).
- To assign a Jolt connection pool to a server, see [“Assign Jolt Connection Pools to a Server” on page 249-3](#).
- To assign a Jolt connection pool to a cluster, see [“Assign Jolt Connection Pools to a Cluster” on page 249-4](#).
- To reset an active connection pool, see [“Resetting a Jolt Connection Pool” on page 249-6](#).



1 Jolt

[“Attributes and Console Screen Reference for Jolt” on page 250-1]

Jolt is a Java-based client API that manages requests to BEA Tuxedo services via a Jolt Service Listener (JSL) running on the Tuxedo server. The Jolt API is embedded within the WebLogic API, and is accessible from a servlet or any other BEA WebLogic application. The Jolt Java client class library can be used in HTTP servlets running in WebLogic Server and provides an interface between HTML browser clients and BEA Tuxedo services.

Configuring Jolt for WebLogic Server

The following sections provide information on how to configure WebLogic Server to interoperate with Tuxedo using Jolt. Consult your [BEA Product Documentation](#) for detailed information on how to configure and use Jolt.

Set Your Environment

You must update your CLASSPATH environment variable used by your WebLogic Server and command-line shells to include the following files from your Tuxedo installation:

- jolt.jar
- joltjse.jar
- joltwls.jar

Create a Jolt Startup & Shutdown Class

You must create a Startup and a Shutdown class to establish and terminate the connection between Tuxedo and WebLogic Server. WebLogic Server uses the `PoolManagerStartUp` class to establish a connection to Tuxedo whenever the WebLogic Server is started or restarted. The shutdown class instructs WebLogic Server to invoke the `PoolManagerShutDown` class to disconnect the Jolt session pools from Tuxedos when the WebLogic Server shuts down. For more information on Startup and Shutdown classes, see [Startup and Shutdown Classes](#).

1. In the left pane, expand the Deployments folder.
2. Click on the name of the Startup & Shutdown folder.
3. Click on the Configure a new Startup Class ... link.
4. In the Name field, enter the name for your Jolt Startup class. Example:
`MyJoltStartupClass`.
5. In the ClassName field, enter the following:
`bea.jolt.pool.servlet.weblogic.PoolManagerStartUp`
6. Click to enable Failure is Fatal.
7. Click Create.
8. Target and Deploy your Jolt Startup class and click Apply.
9. Click on the name of the Startup & Shutdown folder.
10. Click on the Configure a new Shutdown Class ... link.
11. In the Name field, enter the name for your Jolt Shutdown class. Example:
`MyJoltShutdownClass`.
12. In the ClassName field, enter the following:
`bea.jolt.pool.servlet.weblogic.PoolManagerShutDown`
13. Click Create.
14. Target and Deploy your Jolt Shutdown class and click Apply.

Configuring a Jolt Connection Pool

The following sections provide information on how to configure Jolt Connections Pools for you Tuxedo ATMI applications.

Create 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 tab to provide Jolt Service Listener (JSL) addresses for a connection pool. See [Configuring Connection Failure Handling](#).
6. Click the User tab to create a user identity. See [Configuring a User Security Identity](#).
7. Restart the Administration Server so that your changes can take effect.

Assign Jolt Connection Pools to a Server

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 Target and Deploy tab.

3. To assign a connection pool to a selected server, click one or more servers in the list of Independent Servers that has an empty check box. A check mark is displayed next to each server assigned to a Jolt connection pool.
4. To remove a connection pool from a server, click one or more servers in the list of Independent Servers that has a check mark. An empty check box is displayed next to each server that is not assigned to a Jolt connection pool.
5. Click Apply to save your assignments.

Assign Jolt Connection Pools to a Cluster

1. Click the instance node in the left pane under Jolt for the connection pool you want to assign. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Target and Deploy tab.
3. To assign a connection pool to a cluster, select one or more clusters in the list of clusters that has an empty check box. A check mark is displayed next to each cluster assigned to a Jolt connection pool.
 - a. To assign a connection pool to the all of the servers in a cluster, click All servers in the cluster.
 - b. To assign a connection pool to selected servers in a cluster, click Part of the cluster. Select one or more clusters of the available servers.
4. To remove a connection pool from a cluster, select one or more servers in the list of clusters that has a check mark. An empty check box is displayed next to each cluster that is not assigned to a Jolt connection pool.
5. Click Apply to save your assignments.

Configuring Connection Failure Handling

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

2. Click the Configuration tab.
3. Click the Addresses tab.
4. Assign a list of primary JSL addresses.
5. Assign a list of Failover JSL addresses.
6. Click Apply to save your assignments.

Configuring a User Security Identity

1. Click the instance node in the left pane under Jolt for the connection pool you want to assign. A dialog displays in the right pane showing the tabs associated with this instance.
2. Click the Configuration tab.
3. Click the User tab.
4. In User Name, assign a user identity.
5. In User Role, assign a user role.
6. In User Password, click change to assign a user password.
7. In Application Password, click change to assign a user password.
8. Click Apply to save your assignments.

Administering Active Jolt Connection Pools

The following sections provide information on the Active Jolt Connection Pools page. This page allows you to view statistics about the active Jolt connection pools in your WebLogic Server domain.

Monitor Active Instances of 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 Name of the Jolt connection pool you want to monitor.
3. Click the Monitoring tab.
4. Click the Monitor all Active Pools text link.

Resetting a Jolt Connection Pool

You can reset the Jolt connection pool without having to restart WebLogic Server. Clicking on the Reset icon brings down the existing connections to the Jolt servers and reconnects to them.

Attributes and Console Screen Reference for Jolt

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“Jolt Connection Pool --> Configuration --> Addresses” on page 241-1](#)

[“Jolt Connection Pool --> Configuration --> General” on page 242-1](#)

[“Jolt Connection Pool --> Configuration --> User” on page 243-1](#)

[“Jolt Connection Pool --> Target and Deploy” on page 245-1](#)

[“Jolt Connection Pool --> Monitoring” on page 247-1](#)

[“Jolt Connection Pool --> Notes” on page 244-1](#)

[“Active Jolt Connection Pool” on page 248-1](#)

[“Jolt Connection Pool” on page 246-1](#)



Customize Log View

[Tasks](#) [Related Topics](#) [Display Filters](#)

Overview

The Customize Log View page constructs a query that returns the log-message data that you request. The query searches in the current log file only. If the log file has been rotated, the query does not search through old messages that have been moved to another file.

None of the items on this page effect the messages that are actually stored in the log file; instead, they determine the log file data that the Administration Console displays.

The Administration Console does not save your filter criteria. The next time you access the Search Log page, the Administration Console uses the default criteria to display messages.

Tasks

“Viewing Server Logs” on page 253-9

“Viewing the Domain Log” on page 253-12

“Rotating Log Files” on page 253-16

Related Topics

“Overview of WebLogic Server Log Messages and Log Files” on page 253-2

Display Filters

Table 251-1

Display Filter	Description
Columns	Specifies which message attributes the log viewer displays. For more information, refer to “Message Attributes” on page 253-3.
Servers	<p>Applicable only for the domain log viewer.</p> <p>Causes the log viewer to display only the messages that were forwarded to the domain log from the specified server or servers.</p> <p>If you leave the Chosen column empty, the log viewer displays messages that were forwarded from all servers in the domain.</p>
Subsystems	<p>Causes the log viewer to display only the messages that were generated by one or more specific subsystems.</p> <p>If you leave the Chosen column empty, the log viewer displays messages that were generated by all subsystems.</p>
Severities	Displays only the messages that correspond to the specified severity levels. For more information, refer to “Message Severity” on page 253-5.
Users	<p>Displays only the messages that were generated within the security context that you specify.</p> <p>If no IDs are specified, messages from all user IDs can be displayed.</p> <p>Every message includes the user ID under which the associated event was executed. To execute some pieces of internal code, WebLogic Server authenticates the ID of the user who initiates the execution and then runs the code under a special Kernel Identity user ID.</p> <p>J2EE modules such as EJBs that are deployed onto a server instance report the user ID that the module passes to the server.</p>

Table 251-1

Display Filter	Description
Sub String	Searches the Message Text attribute and returns only the messages that contain a match to the string you specify. The search is case sensitive.
Since	Displays the messages that were written to the log file since the value that you specify. If you do not specify a value, the log viewer starts its display from the earliest messages that were written to the log file. The first messages are from the server's initial session. If you set up log file rotation, you see only the messages that are in the current log file. For more information, refer to "Rotating Log Files" on page 253-16.
Max Messages	Specifies the maximum number of messages that the log viewer displays.
Show New messages as they are logged	The Administration Console displays messages as they are generated. Only the messages that match other filter criteria are displayed.
Foward	Reverses the order in which messages are displayed. With this option selected, the messages at the top of the log viewer are the earliest messages in the current log file.



Search Log

[Tasks](#) [Related Topics](#)

Overview

The Search Log page displays messages from the domain log file or an individual server's log file, depending on whether you clicked View Domain Log or View Server Log.

By default, the Search Log page displays up to 500 messages in reverse chronological order. The messages at the top of the window the most recent messages that have been generated.

You can change the default set of messages this page displays by clicking the [Customize this view link](#).

Tasks

“Viewing Server Logs” on page 253-9

“Viewing the Domain Log” on page 253-12

Related Topics

“Overview of WebLogic Server Log Messages and Log Files” on page 253-2

“Rotating Log Files” on page 253-16



1 Server Log

[“Attributes and Console Screen Reference for Logging” on page 254-1]

By default, each WebLogic Server instance maintains a server log, an HTTP access log, and a Java Transaction API (JTA) transaction log. You can also configure a server instance to maintain a Java Database Connectivity (JDBC) log.

The **server log** records information about events such as the startup and shutdown of servers, the deployment of new applications, or the failure of one or more subsystems. The messages include information about the time and date of the event as well as the ID of the user who initiated the event.

You can view and sort these server log messages to detect problems, track down the source of a fault, and track system performance. You can also create client applications that listen for these messages and respond automatically. For example, you can create an application that listens for messages indicating a failed subsystem and sends email to a system administrator.

The following sections describe working with the server log:

- “Overview of WebLogic Server Log Messages and Log Files” on page 253-2
- “Viewing Server Logs” on page 253-9
- “Viewing the Domain Log” on page 253-12
- “Rotating Log Files” on page 253-16
- “Specifying Which Messages a Server Sends to Standard Out” on page 253-20
- “Viewing Standard Out for a Server Started by the Node Manager” on page 253-21
- “Configuration Auditing” on page 253-22
- “Other Logging Tasks” on page 253-27

For related information, refer to:

- ["Setting Up HTTP Access Logs"](#) for information about HTTP access logs.
- "Transaction Log Files" on page 237-10 for information about JTA transaction logs.
- The [Using WebLogic Logging Services](#) guide for information on setting up your application to listen for server log messages.

Overview of WebLogic Server Log Messages and Log Files

Each subsystem within WebLogic Server generates server log messages to communicate its status. For example, when you start a WebLogic Server instance, the Security subsystem writes a message to report its initialization status.

To keep a record of the messages that its subsystems generate, WebLogic Server writes the messages to log files. The server log file is located on the computer that hosts the server instance. By default, the server log file is located below the server instance's root directory: `root-directory\server-name\server-name.log`. For more information, refer to "Changing the Name and Location of the Server Log File" on page 253-30.

To view messages in a server log file, you can log on the WebLogic Server host computer and use a standard text editor, or you can log on to any computer and use the log file viewer in the Administration Console. For more information, refer to "Viewing Server Logs" on page 253-9.

Note: We recommend that you do not modify log files by manually editing them. Modifying a file changes the timestamp and can confuse log file rotation. In addition, editing a file might lock it and prevent updates from WebLogic Server.

In addition to writing messages to a log file, each server instance prints a subset of its messages to standard out. Usually, **standard out** is the shell (command prompt) in which you are running the server instance. However, some operating systems enable you to redirect standard out to some other location. If you use the Node Manager to

start a Managed Server, the Node Manager redirects a server's standard out and standard error to a file on the Node Manager's host computer. By default, a server instance prints only messages of a `WARNING` severity level or higher to standard out. (A subsequent section, "Message Severity," describes severity levels.) You can modify the severity threshold so that the server prints more or fewer messages to standard out.

The following sections provide an overview of WebLogic Server log messages and log files:

- "Message Attributes" on page 253-3
- "Format of Message Output" on page 253-6
- "Local Log Files and Domain Log Files" on page 253-7

Message Attributes

The messages for all WebLogic Server subsystems contain a consistent set of fields (attributes) as described in Table 253-1. In addition, if your application uses WebLogic logging services to generate messages, its messages will contain these attributes.

Table 253-1 Log Message Attributes

Attribute	Description
Timestamp	Time and date when the message originated, in a format that is specific to the locale. The Java Virtual Machine (JVM) that runs each WebLogic Server instance refers to the host computer's operating system for information about the local time zone and format.
Severity	Indicates the degree of impact or seriousness of the event reported by the message. For more information, refer to "Message Severity" on page 253-5.
Subsystem	Indicates the subsystem of WebLogic Server that was the source of the message. For example, Enterprise Java Bean (EJB) container or Java Messaging Service (JMS).

Table 253-1 Log Message Attributes

Attribute	Description
Server Name Machine Name Thread ID	<p>Identify the origins of the message:</p> <ul style="list-style-type: none"> ■ Server Name is the name of the WebLogic Server instance on which the message was generated. ■ Machine Name is the DNS name of the computer that hosts the server instance. ■ Thread ID is the ID that the JVM assigns to the thread in which the message originated. <p>Log messages that are generated within a client JVM client do not include these fields. For example, if your application runs in a client JVM and it uses the WebLogic logging services, the messages that it generates and sends to the WebLogic Server log files will not include these fields.</p>
User ID	<p>The user ID under which the associated event was executed.</p> <p>To execute some pieces of internal code, WebLogic Server authenticates the ID of the user who initiates the execution and then runs the code under a special Kernel Identity user ID.</p> <p>J2EE modules such as EJBs that are deployed onto a server instance report the user ID that the module passes to the server.</p> <p>Log messages that are generated within a client JVM client do not include this field.</p>
Transaction ID	<p>Present only for messages logged within the context of a transaction.</p>
Message ID	<p>A unique six-digit identifier.</p> <p>All message IDs that WebLogic Server system messages generate start with BEA- and fall within a numerical range of 0-499999.</p> <p>Your applications can use a Java class called <code>NonCatalogLogger</code> to generate log messages instead of using an internationalized message catalog. The message ID for <code>NonCatalogLogger</code> messages is always 000000. For more information, refer to "Writing Messages to the WebLogic Server Log."</p>
Message Text	<p>A description of the event or condition.</p>

Message Severity

The **severity** attribute of a WebLogic Server log message indicates the potential impact of the event or condition that the message reports.

Table 253-2 lists the severity levels of log messages from WebLogic Server subsystems, starting from the lowest level of impact to the highest.

Table 253-2 Message Severity

Severity	Meaning
INFO	Used for reporting normal operations.
WARNING	A suspicious operation or configuration has occurred but it might not affect normal operation.
ERROR	A user error has occurred. The system or application can 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. Only WebLogic Server and its subsystems generate messages of this severity.
CRITICAL	A system or service error has occurred. The system can recover but there might be a momentary loss or permanent degradation of service. Only WebLogic Server and its subsystems generate messages of this severity.
ALERT	A particular 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. Only WebLogic Server and its subsystems generate messages of this severity.
EMERGENCY	The server is in an unusable state. This severity indicates a severe system failure or panic. Only WebLogic Server and its subsystems generate messages of this severity.

WebLogic Server subsystems generate many messages of lower severity and fewer messages of higher severity. For example, under normal circumstances, they generate many INFO messages and no EMERGENCY messages.

If your application uses the WebLogic logging services, it can use an additional severity level, DEBUG. WebLogic Server subsystems do not use this severity level. For more information, refer to "[Writing Debug Messages](#)."

Format of Message Output

When a WebLogic Server instance writes a message to the log file, the first line of each message begins with ##### followed by the message attributes. Each attribute is contained between angle brackets.

The following is an example of a message in a log file:

```
#####<Nov 11, 2002 2:07:57 PM EST> <Notice> <WebLogicServer> <MyComputer>  
<MedRecServer> <main> <kernel identity> <> <BEA-000360>  
<Server started in RUNNING mode>
```

In this example, the message attributes are: Timestamp, Severity, Subsystem, Machine Name, Server Name, Thread ID, User ID, Transaction ID, Message ID, and Message Text.

If a message is not logged within the context of a transaction, the angle brackets for Transaction ID are present even though no Transaction ID is present.

If the message includes a stack trace, the stack trace follows the list of message attributes.

WebLogic Server uses the host computer's default character encoding for the messages it writes.

Format of Output to Standard Out and Standard Error

When a WebLogic server instance writes a message to standard out, the output does not include the ##### prefix and does not include the Server Name, Machine Name, Thread ID, and User ID fields.

The following is an example of how the message from the previous section would be printed to standard out:

```
<Nov 11, 2002 2:07:57 PM EST> <Notice> <WebLogicServer> <BEA-000360>  
<Server started in RUNNING mode>
```

In this example, the message attributes are: Timestamp, Severity, Subsystem, Message ID, and Message Text.

Local Log Files and Domain Log Files

Each WebLogic Server instance writes all messages from its subsystems and applications to a log file that is located on the local host computer.

In addition to writing messages to its local log file, each server instance forwards a subset of its messages to a domain-wide log file. By default, servers forward only messages of severity level `ERROR` or higher. While you can modify the set of messages that are forwarded, servers can never forward messages of the `DEBUG` severity level or any stack traces that are included in a message.

The domain log file provides a central location from which to view the overall status of the domain.

For more information, refer to “Specifying the Messages That a Server Forwards to the Domain Log” on page 81-1.

How a Server Instance Forwards Messages to the Domain Log

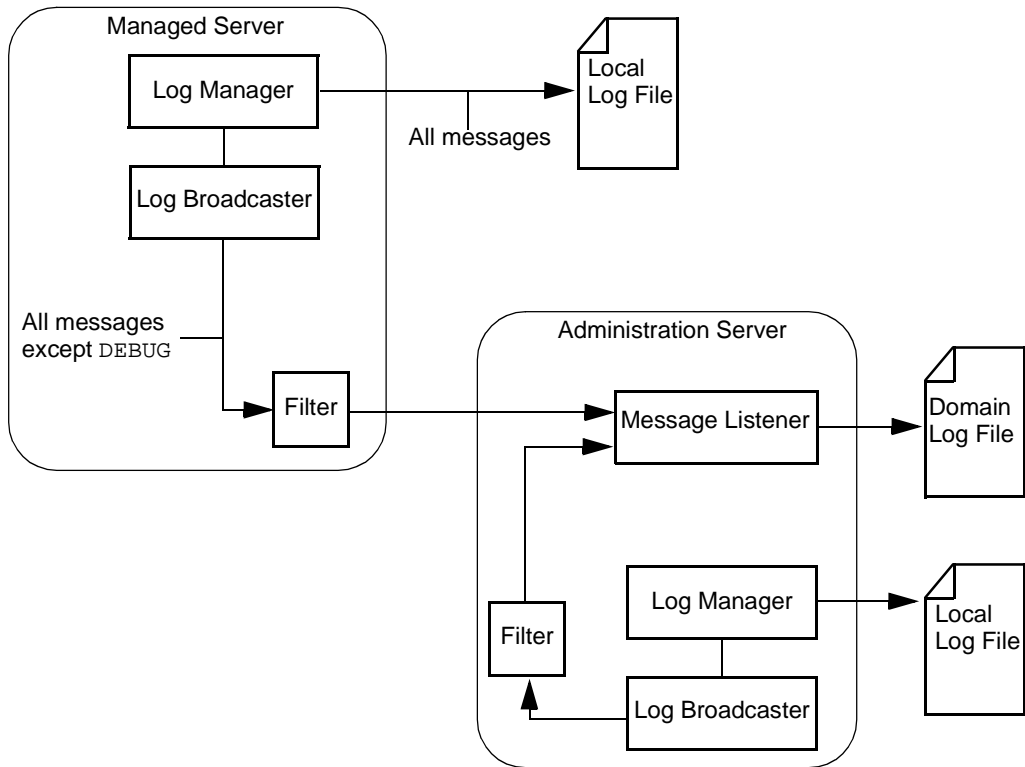
To forward messages to the domain log, each server instance broadcasts its log messages as Java Management Extensions (JMX) notifications. A server broadcasts all messages and message text except for the following:

- Messages of the `DEBUG` severity level.
- Any stack traces that are included in a message.

The Administration Server listens for a subset of these messages and writes them to the domain log file. To listen for these messages, the Administration Server registers a JMX listener with each Managed Server. By default, the listener includes a filter that allows only messages of severity level `ERROR` and higher to be forwarded to the Administration Server. (See Figure 253-3.)

Note: If a Managed Server is running in Managed Server Independence (MSI) mode, it writes to the domain log file directly. See "[MSI Mode and the Domain Log File](#)."

Figure 253-3 WebLogic Server Logging Services



For any given WebLogic Server instance, you can override the default filter and create a domain log filter that causes a different set of messages to be written to the domain log file. For information on setting up a domain log filter for a WebLogic Server instance, refer to “Domain Log Filters” on page 81-1.

If the Administration Server is unavailable, Managed Servers continue to write messages to their local log files, but they do not keep track of which messages they generate while the Administration Server is unavailable. For example, if the Administration Server is unavailable for two hours and then is restored, the domain log will not contain any messages that were generated during the two hours.

Viewing Server Logs

The following tasks describe how to view server logs:

- “Viewing Server Logs from the Administration Console” on page 253-9
- “Viewing Server Logs from a Text Editor” on page 253-11

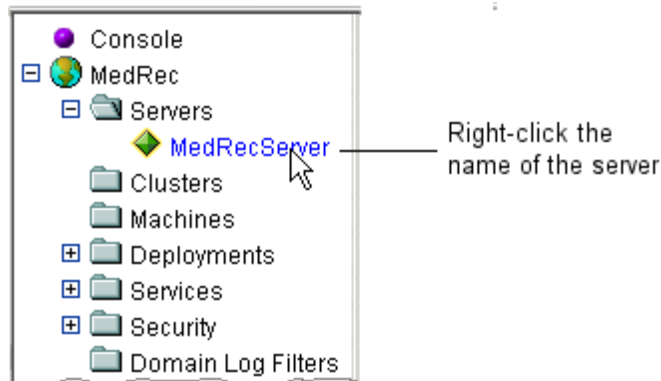
Viewing Server Logs from the Administration Console

You can use the Administration Console to view the log file for any server in the domain, regardless of whether the server is located on a remote computer for which you might not have login privileges. In addition, the Administration Console’s log file viewer provides filtering tools that you can use to limit the set of messages that it displays. For example, you can use the filtering tools to view only the messages that the JDBC subsystem has generated.

To view a server’s log messages from the Administration Console:

1. Start the server if it is not already running.
2. In the left pane of the Administration Console, expand the Servers folder and right-click the server whose log you want to view. (See Figure 253-4.)

Figure 253-4 Right-click the Name of the Server



3. From the pop-up menu, select View Server Log.

The right pane of the Administration Console displays the Search Log page. By default, the Search Log page displays up to 500 messages in reverse chronological order. The messages at the top of the window are the most recent messages that the server has generated.

The log viewer does not display messages that have been rotated into archive log files. For more information, refer to “Rotating Log Files” on page 253-16.

4. To view the probable cause and any action you can take to remedy an error that a message describes, in the Message ID column, click the message ID. (See Figure 253-5.)

Figure 253-5 View Message Details



A pop-up window displays the probable cause and any actions to take.

5. To change the default set of message attributes that the log viewer displays:
 - a. Click Customize this view, which is located at the top of the Search Log page.
 - b. To add message attributes to the log viewer display, on the customize page, next to Columns, move message attributes from the Available column to the Chosen column.
 - c. To remove message attributes from the log viewer display, on the customize page, next to Columns, move message attributes from the Chosen column to the Available column.
 - d. Click Apply.
6. To filter the list of messages:

- a. Click Customize this view, which is located at the top of the Search Log page.
- b. On the customize page, select filtering criteria.

By default, Subsystem, Users, Sub String and Since fields are undefined. If you leave them undefined, then the log viewer returns messages from all subsystems and security contexts since the earliest message in the file was generated.

If you select Forward, the Administration Console reverses the order in which it displays messages. The oldest messages in the log file display at the top of the Search Log page.

- c. Click Apply.

The Administration Console constructs a query that returns the log-message data you request. The query searches in the current log file only. If the log file has been rotated, the query does not search through old messages that have been moved to another file.

None of the items on this page affects the messages that are actually stored in the log file; instead, they determine the log file data that the Administration Console displays.

The Administration Console does not save your filter criteria. The next time you access the Search Log page, the Administration Console uses the default criteria to display messages.

Viewing Server Logs from a Text Editor

You can use a text editor to view messages in the current log file as well as older log files that the server creates per its log file rotation scheme. A text editor displays all attributes of all messages that are in the log file.

You cannot access the additional message details if you view messages from a text editor. Only the Administration Console log viewer provides access to the message details as described in step 4. in “Viewing Server Logs from the Administration Console”.

To view log messages from a text editor:

Note: Do not open the log file that the server is using to store current messages. Instead, create a copy of the log file and open the copy in the text editor.

You do this because, depending on your text editor and operating system, you could prevent the server from logging messages while the original file is open in the text editor. Any changes to the file's timestamp can confuse log file rotation.

1. To determine the location of the server's log file:
 - a. In the left pane of the Administration Server, expand the Servers folder and select the name of a server.
 - b. In the right pane, select Logging →Server and note the value in the Server File Name field.

If the pathname is relative, it is relative to the server's root directory.

For example, if you created a domain in the

`c:\user_projects\domains\MyDomain` directory, and you used the `c:\user_projects\domains\MyDomain\startWebLogicServer.cmd` script to start the server, then, by default, the log file is located in `c:\user_projects\domains\MyDomain\MyServer\MyServer.log`.

If the Server File Name field contains only a filename, the log file is located in the *root-directory\server-name* directory.

For more information, refer to "[A Server's Root Directory](#)."

2. Log on to the computer that hosts the server instance and change to the directory that contains the log file.
3. Create a copy of the log file and open the copy in the text editor.

Viewing the Domain Log

The following tasks describe how to view the domain log:

- "Viewing Domain Logs from the Administration Console" on page 253-13
- "Viewing Domain Logs from a Text Editor" on page 253-15

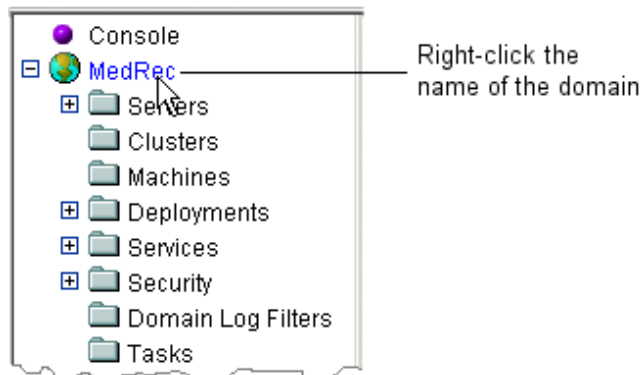
Viewing Domain Logs from the Administration Console

You can use the Administration Console to view the domain log file without logging on to the Administration Server's host computer. In addition, the Administration Console's log file viewer provides filtering tools that you can use to limit the set of messages that it displays. For example, you can use the filtering tools to view only the messages that the JDBC subsystem has generated.

To view a domain's log messages from the Administration Console:

1. In the left pane of the Administration Console, right-click on the name of the domain. (See Figure 253-6.)

Figure 253-6 Right-click the Name of the Domain



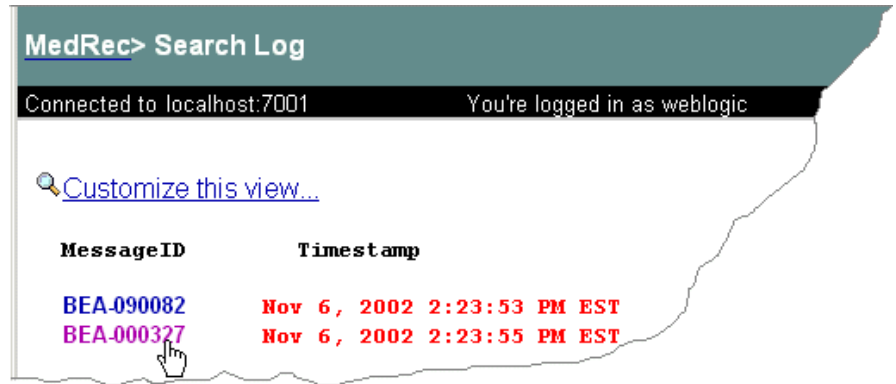
2. From the pop-up menu, select View Domain Log.

The right pane of the Administration Console displays the Search Log page. By default, the Search Log page displays up to 500 messages in reverse chronological order. The messages at the top of the window are the most recent messages that servers have forwarded to the domain log.

The log viewer does not display messages that have been rotated into archive log files. For more information, refer to “Rotating Log Files” on page 253-16.

3. To view the probable cause and any action you can take to remedy an error that a message describes, in the Message ID column, click the message ID. (See Figure 253-7.)

Figure 253-7 View Message Details



A pop-up window displays the probable cause and any actions to take.

4. To change the default set of message attributes that the log viewer displays:
 - a. Click Customize this view, which is located at the top of the Search Log page.
 - b. To add message attributes to the log viewer display, on the customize page, next to Columns, move message attributes from the Available column to the Chosen column.
 - c. To remove message attributes from the log viewer display, on the customize page, next to Columns, move message attributes from the Chosen column to the Available column.
 - d. Click Apply.
5. To filter the list of messages:
 - a. Click Customize this view, which is located at the top of the Search Log page.
 - b. On the customize page, select filtering criteria.

By default, Subsystem, Server, Users, Sub String and Since fields are undefined. If you leave them undefined, then the log viewer returns messages from all subsystems, servers, and security contexts since the earliest message in the file was generated.

If you select Forward, the Administration Console reverses the order in which it displays messages. The oldest messages in the log file display at the top of the Search Log page.

- c. Click Apply.

The Administration Console constructs a query that returns the log-message data you request. The query searches in the current domain log file only. If the log file has been rotated, the query does not search through old messages that have been moved to another file.

None of the items on this page affect the messages that are actually stored in the domain log file; instead, they determine the log file data that the Administration Console displays.

The Administration Console does not save your filter criteria. The next time you access the Search Log page, the Administration Console uses the default criteria to displays messages.

Viewing Domain Logs from a Text Editor

You can use a text editor to view messages in the current domain log file as well as older log files that the Administration Server creates per its domain-log file rotation scheme. A text editor displays all attributes of all messages that are in the log file.

You cannot access the additional message details if you view messages from a text editor. Only the Administration Console log viewer provides access to the message details as described in “Viewing Server Logs from the Administration Console” on page 253-9.

To view domain-log messages from a text editor:

Note: Do not open the log file that the server is using to store current messages. Instead, create a copy of the log file and open the copy in the text editor.

You do this because, depending on your text editor and operating system, you could prevent the server from logging messages while the original file is open in the text editor. Any changes to the file’s timestamp can confuse log file rotation.

1. To determine the location of the domain’s log file:
 - a. In the left pane of the Administration Server, select the domain.
 - b. In the right pane, select Configuration →Logging.

- c. On the Logging subtab, note the value in the File Name field.

If the pathname is relative, it is relative to the root directory of the Administration Server. This is the same directory that contains the domain's `config.xml` file.

For example, if you created a domain in the

`c:\user_projects\domains\MyDomain` directory, then, by default, the log file is located in

`c:\user_projects\domains\MyDomain\wl-domain.log`.

2. Log on to the computer that hosts the Administration Server and change to the directory that contains the domain log file.
3. Create a copy of the log file and open the copy in the text editor.

Rotating Log Files

By default, when you start a WebLogic Server instance in **development mode**, the server automatically renames (rotates) its local server log file as `server-name.log.n`. For the remainder of the server session, log messages accumulate in `server-name.log` until the file grows to a size of 500 kilobytes.

Each time the local log file reaches this size, the server renames the log file and creates a new `server-name.log` to store new messages. By default, the rotated log files are numbered in order of creation `filenamennnnn`, where `filename` is the name configured for the log file. You can configure a server instance to include a time and date stamp in the file name of rotated log files.

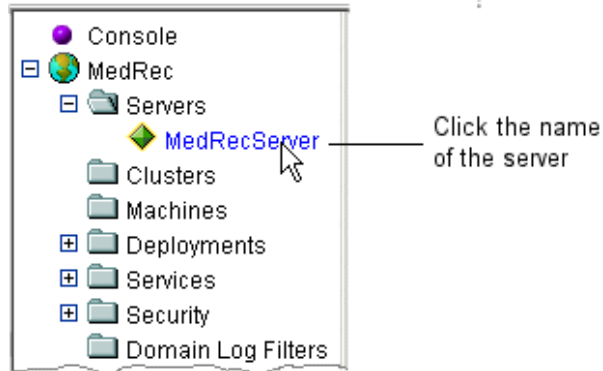
By default, when you start a server instance in **production mode**, the server rotates its local log file whenever the file grows to 5000 kilobytes in size. It does not rotate the local server log file when you start the server. For more information about changing the mode in which a server starts, refer to “Changing the Runtime Mode” on page 495-17.

You can change these default settings for log file rotation. For example, you can change the file size at which the server rotates the log file or you can configure a server to rotate log files based on a time interval. You can also specify the maximum number of rotated files that can accumulate. After the number of log files reaches this number, subsequent file rotations overwrite the oldest log file.

To set up log file rotation:

1. To specify log file rotation for a server log, do the following from the Administration Console:
 - a. In the left pane, expand the Server folder and select a server. (See Figure 253-8.)

Figure 253-8 Click on the Name of a Server



- b. In the right pane, select Logging →Server.
- To specify log file rotation for the domain log:
- a. In the left pane, select the Domain.
 - b. In the right pane, select the Logging tab.
 2. To move old messages to another file when the current log file reaches a specific size:
 - a. In Rotation Type, choose By Size.
 - b. In Minimum File Size, enter the minimum file size that triggers the server to move log messages to a separate file. After the log file reaches the specified minimum size, the next time the server checks the file size, it renames the current log file. After the server renames a file, subsequent messages accumulate in a new file named *FileName*.

- c. If you want to limit the number of log files that the server creates to store old log messages, select **Limit Number of Retained Log Files**. Then in **Log Files to Retain**, enter the maximum number of files. If the server receives additional log messages after reaching the capacity of the last log file, it overwrites the oldest log file.
 - d. Click **Apply**.
3. If you want to move old messages to another file at specific time intervals:
 - a. In **Rotation Type**, choose **By Time**.
 - b. In **Rotation Time**, enter the start time.

Use the following format: *hh:mm*, where *hh* is the hour in a 24-hour format and *mm* is the minute.

At the time that you specify, the server rotates the current log file. If the time that you specify is already past, the server starts its file rotation immediately. Thereafter, the server rotates the log file at an interval that you specify in **File Time Span**.
 - c. In **File Time Span**, enter the interval at which the server saves old messages to another file.
 - d. If you want to limit the number of log files that the server creates to store old log messages, select **Limit Number of Retained Log Files**. Then in **Log Files to Retain**, enter the maximum number of files. If the server receives additional log messages after reaching the capacity of the last log file, it overwrites the oldest log file.
 - e. Click **Apply**.
4. To include a time or date stamp in the file name when the log file is rotated, in the **File Name** field, add `java.text.SimpleDateFormat` variables to the file name. 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

If you do not include a time and date stamp, the rotated log files are numbered in order of creation *filenamennnn*, where *filename* is the name configured for the log file. For example: myserver.log00007.

Notification of Rotation

When the log file exceeds the rotation threshold that you specify, the server instance prints a log message that states that the log file will be rotated. Then it rotates the log file and prints an additional message that indicates the name of the file that contains the old messages.

For example, if you set up log files to rotate by size and you specify 500K as the minimum rotation size, when the server determines that the file is at least 500K in size, the server prints the following message:

```
<Dec 13, 2002 2:48:46 PM EST> <Alert> <Log Management> <BEA-170017>  
<The log file .\MedRecServer\MedRecServer.log will be rotated.  
Please reopen the log file if tailing has stopped. This can happen  
on some platforms like Windows.>
```

The server immediately rotates the file and prints the following message:

```
<Dec 13, 2002 2:48:46 PM EST> <Alert> <Log Management> <BEA-170018>  
<The log file has been rotated to MedRecServer.log00001. Log  
messages will continue to be logged in  
.\MedRecServer\MedRecServer.log.>
```

Note that the severity level for both messages is `Alert` and therefore both messages are printed to standard out by default. The message ID for the message before rotation is always `BEA-170017` and the ID for the message after rotation is always `BEA-170018`.

File systems such as the standard Windows file system place a lock on files that are open for reading. On such file systems, if your application is tailing the log file, or if you are using a command such as the DOS `tail -f` command in a command prompt, the tail operation fails after the server has rotated the log file. The `tail -f` command prints messages to standard out as lines are added to a file. For more information, enter `help tail` in a DOS prompt.

To remedy this situation for an application that tails the log file, you can create a JMX listener that notifies your application when the server emits the log rotation message. When your application receives the message, it can restart its tailing operation. To see an example of a JMX listener, refer to "[Subscribing to Messages](#)" in the *Using WebLogic Logging Services* guide.

Specifying Which Messages a Server Sends to Standard Out

In addition to writing messages to log files, a WebLogic Server instance can print a subset of its messages to standard out. Usually, standard out is the shell (command prompt) in which you are running the server instance. However, some operating systems enable you to redirect standard out to some other location. If you use the Node Manager to start a Managed Server, the Node Manager redirects a server's standard out to a file on the Node Manager's host computer. For more information, refer to "Viewing Standard Out for a Server Started by the Node Manager" on page 253-21.

By default, all messages of `WARNING` severity or higher are printed to standard out and messages of the `DEBUG` severity are not printed to standard out.

To specify which messages a server sends to standard out:

1. In the left pane of the Administration Console, expand the Servers folder and select a server. (See Figure 253-8.)
2. In the right pane, select Logging →Server.
3. Click the Log to Stdout checkbox to place a checkmark there.
4. If you want to send any messages of `DEBUG` severity to standard out, click the Debug to Stdout to place a checkmark there.

WebLogic Server does not generate `DEBUG` messages, but you can configure an application that you write to do so. For more information, refer to [Writing Debug Messages](#) in the *Using WebLogic Logging Services* Guide.

5. In Stdout severity threshold, choose the minimum severity level of the messages that you want to send to standard out. For information about message severity, refer to “Message Severity” on page 253-5.
6. Click Apply to save the settings.

Printing Messages to Standard Error

A WebLogic Server instance can also print messages to standard error, which, by default, is the command prompt in which the server is running. The server prints to standard error under the following circumstances:

- The logging subsystem has encountered an error and is unable to send messages to standard out.
- A server instance encounters errors in its startup cycle before the logging subsystem has initialized.

For information about redirecting standard error messages to a file, refer to “Redirecting System.out and System.err to a File” on page 253-29.

Viewing Standard Out for a Server Started by the Node Manager

If you use the Node Manager to start a Managed Server, the Node Manager redirects to a file on the Node Manager’s host computer all messages that the server instance and its JVM would normally print to the shell (including `verbosegc` output). You can use the Administration Console to view the contents of the file. For more information, refer to "[Managed Server Log Files](#)."

To view standard out messages for a server that was started by the Node Manager:

1. If it is not already running, start the Node Manager on the computer that hosts the Managed Server instance.

2. In the left pane of the Administration Console, expand the Servers folder and select the server. (See Figure 253-8.)
3. In the right pane, select Control → Remote Start Output.
4. On the Remote Start Output tab, click the View Server output link.

The right pane of the Administration Console displays standard out messages from the most recent (or current) server session.

Configuration Auditing

You can configure the Administration Server to emit log messages when a user changes the configuration or invokes management operations on any resource within a domain. For example, if a user disables SSL on a Managed Server in a domain, the Administration Server emits log messages. These messages provide an audit trail of changes within a domain's configuration (configuration auditing).

The following sections describe configuration auditing:

- “Enabling Configuration Auditing” on page 253-23
- “Configuration Auditing Messages” on page 253-23

The Administration Server writes configuration auditing messages to its local log file. Because all configuration auditing messages are of the `INFO` severity, they are not written to the domain-wide message log by default. For information on changing this default, see “Domain Log Filters” on page 81-1.

In addition to writing messages to its local log file, the Administration Server broadcasts configuration auditing messages as JMX notifications. You can create a JMX listener and filter that responds to these messages. For example, if the Administration Server emits a message that indicates an unauthorized user has attempted to change the domain's configuration, the JMX listener and filter can send email. See [“Listening for Configuration Auditing Messages”](#) in *Programming WebLogic Management Services with JMX*.

Enabling Configuration Auditing

To enable the Administration Server to emit configuration-auditing messages:

1. In the left pane of the Administration Console, click the name of the domain.
2. In the right pane, on the Configuration tab, click the General tab.
3. On the General tab, place a check mark in the Enable Auditing of Administration Changes check box.
4. Click Apply.
5. Restart the Administration Server.

You can also do the following to enable configuration auditing:

- When you start the Administration Server, include the following Java option in the `weblogic.Server` command:

```
-Dweblogic.AdministrationMBeanAuditingEnabled=true
```

See "[weblogic.Server Command-Line Reference](#)."

- After the Administration Server has started, use the `weblogic.Admin` utility to change the value of the `AdministrationMBeanAuditingEnabled` attribute of the `DomainMBean`.

For example, the following command disables configuration auditing for the `examples` domain:

```
java weblogic.Admin SET  
-mbean examples:Name=examples,Type=Domain  
-property AdministrationMBeanAuditingEnabled true
```

For information about using `weblogic.Admin` to change values of MBean attributes, see "[Commands for Managing WebLogic Server MBeans](#)."

Configuration Auditing Messages

All configuration auditing messages are of the `Info` severity and are identified by message IDs that fall within the range of 159900-159910.

The messages use managed bean (MBean) object names to identify resources. MBean object names provide an unambiguous identification regardless of the interface (Administration Console, command-line utility, or API) that is used to invoke operations or modify the resource. See "[WebLogic Server Management Namespace](#)."

Table 253-1 summarizes the messages.

Table 253-1 Summary of Configuration Auditing Messages

When This Event Occurs...	WebLogic Server Generates a Message With This ID...	And This Message Text...
Authorized user creates a resource.	159900	USER <i>username</i> CREATED <i>MBean-name</i> where <i>username</i> identifies the WebLogic Server user who logged in and created a resource.
Unauthorized user attempts to create a resource.	159901	USER <i>username</i> CREATED <i>MBean-name</i> FAILED weblogic.management. NoAccessRuntimeException: <i>exception-text stack-trace</i> where <i>username</i> identifies the unauthorized WebLogic Server user.
Authorized user deletes a resource.	159902	USER <i>username</i> REMOVED <i>MBean-name</i> where <i>username</i> identifies the WebLogic Server user who logged in and created a resource.
Unauthorized user attempts to delete a resource.	159903	USER <i>username</i> REMOVE <i>MBean-name</i> FAILED weblogic.management. NoAccessRuntimeException: <i>exception-text stack-trace</i> where <i>username</i> identifies the WebLogic Server user who logged in and created a resource.
Authorized user changes a resource's configuration.	159904	USER <i>username</i> MODIFIED <i>MBean-name</i> ATTRIBUTE <i>attribute-name</i> FROM <i>old-value</i> TO <i>new-value</i> where <i>username</i> identifies the WebLogic Server user who logged in and changed the resource's configuration.

Table 253-1 Summary of Configuration Auditing Messages (Continued)

When This Event Occurs...	WebLogic Server Generates a Message With This ID...	And This Message Text...
Unauthorized user attempts to change a resource's configuration.	159905	<p>USER <i>username</i> MODIFY <i>MBean-name</i> ATTRIBUTE <i>attribute-name</i> FROM <i>old-value</i> TO <i>new-value</i> FAILED weblogic.management. NoAccessRuntimeException: <i>exception-text stack-trace</i> where <i>username</i> identifies the unauthorized WebLogic Server user.</p>
Authorized user invokes an operation on a resource. For example, a user deploys an application or starts a server instance.	159907	<p>USER <i>username</i> INVOKED ON <i>MBean-name</i> METHOD <i>operation-name</i> PARAMS <i>specified-parameters</i> where <i>username</i> identifies the WebLogic Server user who logged in and invoked a resource operation.</p>
Unauthorized user attempts to invoke an operation on a resource.	159908	<p>USER <i>username</i> INVOKED ON <i>MBean-name</i> METHOD <i>operation-name</i> PARAMS <i>specified-parameters</i> FAILED weblogic.management. NoAccessRuntimeException: <i>exception-text stack-trace</i> where <i>username</i> identifies the unauthorized WebLogic Server user.</p>
Authorized user enables configuration auditing.	159909	<p>USER <i>username</i>, Configuration Auditing is enabled where <i>username</i> identifies the WebLogic Server user who enabled configuration auditing.</p>
Authorized user disables configuration auditing.	159910	<p>USER <i>username</i>, Configuration Auditing is disabled where <i>username</i> identifies the WebLogic Server user who disabled configuration auditing.</p>

Note: Each time an authorized user adds, modifies, or deletes a resource the Management subsystem also generates `Info` message with the ID 140009. For example:

```
<Sep 15, 2003 11:54:47 AM EDT> <Info> <Management> <140009>
<Configuration changes for domain saved to the repository.>
```

The Management subsystem generates this message regardless of whether configuration auditing is enabled.

While the message informs you that the domain's configuration has changed, it does not provide the detailed information that configuration auditing messages provide. Nor does the Management subsystem generate this message when you invoke operations on resources.

Table 253-2 lists additional message attributes for configuration auditing messages. All configuration auditing messages specify the same values for these attributes.

Table 253-2 Common Message Attributes and Values

Message Attribute	Attribute Value
Severity	Info
Subsystem	Configuration Audit
User ID	kernel identity This value is always <code>kernel identity</code> , regardless of which user modified the resource or invoked the resource operation.
Server Name	<i>AdminServerName</i> Because the Administration Server maintains the configuration data for all resources in a domain, this value is always the name of the Administration Server.
Machine Name	<i>AdminServerHostName</i> Because the Administration Server maintains the configuration data for all resources in a domain, this value is always the name of the Administration Server's host machine.

Table 253-2 Common Message Attributes and Values

Message Attribute	Attribute Value
Thread ID	<i>execute-thread</i> The value depends on the number of execute threads that are currently running on the Administration Server.
Timestamp	<i>timeStamp</i> at which the message is generated.

Other Logging Tasks

The following sections describe other tasks related to WebLogic Server log messages:

- “Viewing Standard Out for a Server Set Up as a Windows Service” on page 253-28
-
- “Redirecting System.out and System.err to a File” on page 253-29
- “Disabling a Server from Forwarding Messages to the Domain Log” on page 253-30
- “Changing the Name and Location of the Server Log File” on page 253-30
- “Changing the Name and Location of the Domain Log File” on page 253-31
- “Enabling JDBC Logging” on page 253-32
- “Enabling and Configuring an HTTP Log” on page 253-33

For more logging tasks, refer to “Specifying the Messages That a Server Forwards to the Domain Log” on page 81-1.

Viewing Standard Out for a Server Set Up as a Windows Service

By default, if you set up a server to run as a Windows service, you cannot view the server's standard out. For information on redirecting the server and JVM's standard out to a file, refer to "[Redirecting JVM Messages to a File](#)."

Viewing Localized Time Stamps on Windows

On a Windows computer, you can use the Regional Options Control Panel to enable support for localized currencies, times, and dates. Most SDKs detect the setting on the Regional Options Control Panel and modify their standard out and standard error output to print time and date stamps in a format that matches your locale.

If you have selected a locale and language other than English, modify the settings for your Windows command prompt so that messages that WebLogic Server prints to standard out contain properly-formatted time and date stamps:

Note: You do not need to follow this procedure for server instances that a Node Manager starts. Node Manager redirects its standard out to a file, which you can view from the Administration Console. See "Viewing Standard Out for a Server Started by the Node Manager" on page 253-21.

1. In your WebLogic Server startup script, add the following command **before** the command that invokes the `weblogic.Server` class:

```
chcp 1252
```

This command causes the command prompt to encode all output in the ANSI character set.

For example:

```
chcp 1252
```

```
%JAVA_HOME%\bin\java" %JAVA_VM% %MEM_ARGS% %JAVA_OPTIONS%  
-Dweblogic.Name=%SERVER_NAME%  
-Dweblogic.management.username=WLS_USER%  
-Dweblogic.management.password=WLS_PW%  
-Dweblogic.ProductionModeEnabled=%PRODUCTION_MODE%
```

```
-Djava.security.policy="%WL_HOME%\server\lib\weblogic.policy"  
weblogic.Server  
  
ENDLOCAL
```

2. Open a Windows command prompt and do the following:
 - a. To display the properties sheet, right-click in the title bar and select Properties.
 - b. In the Properties window, select the Font tab.
 - c. In the Font list, select a TrueType font.
 - d. Click OK.
 - e. In the Apply Properties window, select Save properties for future windows with same title. Then click OK.
3. Invoke the WebLogic Server start script.

Redirecting System.out and System.err to a File

In addition to the configurable set of log messages that a server instance prints to standard out, servlets can invoke `system.out.println` and the JVM within which a WebLogic Server instance runs can send messages to standard error and standard out. If you use a WebLogic Server script to start a server instance, there is no default, persistent storage for the standard error and standard out messages.

If you want to keep a record of these messages, edit the WebLogic Server script so that the `JAVA_OPTIONS` variable specifies the following:

```
-Dweblogic.Stdout="stdout-filename"  
-Dweblogic.Stderr="stderr-filename"
```

Where *stdout-filename* is the name of a file that you want to save standard out messages and *stderr-filename* is the name of a file that you want to save standard error messages.

To view the contents of these files, use a text editor or command prompt utility such as the DOS `tail` program. You cannot view them from the Administration Console.

Note: WebLogic Server prompts for entering your username and password are sent to standard out. If you use `-Dweblogic.Stdout`, you will no longer see the prompts to enter your username and password. To bypass this

prompt, use a boot identity file as described in “Boot Identity Files” on page 497-15.

Redirecting Garbage Collection Comments

While the `-Dweblogic.Stdout` and `-Dweblogic.Stderr` options cause a JVM to redirect all of its `java.lang.System.out` and `java.lang.System.err` messages to a file, a JVM does not print its garbage collection comments to `System.out` or `System.err`. If you start a JVM with the `-verbosegc` option, the JVM prints the `verbosegc` output to the shell in which the JVM is running, regardless of whether you specify `-Dweblogic.Stdout` or `-Dweblogic.Stderr`. Some JVMs provide non-standard options for printing garbage collection comments to a file. For more information, view the help for your JVM’s non-standard options by entering `java -X` in a shell.

Disabling a Server from Forwarding Messages to the Domain Log

By default, a server instance forwards log messages of an `ERROR` or higher severity to domain log.

To prevent a server from sending messages to the domain log:

1. In the left pane of the Administration Console, expand the Servers folder and select the server. (See Figure 253-8.)
2. In the right pane, click Logging → Domain.
3. Click on the Log to Domain Logfile checkbox to remove the checkmark.
4. Click Apply.

Changing the Name and Location of the Server Log File

By default, the local server log file is named `./SERVER_NAME/SERVER_NAME.log`, where `SERVER_NAME` is the name of the server. The path is relative to the server’s root directory.

To change the default name or location of a server's local log file:

1. In the left pane of the Administration Console, expand the Servers folder and select the server. (See Figure 253-8.)
2. In the right pane, select Logging →Server.
3. In the File Name box, enter a path and filename for the server log.

Enter an absolute pathname or a pathname that is relative to the server's root directory. If you use the Node Manager to start a Managed Server, the root directory is located on the computer that hosts the Node Manager process. For more information, refer to "[A Server's Root Directory](#)."

For information about including a time stamp in the server log's file name, refer to "Rotating Log Files" on page 253-16.

4. Click Apply to apply your changes.
5. Restart the server.

The server writes all subsequent domain messages to the new file.

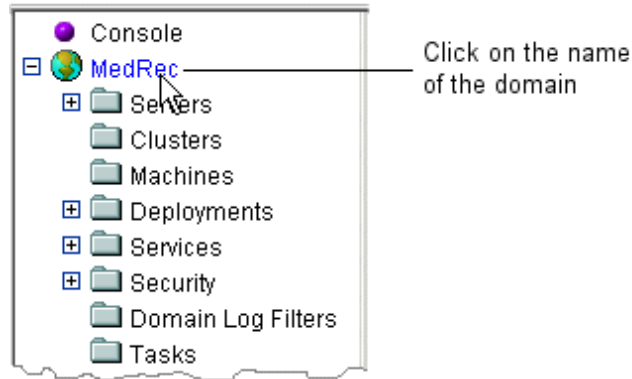
Changing the Name and Location of the Domain Log File

The default name for a domain log file is `./wl-domain.log`. The path is relative to the root directory of the Administration Server.

To change the default name or location of the domain's log file:

1. In the left pane of the Administration Console, select the domain. (See Figure 253-9.)

Figure 253-9 Click on the Name of the Domain



2. In the right pane, select Configuration →Logging.

3. In the File Name box, enter a path and filename for the domain log.

Enter an absolute pathname or a pathname that is relative to the root directory of the Administration Server. For more information, refer to "[A Server's Root Directory](#)."

For information about including a time and date stamp in the name of rotated domain log files, refer to "Rotating Log Files" on page 253-16.

4. Click Apply to apply your changes.

Restart the Administration Server. The Administration Server writes all subsequent domain messages to the new file.

Enabling JDBC Logging

To enable JDBC logging:

1. In the left pane of the Administration Console, expand the Servers folder and select the server. (See Figure 253-8.)
2. In the right pane, select Logging →JDBC.
3. On the JDBC tab, select Enable JDBC Logging.
4. Click Apply.

5. Restart the server.

Enabling and Configuring an HTTP Log

To keep a log of all HTTP requests on a given server instance:

1. In the left pane of the Administration Console, expand the Servers folder and select the server. (See Figure 253-8.)
2. In the right pane, select Logging → HTTP.
3. On the HTTP tab, click Enable HTTP Logging.
4. In the Format list, determine the format of the HTTP log file by selecting Common or Extended.
5. To determine the frequency with which the server empties its HTTP-request buffer and writes the data to the HTTP log file:
 - a. In Log Buffer Size, specify the maximum size (in kilobytes) of the HTTP-request buffer.
 - b. In the Flush Every box, specify the interval (in seconds) at which the server checks the size of the HTTP-request buffer. If the buffer has reached the maximum size, the server writes the data to the HTTP log file.
6. By default, the server moves old HTTP requests to another file when the current HTTP log file grows beyond 5000 kilobytes. Also by default, the server can create an unlimited number of these archive log files.

To change these defaults:

- a. In Maximum Log File Size, enter the file size that triggers the server to move HTTP requests to a separate file. After the HTTP log file reaches the specified size, the next time the server checks the file size, it will rename the current log file. After the server renames a file, subsequent messages accumulate in a new file named `access.log`.

- b. To limit the number of HTTP log files that the server creates to store old HTTP requests, select Limit Number of Retained Log Files. Then in Log Files to Retain, enter the maximum number of files. If the server receives additional HTTP requests after reaching the capacity of the last log file, it overwrites the oldest log file.
7. To create another file at specific time intervals instead of when the log file reaches a specific size:
- a. In Rotation Type, choose `date`.
 - b. In Rotation Period, enter the interval at which the server saves old HTTP requests to another file.
 - c. In Rotation Time, enter the start time.

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](#).

At the time that you specify, the server rotates the current log file. If the time that you specify is already past, the server starts its file rotation immediately. Thereafter, the server rotates the HTTP log file at an interval that you specify in Rotation Period.

- d. To limit the number of log files that the server creates to store old HTTP requests, select Limit Number of Retained Log Files. In Log Files to Retain, enter the maximum number of files. If the server receives additional HTTP requests after reaching the capacity of the last log file, it overwrites the oldest log file.
8. To include a time or date stamp in the file name when the log file is rotated, in the File Name field, add `java.text.SimpleDateFormat` variables to the file name. Surround each variable with percentage (%) characters.

For example, if you enter the following value in the File Name field:

```
access_%yyyy%_%MM%_%dd%_%hh%_%mm%.log
```

the virtual host's HTTP log file will be named:

```
access_yyyy_MM_dd_hh_mm.log
```

When the server instance rotates the HTTP log file, the rotated file name contains the date stamp. For example, if the server instance rotates the log file on 2 April, 2003 at 10:05 AM, the log file that contains the old log messages

will be named:

`access_2003_04_02_10_05.log`

If you do not include a time and date stamp, the rotated log files are numbered in order of creation *filenamennnnn*, where *filename* is the name configured for the log file. For example: `access.log00007`.

9. Click Apply.

10. Restart the server.

Attributes and Console Screen Reference for Logging

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“Domain --> Configuration --> Logging” on page 68-1](#)

[“Server --> Logging --> Server” on page 473-1](#)

[“Server --> Logging --> Domain” on page 445-1](#)

[“Server --> Logging --> HTTP” on page 446-1](#)

[“Server --> Logging --> JDBC” on page 447-1](#)

[“Server --> Logging --> JTA” on page 448-1](#)

[“Virtual Host --> Configuration --> Logging” on page 536-1](#)

[“Domain Log Filters” on page 80-1](#)

[“Domain Log Filter --> Configuration” on page 77-1](#)

[“Domain Log Filters --> Target” on page 79-1](#)

[“Domain Log Filter --> Notes” on page 78-1](#)

[“Search Log” on page 252-1](#)

[“Customize Log View” on page 251-1](#)



Machine --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to assign a name to a physical machine that hosts one or more Managed Servers.

Tasks

“Configuring a Machine” on page 268-1

“Cloning a Machine” on page 268-3

“Deleting a Machine” on page 268-3

“Assign a WebLogic Server Instance to a Machine” on page 268-3

Related Topics

["Configure a Machine to Use Node Manager"](#)

Attributes

Table 255-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> weblogic.management.configuration.MachineMBean</p> <p><i>Attribute:</i> Name</p>	

Machine --> Configuration --> Node Manager

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Before using Node Manager to manage servers in your domain, you must create a configuration entry for each machine that runs a Node Manager process. The basic machine entry defines the name of a machine in your domain, the servers that run on that machine, and the connection information for connecting to a Node Manager process on the machine.

This page defines the listen address and port number that the Administration Server uses to connect to the Node Manager process running on the machine.

Tasks

“Configuring a Machine” on page 268-1

“Cloning a Machine” on page 268-3

“Deleting a Machine” on page 268-3

“Assign a WebLogic Server Instance to a Machine” on page 268-3

Related Topics

["Configuring Node Manager"](#)

Attributes

Table 256-1

Attribute Label	Description	Value Constraints
Listen Address	The address on which NodeManager listens for connections. <i>MBean:</i> weblogic.management.configuration.NodeManagerMBean <i>Attribute:</i> ListenAddress	<i>Default:</i> "localhost" <i>Dynamic:</i> yes
Listen Port	The listen port of the NodeManager <i>MBean:</i> weblogic.management.configuration.NodeManagerMBean <i>Attribute:</i> ListenPort	<i>Minimum:</i> 0 <i>Maximum:</i> 65534 <i>Default:</i> 5555 <i>Dynamic:</i> yes
Debug Enabled	Whether or not communication with this NodeManager needs to be debugged. <i>MBean:</i> weblogic.management.configuration.NodeManagerMBean <i>Attribute:</i> DebugEnabled	<i>Default:</i> false <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false <i>Dynamic:</i> yes

Machine --> Configuration --> Servers

[Tasks](#) [Related Topics](#)

Overview

Use this tab to identify the Managed Servers that run on this machine.

Tasks

“Configuring a Machine” on page 268-1

“Cloning a Machine” on page 268-3

“Deleting a Machine” on page 268-3

“Assign a WebLogic Server Instance to a Machine” on page 268-3

Related Topics

["Configuring Node Manager"](#)



Machine --> Monitoring --> Node Manager Log

[Related Topics](#)

Overview

This page displays the contents of the Node Manager log file if Node Manager is running on the machine.

Tasks

“Monitoring Node Manager Log” on page 268-4

Related Topics

- [“Node Manager Log Files”](#) in *Configuring and Managing WebLogic Server*
- [“Configuring Node Manager”](#) in *Configuring and Managing WebLogic Server*
- [“Starting and Stopping Node Manager”](#) in *Configuring and Managing WebLogic Server*
- [“Troubleshooting Node Manager”](#) in *Configuring and Managing WebLogic Server*



Machine --> Notes

Attributes

Overview

Use this tab to enter optional notes to describe the configuration or function of this machine.

Attributes

Table 259-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.MachineMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes



Machine

The table on this page displays machines configured for the current domain. A machine definition associates server instances with the physical machine upon which they run.

- For more information, see [Machines](#).



Unix Machine --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to configure a Unix machine that hosts a Managed Server.

Tasks

“Configuring a Machine” on page 268-1

“Cloning a Machine” on page 268-3

“Deleting a Machine” on page 268-3

“Assign a WebLogic Server Instance to a Machine” on page 268-3

Related Topics

["Configuring Node Manager"](#)

Attributes

Table 261-1

Attribute Label	Description	Value Constraints
Name	The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.	
Enable Post-Bind UID	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.	<i>Default:</i> false <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false <i>Configurable:</i> yes
Post-Bind UID	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.	<i>Default:</i> "nobody" <i>Configurable:</i> yes
Enable Post-Bind GID	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.	<i>Default:</i> false <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false <i>Configurable:</i> yes
Post-Bind GID	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.	<i>Default:</i> "nobody" <i>Configurable:</i> yes

Unix Machine --> Configuration --> Node Manager

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Before using Node Manager to manage servers in your domain, you must create a configuration entry for each machine that runs a Node Manager process. The machine entry defines the name of a machine, the servers that run on that machine, and the connection information for connecting to a Node Manager process on the machine.

This page defines the listen address and port number that the Administration Server uses to connect with the Node Manager process running on the machine.

Tasks

“Configuring a Machine” on page 268-1

“Cloning a Machine” on page 268-3

“Deleting a Machine” on page 268-3

“Assign a WebLogic Server Instance to a Machine” on page 268-3

Related Topics

["Configuring Node Manager"](#)

Attributes

Table 262-1

Attribute Label	Description	Value Constraints
Listen Address	The address on which NodeManager listens for connections. <i>MBean:</i> weblogic.management.configuration.NodeManagerMBean <i>Attribute:</i> ListenAddress	<i>Default:</i> "localhost" <i>Dynamic:</i> yes
Listen Port	The listen port of the NodeManager <i>MBean:</i> weblogic.management.configuration.NodeManagerMBean <i>Attribute:</i> ListenPort	<i>Minimum:</i> 0 <i>Maximum:</i> 65534 <i>Default:</i> 5555 <i>Dynamic:</i> yes
Debug Enabled	Whether or not communication with this Node Manager process needs to be debugged. <i>MBean:</i> weblogic.management.configuration.NodeManagerMBean <i>Attribute:</i> DebugEnabled	<i>Default:</i> false <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false <i>Dynamic:</i> yes

Unix Machine --> Configuration --> Servers

[Tasks](#) [Related Topics](#)

Overview

Use this tab to specify the WebLogic Server instances that are hosted on this machine.

Tasks

“Configuring a Machine” on page 268-1

“Cloning a Machine” on page 268-3

“Deleting a Machine” on page 268-3

“Assign a WebLogic Server Instance to a Machine” on page 268-3

Related Topics

["Configuring Node Manager"](#)



Machine --> Monitoring --> Node Manager Status

[Tasks](#) [Related Topics](#)

Overview

This page displays the status of the Node Manager process running on the selected machine.

Tasks

“Monitoring Node Manager Status” on page 268-4

“Configuring a Machine” on page 268-1

Related Topics

- [“Configuring Node Manager”](#) in *Configuring and Managing WebLogic Server*
- ["Node Manager Environment Variables"](#) in *Configuring and Managing WebLogic Server*
- ["Node Manager Properties"](#) in *Configuring and Managing WebLogic Server*
- ["Server Properties"](#) in *Configuring and Managing WebLogic Server*
- [“Starting and Stopping Node Manager”](#) in *Configuring and Managing WebLogic Server*
- [“Troubleshooting Node Manager”](#) in *Configuring and Managing WebLogic Server*



UNIX Machine --> Monitoring --> Node Manager Status

[Tasks](#) [Related Topics](#)

Overview

This page displays the status of Node Manager instance running on the selected machine.

Tasks

[“Monitoring Node Manager Status” on page 268-4](#)

[“Configuring a Machine” on page 268-1](#)

Related Topics

- [“Configuring Node Manager”](#) in *Configuring and Managing WebLogic Server*
- ["Node Manager Environment Variables"](#) in *Configuring and Managing WebLogic Server*
- ["Node Manager Properties"](#) in *Configuring and Managing WebLogic Server*
- ["Server Properties"](#) in *Configuring and Managing WebLogic Server*
- [“Starting and Stopping Node Manager”](#) in *Configuring and Managing WebLogic Server*
- [“Troubleshooting Node Manager”](#) in *Configuring and Managing WebLogic Server*



UNIX Machine --> Monitoring --> Node Manager Log

[Tasks](#) [Related Topics](#)

Overview

This page displays the contents of the Node Manager log file if Node Manager is running on the machine.

Tasks

“Monitoring Node Manager Log” on page 268-4

Related Topics

- [“Node Manager Log Files”](#) in *Configuring and Managing WebLogic Server*
- ["Configuring Node Manager"](#) in *Configuring and Managing WebLogic Server*
- ["Starting and Stopping Node Manager"](#) in *Configuring and Managing WebLogic Server*
- ["Troubleshooting Node Manager"](#) in *Configuring and Managing WebLogic Server*



Unix Machine --> Notes

Attributes

Overview

Use the Unix Machine --> Notes tab to store optional user information about this Unix machine.

Attributes

Table 267-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration.	<i>Dynamic:</i> yes



1 Machines

[“Attributes and Console Screen Reference for Machines” on page 269-1]

A machine is a logical representation of the physical machine (computer) that hosts one or more WebLogic Server instances.

If you run Node Manager on a machine that does not host an Administration Server, as is typical in production environments, you must create a machine configuration for each computer that runs a Node Manager process. The machine configuration includes information about the listen address and port number that the Administration Server uses to connect with the Node Manager process running on that machine.

In a cluster, WebLogic Server uses machines to ensure that server session data is replicated on separate pieces of hardware.

If the computer runs a UNIX operating system, you can create a UNIX machine configuration, which enables you to assign the process under which a WebLogic Server instance runs to a user ID (UID) or group ID (GID). The WebLogic Server process is assigned (bound) to the UID or GID after the computer has carried out all privileged startup actions.

Tasks

Configuring 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 link (or, if you are configuring a UNIX machine, click the Configure a New Unix Machine link). A dialog displays in the right pane showing the tabs associated with the new machine.
3. Enter a name for the new machine in the Name attribute field. This name is used to identify the machine within the WebLogic Server domain; it does not have to correspond to the machine's network name.

Note: The name must be unique in relation to all other configurable resources in the domain.

4. If you are creating a UNIX machine and you want to bind the processes under which WebLogic Server instances run to a user ID or group ID, do any of the following:
 - To bind the server processes to a user ID, select Enable Post-Bind UID and enter the user ID in the Post-Bind UID box.
 - To bind the server processes to a group ID, select Enable Post-Bind GID and enter the group ID in the Post-Bind GID box.
5. Click Create to create a machine instance with the name you specified. The new instance is added to the Machines node in the left pane.
6. Click the Node Manager tab to define the Node Manager connection and authentication attributes for the machine.

Enter the DNS name or IP address upon which Node Manager listens in the Listen Address box. If you identify the Listen Address by IP address, you must disable Host Name Verification on Administration Servers that will access Node Manager. For more information and instructions, see [“Using a Hostname Verifier”](#) in *Managing WebLogic Security*.

Click Apply to apply your changes.

7. Click the Servers tab to identify which Managed Servers reside on this machine. To assign an existing server to this machine, select the server name in the Available column, and click the appropriate arrow to move the server to the Chosen column.
8. Click Apply to apply your changes. The new machine entry now specifies the attributes required to connect to the Node Manager process running on the machine, as well as identify which WebLogic Server instances reside on the machine.

Cloning 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 a name for the new machine 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 connection information as needed.
6. Click the Servers tab and modify the list of servers that this machine hosts.
7. Click Apply to save the changes.

Deleting 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 instance is deleted from the Machines node in the left pane.

Assign a WebLogic Server Instance to 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 name of the machine to which you wish to assign a server instance.
3. Click the Servers Tab in the right pane.

4. Select a server from the Available box.
5. Click the right arrow to move the server to the Chosen box.
6. Click Apply.

Monitoring Node Manager Status

1. Click the Machines node. The Machines table displays in the right pane showing all the machines defined in the domain
2. Click the name of the machine on which you wish to monitor Node Manager.
3. Click the Monitoring tab in the right pane. If Node Manager is currently running on the machine, the Node Manager Status tab displays the following information about the Node Manager process:
 - State—current operating state
 - BEA Home—the BEA home directory used by the Node Manager process when starting Managed Servers on the machine
 - Java Home—the directory containing the JDK used by the Node Manager process
 - Listen Address—the address upon which the Node Manager process listens for requests
 - Listen Port—the port upon which the Node Manager process listens for requests
 - CLASSPATH—the classpath defined on the machine
4. To view the contents of the Node Manager Log, click the Node Manager Log tab in the right pane. If Node Manager is currently running on the machine, contents of the log file are displayed.

Monitoring Node Manager Log

1. Click the Machines node. The Machines table displays in the right pane showing all the machines defined in the domain

2. Click the name of the machine on which you wish to monitor Node Manager.
3. Click the Monitoring tab in the right pane.
4. To view the contents of the Node Manager Log, click the Node Manager Log tab in the right pane. If Node Manager is currently running on the machine, contents of the log file are displayed.

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 Administration Servers" on page 497-2.
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. 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."](#)

- e. 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.
- f. Click Create.
- g. 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.
- h. If you want to use the Node Manager to start server instances on this UNIX machine, click the 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 268-7.

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.
See “Starting Administration Servers” on page 497-2 and “Starting Managed Servers From a WebLogic Server Script” on page 497-8.
 - (For Managed Servers only) Start the Node Manager. Then use the Node Manager to start Managed Servers.
See "[Starting and Stopping Node Manager](#)" and “Starting Managed Servers from the Administration Console” on page 497-5.

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 use 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 "[Configuring Node Manager](#)."

With the above configuration, Node Manager will accept requests only on a secure port and only from a single, known host.

Attributes and Console Screen Reference for Machines

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

“Machine” on page 260-1

“Machine --> Configuration --> General” on page 255-1

“Machine --> Configuration --> Node Manager” on page 256-1

“Machine --> Configuration --> Servers” on page 257-1

“Machine --> Monitoring --> Node Manager Log” on page 258-1

“Machine --> Monitoring --> Node Manager Status” on page 264-1

“Machine --> Notes” on page 259-1

“Unix Machine --> Configuration --> General” on page 261-1

“Unix Machine --> Configuration --> Node Manager” on page 262-1

“Unix Machine --> Configuration --> Servers” on page 263-1

“UNIX Machine --> Monitoring --> Node Manager Log” on page 266-1

“UNIX Machine --> Monitoring --> Node Manager Status” on page 265-1

“Unix Machine --> Notes” on page 267-1



Mail Session --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

In the Mail Session > Configuration tab, you configure a new WebLogic Server mail session.

Tasks

“Configuring a New Mail Session” on page 274-1

“Cloning a Mail Session” on page 274-2

“Deleting a Mail Session” on page 274-3

Related Topics

“Mail” on page 274-1

See ["Using JavaMail with WebLogic Server"](#) in *Developing WebLogic Server Applications*.

Attributes

Table 270-1

Attribute Label	Description	Value Constraints
Name	The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration. <i>MBean:</i> weblogic.management.configuration.MailSessionMBean <i>Attribute:</i> Name	
JNDIName	The JNDIName attribute of the RMCFactoryMBean object <i>MBean:</i> weblogic.management.configuration.MailSessionMBean <i>Attribute:</i> JNDIName	
Properties	The properties attribute of the MailSessionMBean object <i>MBean:</i> weblogic.management.configuration.MailSessionMBean <i>Attribute:</i> Properties	

Mail Session --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

In the Mail Session > Notes tab, you enter any special notes relating to the configured mail session.

Tasks

“Configuring a New Mail Session” on page 274-1

“Cloning a Mail Session” on page 274-2

“Deleting a Mail Session” on page 274-3

Related Topics

“Mail” on page 274-1

See "Using JavaMail with WebLogic Server" in *Developing WebLogic Server Applications*.

Attributes

Table 271-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration. MailSessionMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

Mail Session --> Target and Deploy

[Tasks](#) [Related Topics](#)

Overview

WebLogic Server includes the JavaMail API version 1.1.3 reference implementation from Sun Microsystems. Using the JavaMail API, you can add email capabilities to your WebLogic Server applications. JavaMail provides access from Java applications to IMAP- and SMTP-capable mail servers on your network or the Internet. It does not provide mail server functionality; so you must have access to a mail server to use JavaMail.

Tasks

“Assigning a Mail Session” on page 274-3

Related Topics

See "Using JavaMail with WebLogic Server" in *Developing WebLogic Server Applications*.



Mail Session

WebLogic Server includes the JavaMail API version 1.1.3 reference implementation from Sun Microsystems. Using the JavaMail API, you can add email capabilities to your WebLogic Server applications. JavaMail provides access from Java applications to IMAP- and SMTP-capable mail servers on your network or the Internet. It does not provide mail server functionality; so you must have access to a mail server to use JavaMail.

To configure JavaMail for use in WebLogic Server, you create a Mail Session in the WebLogic Server Administration Console. This allows server-side components and applications to access JavaMail services with JNDI, using Session properties you preconfigure for them. For example, by creating a Mail Session, you can designate the mail hosts, transport and store protocols, and the default mail user in the Administration Console so that components that use JavaMail do not have to set these properties. Applications that are heavy email users benefit because WebLogic Server creates a single Session object and makes it available via JNDI to any component that needs it.

For more information, see “Mail” on page 274-1.



1 Mail

[“Attributes and Console Screen Reference for Mail Sessions” on page 275-1]

WebLogic Server includes the JavaMail API version 1.1.3 reference implementation from Sun Microsystems. Using the JavaMail API, you can add email capabilities to your WebLogic Server applications. JavaMail provides access from Java applications to IMAP- and SMTP-capable mail servers on your network or the Internet. It does not provide mail server functionality; so you must have access to a mail server to use JavaMail.

To configure JavaMail for use in WebLogic Server, you create a Mail Session in the WebLogic Server Administration Console. This allows server-side components and applications to access JavaMail services with JNDI, using Session properties you preconfigure for them. For example, by creating a Mail Session, you can designate the mail hosts, transport and store protocols, and the default mail user in the Administration Console so that components that use JavaMail do not have to set these properties. Applications that are heavy email users benefit because WebLogic Server creates a single Session object and makes it available via JNDI to any component that needs it.

Tasks

Configuring a New Mail Session

To configure a new mail session, proceed as follows:

1. Start the WebLogic Server Administration Console.

2. In the left pane of the Console, select Services > Mail.
3. In the right pane of the Console, click the Configure a New Mail Session option. A dialog displays in the right pane showing the tabs associated with configuring a new mail session.
4. Using the available tabs, enter the following information:
 - Configuration—Enter values in the Name, JNDI Name, and Properties attribute fields.
 - Targets—Indicate the Targets-Server for this configured mail session by moving the server from the Available list to the Chosen list.
 - Notes—Enter notes related to the mail session.
5. Click Apply 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.

Cloning a Mail Session

1. Under Services, 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 (to the left of the Delete 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. Using the available tabs, enter the following information:
 - Configuration—Enter values in the Name, JNDI Name, and Properties attribute fields.
 - Targets—Indicate the Targets-Server for this configured mail session by moving the server from the Available list to the Chosen list.
 - Notes—Enter notes related to the mail session.
4. Click Clone 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.

Deleting a Mail Session

1. Under Services, 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.

Assigning a Mail Session

1. Click the instance node in the left pane 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 Target and Deploy tab.
3. Complete the following steps for the Target and Deploy tab:
 - a. Select one or more targets to which you wish to assign mail.
 - b. Click Apply to save your assignments.

Attributes and Console Screen Reference for Mail Sessions

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“Mail Session” on page 273-1](#)

[“Mail Session --> Configuration” on page 270-1](#)

[“Mail Session --> Notes” on page 271-1](#)

[“Mail Session --> Target and Deploy” on page 272-1](#)



General Bridge Destination --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines configuration attributes for a general bridge destination for non-JMS messaging providers. Required attributes for this tab include the name of the general bridge destination within the domain, the name of the adapter used to communicate with the specified destination, the adapter `CLASSPATH`, and a list of properties to pass to the adapter.

Optionally, you can also specify a user name and password for each target and source general bridge destination

Tasks

[“Configuring General Bridge Destinations” on page 289-9](#)

[“Configuring a Messaging Bridge Instance” on page 289-11](#)

Related Topics

[“Simple Access to Remote or Foreign JMS Providers” on page 232-46](#)

Attributes

Table 276-1

Attribute Label	Description	Value Constraints
Name	<p>A general bridge destination name for the actual JMS destination being mapped to the bridge. This name must be unique across a WebLogic domain.</p> <p>For example, if you are bridging between WebLogic Server releases 6.1 and 8.1, for the source destination you could change the default destination name to “61to81SourceDestination”. Then when you create the corresponding target destination, you could name it as “61to81TargetDestination”. Once the bridge destinations are configured, these names are listed as options in the Source Destination and Target Destination attributes on the Bridges →General tab.</p> <p><i>MBean:</i> weblogic.management.configuration.BridgeDestinationMBean</p> <p><i>Attribute:</i> Name</p>	

Table 276-1

Attribute Label	Description	Value Constraints
Adapter JNDI Name	<p>The JNDI name of the resource adapter used to communicate with the general bridge destination. This name is specified in the adapter's deployment descriptor file and is used by the WebLogic Server Connector container to bind the adapter in WebLogic Server JNDI.</p> <p>Note: If the adapter isn't deployed, use the Deployments -> Connector Modules node to deploy it.</p> <p><i>MBean:</i> weblogic.management.configuration.BridgeDestinationMBean</p> <p><i>Attribute:</i> AdapterJNDIName</p>	<p><i>Default:</i> JMS_XA_ADAPTER_JNDI</p>
Adapter Classpath	<p>The CLASSPATH of the bridge destination. This is used mainly to connect to another release of WebLogic Server.</p> <p>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.</p> <p>Note: When connecting to a third-party JMS product, the bridge destination must supply the product's CLASSPATH in the WebLogic Server CLASSPATH.</p> <p><i>MBean:</i> weblogic.management.configuration.BridgeDestinationMBean</p> <p><i>Attribute:</i> Classpath</p>	

Table 276-1

Attribute Label	Description	Value Constraints
Properties (key=value)	<p>Specifies all the properties of the bridge destination. The destination properties are string values that must be separated by a semicolon (;).</p> <p>The following properties are required for all JMS implementations:</p> <p><code>ConnectionURL=</code> The URL used to establish a connection to the destination.</p> <p><code>ConnectionFactoryJNDIName=</code> The JMS connection factory used to create a connection for the actual destination being mapped to the general bridge destination.</p> <p><code>DestinationJNDIName=</code> The JNDI name of the actual destination being mapped to the general bridge destination.</p> <p><code>DestinationType=</code> Specify whether the destination type is either a Queue or Topic.</p> <p><code>InitialContextFactory=</code> The factory used to get the JNDI context.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.BridgeDestinationMBean</code></p> <p><i>Attribute:</i> Properties</p>	

Table 276-1

Attribute Label	Description	Value Constraints
User Name	<p>The optional user name that the adapter will use to access the bridge destination.</p> <p>Note: All operations done to the specified destination are done using this user name and the corresponding password. Therefore, the User Name/Password for the source and target destinations must have permission to the access the underlying destinations in order for the messaging bridge to work.</p> <p><i>MBean:</i> weblogic.management.configuration.BridgeDestinationMBean</p> <p><i>Attribute:</i> UserName</p>	
User Password	<p>The user password that the adapter uses to access the bridge destination.</p> <p><i>MBean:</i> weblogic.management.configuration.BridgeDestinationMBean</p> <p><i>Attribute:</i> UserPassword</p>	<i>Encrypted:</i> yes



General Bridge Destination --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to supply optional information about this general (non-JMS) bridge configuration.

Tasks

[“Configuring General Bridge Destinations” on page 289-9](#)

Related Topics

Attributes

Table 277-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.BridgeDestinationMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes



General Bridge Destination

Each WebLogic Messaging Bridge consists of two destinations that are being bridged: a source destination that the bridge reads messages from and a target destination where the bridge sends the messages that it receives from the source destination. For each non-JMS source and target destination to be mapped to a messaging bridge, you must configure a General Bridge Destination instance. However, a custom adapter must be provided by a third-party OEM vendor or from BEA Professional Services to access non-JMS source or target destinations.

Note: An easier alternative for configuring bridge destinations for WebLogic JMS and third-party JMS products is to create JMS Bridge Destination instances.

To create a new non-JMS bridge destination, click the [Configure a new General Bridge Destination](#) link.

- For more information on creating a general bridge destination, see [“Configuring General Bridge Destinations” on page 289-9](#).
- For information on creating JMS bridge destinations for WebLogic JMS and third-party JMS products, see [“Configuring JMS Bridge Destinations” on page 289-7](#).



Messaging Bridge --> Configuration --> Connection Retry

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines connection retry configuration attributes for a messaging bridge. The source and target destinations for a messaging bridge will not always be available. Therefore, the messaging bridge must be able to reconnect to the destination at some periodic interval. These attributes govern the time between reconnection attempts.

Tasks

[“Configuring a Messaging Bridge Instance” on page 289-11](#)

[“Targeting a Messaging Bridge to a Server, a Cluster, or a Migratable Target” on page 289-16](#)

Related Topics

[“Simple Access to Remote or Foreign JMS Providers” on page 232-46](#)

Attributes

Table 279-1

Attribute Label	Description	Value Constraints
Minimum Delay (seconds)	<p>The minimum amount of time, in seconds, that the messaging bridge will wait before it tries to reconnect to the source or target destination after a failure.</p> <p>This attribute works with the <code>Maximum Delay</code> and <code>Incremental Delay</code> attributes. After the first failure to connect to a destination, the bridge will wait for the number of seconds defined by <code>Minimum Delay</code>.</p> <p>If the second trial also fails, it will increase its waiting time by the number of seconds defined by <code>Incremental Delay</code>. The maximum delay time is defined by <code>Maximum Delay</code>. Once the waiting time is increased to the maximum value, the bridge will not increase its waiting time anymore.</p> <p>Once the bridge successfully connects to the destination, its waiting time will be reset to the initial value defined by <code>Minimum Delay</code>.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.MessagingBridgeMBean</code></p> <p><i>Attribute:</i> <code>ReconnectDelayMinimum</code></p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 15</p> <p><i>Dynamic:</i> yes</p>

Table 279-1

Attribute Label	Description	Value Constraints
Incremental Delay (seconds)	<p>The incremental delay time, in seconds, that the messaging bridge will wait longer between one failed reconnection attempt and the next retry.</p> <p>This attribute works with the Minimum Delay and Maximum Delay attributes. After the first failure to connect to a destination, the bridge will wait for the number of seconds defined by Minimum Delay.</p> <p>Each time a reconnect attempt fails, the bridge will increase its waiting time by the number of seconds defined by Incremental Delay. The maximum delay time is defined by Maximum Delay. Once the waiting time is increased to the maximum value, the bridge will not increase its waiting time anymore.</p> <p>Once the bridge successfully connects to the destination, its waiting time will be reset to the minimum value defined by Minimum Delay.</p> <p><i>MBean:</i> weblogic.management.configuration.MessagingBridgeMBean</p> <p><i>Attribute:</i> ReconnectDelayIncrease</p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 5</p> <p><i>Dynamic:</i> yes</p>

Table 279-1

Attribute Label	Description	Value Constraints
Maximum Delay (seconds)	<p>The longest time, in seconds, that the messaging bridge will wait between one failed attempt to reconnect to the source or target and the next retry.</p> <p>This attribute works with the Minimum Delay and Incremental Delay attributes. After the first failure to connect to a destination, the bridge will wait for the number of seconds defined by Minimum Delay.</p> <p>Each time a reconnect attempt fails, the bridge will increase its waiting time by the number of seconds defined by Incremental Delay. The maximum delay time is defined by Maximum Delay. Once the waiting time is increased to the maximum value, the bridge will not increase its waiting time anymore.</p> <p>Once the bridge successfully connects to the destination, its waiting time will be reset to the initial value defined by Minimum Delay.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.MessagingBridgeMBean</code></p> <p><i>Attribute:</i> <code>ReconnectDelayMaximum</code></p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 60</p> <p><i>Dynamic:</i> yes</p>

Messaging Bridge --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

A messaging bridge communicates with the configured source and target destinations. For each mapping of a source destination to a target destination, whether it is another WebLogic JMS implementation, a third-party JMS provider, or another non-JMS messaging product, you must configure a messaging bridge instance.

This page allows you to define general configuration attributes for a messaging bridge instance, including the source and target destination for the mapping, a message filtering selector, and a quality of service.

Tasks

[“Configuring a Messaging Bridge Instance” on page 289-11](#)

[“Targeting a Messaging Bridge to a Server, a Cluster, or a Migratable Target” on page 289-16](#)

Related Topics

[“Monitoring All Messaging Bridges” on page 289-23](#)

[“Simple Access to Remote or Foreign JMS Providers” on page 232-46](#)

[“Using Message-Driven Beans”](#) in *Programming WebLogic Enterprise JavaBeans*

MDBTransaction interface in the [weblogic.jms.extensions](#)

Attributes

Table 280-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this messaging bridge configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.MessagingBridgeMBean</code></p> <p><i>Attribute:</i> <code>Name</code></p>	
Source Destination	<p>The source bridge destination <i>from which</i> messages are received by the messaging bridge.</p> <p>This must be an instance of either the JMS Bridge Destination (<code>JMSBridgeDestinationMBean</code>) or the General Bridge Destination (<code>BridgeDestinationMBean</code>), which are used to define the source destination that the messaging bridge will read messages from.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.MessagingBridgeMBean</code></p> <p><i>Attribute:</i> <code>SourceDestination</code></p>	<p><i>Default:</i> <code>null</code></p>

Table 280-1

Attribute Label	Description	Value Constraints
Target Destination	<p>The target bridge destination <i>to which</i> messages are sent from the messaging bridge.</p> <p>This must be an instance of either the JMS Bridge Destination (JMSBridgeDestinationMBean) or the General Bridge Destination (BridgeDestinationMBean), which are used to define the target destination that the messaging bridge will send the messages it receives from the source destination.</p> <p><i>MBean:</i> weblogic.management.configuration.MessagingBridgeMBean</p> <p><i>Attribute:</i> TargetDestination</p>	<i>Default:</i> null
Selector	<p>A message selector for filtering the messages that are sent across the messaging bridge.</p> <p>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.</p> <p><i>MBean:</i> weblogic.management.configuration.MessagingBridgeMBean</p> <p><i>Attribute:</i> Selector</p>	

Table 280-1

Attribute Label	Description	Value Constraints
Quality Of Service	<p>The quality-of-service (QOS) guarantee for forwarding messages across a messaging bridge.</p> <p><i>Exactly-once</i> - Each message in the source destination will be transferred to the target exactly once. This is the highest QOS a bridge can offer. In order to use this QOS:</p> <ul style="list-style-type: none"> ■ Any WebLogic implementation must be release 6.1 or later. ■ The source and target JMS connection factories must be configured to use the XAConnectionFactory. ■ The transaction <code>jms-xa-adp.rar</code> adapter must be deployed and identified in the Adapter JNDI Name attribute as <code>"eis.jms.WLSConnectionFactoryJNDIXA"</code> for both the source and target destinations. <p><i>Atmost-once</i> - Each message is sent at most one time. Some messages may not be delivered to the target destination.</p> <p><i>Duplicate-okay</i> - Each message is sent at least one time. Duplicate messages can be delivered to the target destination.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.MessagingBridgeMBean</code></p> <p><i>Attribute:</i> <code>QualityOfService</code></p>	<p><i>Default:</i> Exactly-once</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ Exactly-once ■ Atmost-once ■ Duplicate-okay

Table 280-1

Attribute Label	Description	Value Constraints
QOS Degradation Allowed	<p>Indicates whether the messaging bridge automatically degrades the requested QOS (quality-of-service) when the configured one is not available.</p> <p>If this occurs, a message is delivered to the WebLogic startup window (or log file). If QOS Degradation Allowed is not selected, and the messaging bridge cannot satisfy the requested QOS, it will result in an error and the messaging bridge will not start.</p> <p><i>MBean:</i> weblogic.management.configuration.MessagingBridgeMBean</p> <p><i>Attribute:</i> QOSDegradationAllowed</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false
Maximum Idle Time (seconds)	<p>The maximum number of seconds the messaging bridge will remain idle.</p> <p>For bridges running in <i>asynchronous mode</i>, the maximum idle time defines the longest time the bridge will stay idle before it checks the health of its connections.</p> <p>For bridges running in <i>synchronous mode</i>, the maximum idle time defines the amount of time the bridge can block on a receive call if no transaction is involved.</p> <p><i>MBean:</i> weblogic.management.configuration.MessagingBridgeMBean</p> <p><i>Attribute:</i> IdleTimeMaximum</p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> 60</p> <p><i>Dynamic:</i> yes</p>

Table 280-1

Attribute Label	Description	Value Constraints
Asynchronous Mode Enabled	<p>Indicate whether a messaging bridge works in asynchronous mode. Messaging bridges that work in asynchronous mode are driven by the source destination. The messaging bridge listens for messages and forwards them as they arrive. When Asynchronous Mode is disabled, the bridge works in synchronous mode, even if the source supports asynchronous receiving.</p> <p>Note: For a messaging bridge with a QOS of <i>Exactly-once</i> to work in asynchronous mode, the source destination has to support the <code>MDBTransaction</code> interface described in the <code>weblogic.jms.extensions</code> Javadoc. Otherwise, the bridge automatically switches to synchronous mode if it detects that <code>MDBTransactions</code> are not supported by the source destination. For more information about <code>MDBTransactions</code>, see “Using Message-Driven Beans” in <i>Programming WebLogic Enterprise JavaBeans</i>. (See the links in the Related Topics section.)</p> <p><i>MBean:</i> <code>weblogic.management.configuration.MessagingBridgeMBean</code></p> <p><i>Attribute:</i> <code>AsyncEnabled</code></p>	<p><i>Default:</i> true</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false

Table 280-1

Attribute Label	Description	Value Constraints
Durability Enabled	<p>Durability is specified only for JMS topics or for third-party destinations with similar characteristics as a JMS topic. This attribute is ignored if the source destination is a JMS queue.</p> <p>By enabling durability, a messaging bridge creates a durable subscription for the source destination. 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 bridge destination once it is restarted. If this attribute is not selected, messages that are sent to the source JMS topic while the bridge is down cannot be forwarded to the target destination.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.MessagingBridgeMBean</code></p> <p><i>Attribute:</i> <code>DurabilityEnabled</code></p>	<p><i>Default:</i> <code>true</code></p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ <code>true</code>■ <code>false</code>

Table 280-1

Attribute Label	Description	Value Constraints
Started	<p>Indicates the initial state of the messaging bridge when it is configured and whenever the server is restarted. You can also use this field to dynamically start and stop the messaging bridge. To stop the bridge, clear the check box. Conversely, reselect the check box to restart the bridge.</p> <p>Note: Unless there is a configuration issue that prevents the messaging bridge from starting, this field indicates the expected run-time state of the messaging bridge. For information on monitoring all the configured messaging bridges in your domain, see the "Monitoring All Messaging Bridges" link in the Related Topics section.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.MessagingBridgeMBean</code></p> <p><i>Attribute:</i> <code>Started</code></p>	<p><i>Default:</i> true</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false <p><i>Dynamic:</i> yes</p>

Messaging Bridge --> Configuration --> Transactions

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines the transaction attributes for a messaging bridge, including setting transaction timeout intervals, transaction batch sizes, and batch intervals.

Tasks

[“Configuring a Messaging Bridge Instance” on page 289-11](#)

[“Targeting a Messaging Bridge to a Server, a Cluster, or a Migratable Target” on page 289-16](#)

Related Topics

[“Simple Access to Remote or Foreign JMS Providers” on page 232-46](#)

Attributes

Table 281-1

Attribute Label	Description	Value Constraints
Transaction Timeout	<p>The amount of time, in seconds, that the transaction manager will wait for each transaction before timing it out.</p> <p>Transaction timeouts are used when the QOS for a bridge requires transactions. If a bridge is configured with the <i>Exactly-once</i> QOS, the receiving and sending is completed in one transaction.</p> <p><i>MBean:</i> weblogic.management.configuration.MessagingBridgeMBean</p> <p><i>Attribute:</i> TransactionTimeout</p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 30</p> <p><i>Dynamic:</i> yes</p>
Batch Size	<p>The number of messages that are processed within one transaction.</p> <p><i>Note:</i> This attribute only applies to bridges that work in synchronous mode and whose QOS require two-phase transactions.</p> <p><i>MBean:</i> weblogic.management.configuration.MessagingBridgeMBean</p> <p><i>Attribute:</i> BatchSize</p>	<p><i>Minimum:</i> 0</p> <p><i>Default:</i> 10</p> <p><i>Dynamic:</i> yes</p>

Table 281-1

Attribute Label	Description	Value Constraints
Batch Interval (milliseconds)	<p>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.</p> <p>This attribute only applies to bridges that work in synchronous mode and whose QOS require two-phase transactions.</p> <p>The default value of -1 indicates that the messaging bridge should wait until the number of messages reaches the Batch Size amount before it completes a transaction. However, internally the message batch will be timed out before the transaction times out, and the transaction will be committed regardless of whether the Batch Size amount has been reached or not.</p> <p><i>Caution:</i> This value should not be set higher than the Transaction Timeout value. Otherwise, if there is not enough volume to make a batch of messages within the time specified by the Batch Interval value, all pending batched messages may be rolled back and resent. In some cases, this may result in an infinite loop.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.MessagingBridgeMBean</code></p> <p><i>Attribute:</i> <code>BatchInterval</code></p>	<p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>



Messaging Bridge --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to supply optional information about this messaging bridge configuration.

Tasks

[“Configuring a Messaging Bridge Instance” on page 289-11](#)

Related Topics

[“Simple Access to Remote or Foreign JMS Providers” on page 232-46](#)

Attributes

Table 282-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.MessagingBridgeMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes



Messaging Bridge --> Target and Deploy

[Tasks](#) [Related Topics](#)

Overview

This tab allows you to target the messaging bridge to independent servers, server clusters, or migratable targets in the domain.

Note: Migratable targets define a set of WebLogic Server instances in a cluster that can potentially host JMS as an “exactly-once” service. Unlike applications, exactly-once services are not active on all server instances in are cluster. They are instead "pinned" to a single server in the cluster to preserve data consistency.

Tasks

[“Configuring a Messaging Bridge Instance” on page 289-11](#)

[“Targeting a Messaging Bridge to a Server, a Cluster, or a Migratable Target” on page 289-16](#)

Related Topics

[“Simple Access to Remote or Foreign JMS Providers” on page 232-46](#)



Messaging Bridge

A messaging bridge communicates with configured source and target bridge destinations using the resource adapters provided with WebLogic Server. For each mapping of a source destination to a target destination, whether it is another WebLogic JMS implementation or a third-party JMS product, you must configure a Messaging Bridge instance. Each instance defines the source and target destination for the mapping, a message filtering selector, a quality of service (QOS), transaction semantics, and reconnection parameters.

Note: For non-JMS messaging products, a customized adapter must be provided by a third-party OEM vendor or BEA Professional Services to access non-JMS source or target destinations.

To create a new messaging bridge, click the [Configure a new Messaging Bridge](#) link.

- For more information about creating a messaging bridge, see [“Configuring a Messaging Bridge Instance” on page 289-11](#).
- For more information on deleting a messaging bridge, see [“Targeting a Messaging Bridge to a Server, a Cluster, or a Migratable Target” on page 289-16](#).



Messaging Bridge Runtime

This page enables you to monitor the status of all the WebLogic messaging bridges in your domain. A messaging bridge communicates with configured source and target bridge destinations using the resource adapters provided with WebLogic Server.

- For more information about creating a messaging bridge, see [“Messaging Bridge Configuration Tasks” on page 289-3](#).



JMS Bridge Destination --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to supply optional information about this JMS messaging bridge destination configuration.

Tasks

[“Configuring JMS Bridge Destinations” on page 289-7](#)

Related Topics

Attributes

Table 286-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.JMSBridgeDestinationMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes



JMS Bridge Destination

Each WebLogic Messaging Bridge consists of two destinations that are being bridged: a source destination that the bridge reads messages from and a target destination where the bridge sends the messages that it receives from the source destination. For each source and target destination to be mapped to a messaging bridge, whether it's a WebLogic JMS implementation or a third-party JMS product, you must configure a JMS Bridge Destination instance.

Note: For non-JMS messaging products, you must configure a General Bridge Destination instance for each source and target destination to be mapped by a messaging bridge.

To create a new JMS bridge destination, click the [Configure a new JMS Bridge Destination](#) link.

- For more information on creating a bridge destination, see [“Configuring JMS Bridge Destinations” on page 289-7](#).
- For information on creating bridge destinations for non-JMS messaging products, see [“Configuring General Bridge Destinations” on page 289-9](#).



JMS Bridge Destination --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines the configuration attributes for a JMS Bridge Destination—either for WebLogic JMS or a third-party JMS product. Attributes on this tab include the name of the JMS bridge destination within the domain, the JNDI name of the adapter used to communicate with the specified destination, the adapter `CLASSPATH`, properties to pass to the adapter, and the destination type (queue or topic).

Optionally, you can also specify a user name and password for each target and source JMS bridge destination.

Tasks

[“Configuring JMS Bridge Destinations” on page 289-7](#)

Related Topics

[“Simple Access to Remote or Foreign JMS Providers” on page 232-46](#)

Attributes

Table 288-1

Attribute Label	Description	Value Constraints
Name	<p>A JMS bridge destination name for the actual JMS destination being mapped to the bridge. This name must be unique across a WebLogic domain.</p> <p>For example, if you are bridging between WebLogic Server releases 6.1 and 8.1, for the source destination you could change the default bridge destination name to “61to81SourceDestination”. Then, when you create the corresponding target destination, you could name it as “61to81TargetDestination”. Once the bridge destinations are configured, these names are listed as options in the Source Destination and Target Destination attributes on the Bridges →General tab.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSBridgeDestinationMBean</p> <p><i>Attribute:</i> Name</p>	

Table 288-1

Attribute Label	Description	Value Constraints
Adapter JNDI Name	<p>The JNDI name of the resource adapter used to communicate with the JMS bridge destination. This name is specified in the adapter's deployment descriptor file and is used by the WebLogic Server Connector container to bind the adapter in WebLogic Server JNDI.</p> <p>Note: If the adapter isn't deployed, use the Deployments -> Connector Modules node to deploy it.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSBridgeDestinationMBean</p> <p><i>Attribute:</i> AdapterJNDIName</p>	<p><i>Default:</i> JMS_XA_ADAPTER_JNDI</p>
Adapter Classpath	<p>The CLASSPATH of the JMS bridge destination. This is used mainly to connect to another release of WebLogic Server.</p> <p>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.</p> <p>Note: When connecting to a third-party JMS product, the bridge destination must supply the product's CLASSPATH in the WebLogic Server CLASSPATH.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSBridgeDestinationMBean</p> <p><i>Attribute:</i> Classpath</p>	

Table 288-1

Attribute Label	Description	Value Constraints
Connection URL	The connection URL for a JMS bridge destination. <i>MBean:</i> weblogic.management.configuration.JMSBridgeDestinationMBean <i>Attribute:</i> ConnectionURL	
Initial Context Factory	The initial context factory name for a JMS bridge destination. <i>MBean:</i> weblogic.management.configuration.JMSBridgeDestinationMBean <i>Attribute:</i> InitialContextFactory	<i>Default:</i> weblogic.jndi.WLInitialContextFactory
Connection Factory JNDI Name	The JMS connection factory used to create a connection for the actual JMS destination being mapped to the JMS bridge destination. Note: In order to use the Exactly-once quality of service for transactions, the connection factory has to be a XAConnection Factory. <i>MBean:</i> weblogic.management.configuration.JMSBridgeDestinationMBean <i>Attribute:</i> ConnectionFactoryJNDIName	<i>Default:</i> null
Destination JNDI Name	The JNDI name of the actual JMS destination being mapped to the JMS bridge destination. <i>MBean:</i> weblogic.management.configuration.JMSBridgeDestinationMBean <i>Attribute:</i> DestinationJNDIName	<i>Default:</i> null

Table 288-1

Attribute Label	Description	Value Constraints
Destination Type	<p>The destination type (queue or topic) for a JMS bridge destination.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSBridgeDestinationMBean</p> <p><i>Attribute:</i> DestinationType</p>	<p><i>Default:</i> Queue</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ Queue■ Topic
User Name	<p>The optional user name that the adapter will use to access the bridge destination.</p> <p>Note: All operations done to the specified destination are done using this user name and the corresponding 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.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSBridgeDestinationMBean</p> <p><i>Attribute:</i> UserName</p>	
User Password	<p>The user password that the adapter uses to access the bridge destination.</p> <p><i>MBean:</i> weblogic.management.configuration.JMSBridgeDestinationMBean</p> <p><i>Attribute:</i> UserPassword</p>	<p><i>Encrypted:</i> yes</p>



1 Messaging Bridge

[“Attributes and Console Screen Reference for Messaging Bridge” on page 290-1]

The following sections explain how to configure and manage a WebLogic Messaging Bridge:

- [“What Is a Messaging Bridge?” on page 289-2](#)
- [“Messaging Bridge Configuration Tasks” on page 289-3](#)
- [“Using the Messaging Bridge to Interoperate with Different WebLogic Server Releases and Domains” on page 289-17](#)
- [“Using the Messaging Bridge to Access a Third-Party Messaging Provider” on page 289-22](#)
- [“Managing a Messaging Bridge” on page 289-23](#)

You may also want to refer these WebLogic JMS sections:

- [“JMS: Configuring” on page 232-1](#)
- [“JMS: Tuning” on page 235-1](#)
- [“JMS: Monitoring” on page 234-1](#)

What Is a Messaging Bridge?

The WebLogic Messaging Bridge allows you to configure a forwarding mechanism between any two messaging products—thereby, providing interoperability between separate implementations of WebLogic JMS, or between WebLogic JMS and another messaging product. You can use the Messaging Bridge to integrate your messaging applications between:

- Any two implementations of WebLogic JMS, including those from separate releases of WebLogic Server.
- WebLogic JMS implementations that reside in separate WebLogic domains.
- WebLogic JMS with a third-party JMS product (for example, MQSeries).
- WebLogic JMS with non-JMS messaging products (only by using specialized connector adapters that are not provided with WebLogic Server).

A messaging bridge consists of two destinations that are being bridged: a source destination that the bridge reads messages from and a target destination where the bridge sends the messages that it receives from the source destination. For WebLogic JMS and third-party JMS products, a messaging bridge communicates with source and target destinations using the resource adapters provided with WebLogic Server. For non-JMS messaging products, a custom connector adapter must be obtained from a third-party OEM vendor or by contacting BEA Professional Services in order to access non-JMS source or target destinations.

Source and target bridge destinations can be either queues or topics. You can also specify a quality of service (QOS), as well as message filters, transaction semantics, and connection retry policies. Once a messaging bridge is configured, it is easily managed from the Administration Console, including temporarily suspending bridge traffic whenever necessary, tuning the execute thread pool size to suit your implementation, and monitoring the status of all your configured bridges.

Messaging Bridge Configuration Tasks

Before you can deploy a messaging bridge, you need to configure its required components:

- “About the Bridge’s Resource Adapters” on page 289-3
- “Deploying the Bridge’s Resource Adapters” on page 289-5
- “Configuring Source and Target Bridge Destinations” on page 289-6
- “Configuring a Messaging Bridge Instance” on page 289-11

About the Bridge’s Resource Adapters

A messaging bridge uses resource adapters to communicate with the configured source and target JMS destinations. You need to associate both the source and target JMS destinations with a *supported* resource adapter in order for the bridge to communicate with them. The JNDI name for the adapter is configured as part of the resource adapter’s deployment descriptor.

Note: Although WebLogic JMS includes a “General Bridge Destination” framework for accessing non-JMS messaging products, WebLogic Server does not provide supported connector adapters for such products. Therefore, you must obtain a custom connector adapter from a third-party OEM vendor and consult their documentation for configuration instructions. You can also contact BEA Professional Services for information about obtaining a custom connector adapter.

The supported resource adapters are located in the `WL_HOME\server\lib` directory and are described in the following table.

Table 289-1 Messaging Bridge Resource Adapters and JNDI Names

Adapter	JNDI Name	Description
<code>jms-xa-adp.rar</code>	<code>eis.jms.WLSConnectionFactoryJNDIXA</code>	<p>Provides transaction semantics via the <code>XAResource</code>. Used when the required QOS is <i>Exactly-once</i>. This envelops a received message and sends it within a user transaction (XA/JTA). The following requirements are necessary in order to use this resource adapter:</p> <ul style="list-style-type: none"> ■ Any WebLogic Server implementation being bridged must be release 6.1 or later. ■ The source and target JMS connection factories must be configured to use the <code>XAConnectionFactory</code>. <p>Note: Before deploying this resource adapter, refer to the “Using the Messaging Bridge to Interoperate with Different WebLogic Server Releases and Domains” on page 289-17 for specific transactional configuration requirements and guidelines.</p>
<code>jms-notran-adp.rar</code>	<code>eis.jms.WLSConnectionFactoryJNDINoTX</code>	<p>Provides no transaction semantics. Used when the required QOS is <i>Atmost-once</i> or <i>Duplicate-okay</i>. If the requested QOS is <i>Atmost-once</i>, the resource adapter uses the <code>AUTO_ACKNOWLEDGE</code> mode. If the requested QOS is <i>Duplicate-okay</i>, <code>CLIENT_ACKNOWLEDGE</code> is used.</p> <p>Note: For more information about the acknowledge modes used in non-transacted sessions, see “WebLogic JMS Fundamentals” in <i>Programming WebLogic JMS</i>.</p>

Table 289-1 Messaging Bridge Resource Adapters and JNDI Names

Adapter	JNDI Name	Description
jms-notran-adp51.rar	eis.jms.WLS51ConnectionFactoryJNDINoTX	Provides interoperability with release 5.1 when either the source or target destination is a 5.1 server instance. This resource adapter provides no transaction semantics; therefore, it only supports a QOS of <i>Atmost-once</i> or <i>Duplicate-okay</i> . If the requested QOS is <i>Atmost-once</i> , the resource adapter uses the AUTO_ACKNOWLEDGE mode. If the requested QOS is <i>Duplicate-okay</i> , CLIENT_ACKNOWLEDGE is used.

You will specify the appropriate resource adapter by its JNDI name when you configure each source and target bridge destination.

Deploying the Bridge's Resource Adapters

Before you configure the messaging bridge destinations, deploy the appropriate resource adapters in the WebLogic Server domain that is hosting the messaging bridge, as follows:

1. Select the domain in which you will deploy the adapters (for example, Examples).
2. Select the Deployments →Connector Modules option to open the Resource Connectors page.
3. Follow the deployment assistant instructions to deploy the appropriate resource adapter, as defined in [Table 289-1, “Messaging Bridge Resource Adapters and JNDI Names,” on page 1-4](#).
 - jms-xa-adp.rar
 - jms-notran-adp.rar
 - jms-notran-adp51.rar
4. You must target the resource adapter to the same WebLogic Server instance that the messaging bridge will be targeted to. For more information, see “Targeting a Messaging Bridge to a Server, a Cluster, or a Migratable Target” on page 289-16.

Note: When configuring a messaging bridge to interoperate between WebLogic Server release 8.1 and release 5.1, then the release 5.1 resource adapter (`jms-notran-adp51.rar`) and the non-transaction adapter (`jms-notran-adp.rar`) must be deployed on the 8.1 domain running the messaging bridge.

For more information on deploying resource adapters, see “[Packaging and Deploying Connectors](#)” in *Programming WebLogic Server J2EE Connectors*.

Configuring Source and Target Bridge Destinations

A messaging bridge connects two actual destinations that are mapped to bridge destinations: a source destination *from which* messages are received, and a target destination *to which* messages are sent. Depending on the messaging products that need to be bridged, there are two types of bridge destinations:

- **JMS Bridge Destination** – For JMS messaging products, whether it is a WebLogic JMS implementation or a third-party JMS provider, you need to configure a `JMSBridgeDestination` instance for each actual source and target JMS destination being mapped to a messaging bridge.
- **General Bridge Destination** – For non-JMS messaging products, you need to configure a generic `BridgeDestination` instance for each actual source and target destination being mapped to a messaging bridge.

Before starting the procedures in this section, refer to the “Using the Messaging Bridge to Interoperate with Different WebLogic Server Releases and Domains” on page 289-17 or “Using the Messaging Bridge to Access a Third-Party Messaging Provider” on page 289-22 sections for specific configuration requirements and guidelines.

Note: When configuring third-party JMS provider bridge destination, you can use the Foreign JMS Server feature to quickly configure multiple source or target destinations. For more information, see “Simple Access to Remote or Foreign JMS Providers” on page 232-48.

Configuring JMS Bridge Destinations

A `JMSBridgeDestination` instance defines a unique name for a bridge's source and target destinations within a WebLogic domain, the name of the adapter used to communicate with the specified destination, property information to pass to the adapter (Connection URL, Connection Factory JNDI Name, etc.), and, optionally, a user name and password.

You need to configure a `JMSBridgeDestination` instance for each actual source and target JMS destination to be mapped to a messaging bridge. Therefore, when you finish defining attributes for a source JMS bridge destination, repeat these steps to configure a target JMS bridge destination, or vice versa. You will designate the source and target JMS Bridge Destinations in “Configuring a Messaging Bridge Instance” on page 289-11.

To configure a source or target JMS bridge destination, follow these steps.

1. Expand the Services →Messaging Bridge node.
2. Click the JMS Bridge Destinations node to open the JMS Bridge Destinations page in the right pane.
3. Click the Configure a new JMS Bridge Destination link. A Configuration dialog shows the tabs associated with configuring a new JMS bridge destination.
4. On the Configuration General tab, define the general configuration attributes for a JMS bridge destination:
 - Enter a JMS bridge destination name for the actual JMS destination being mapped to the bridge. This name must be unique across a WebLogic domain. For example, if you are bridging between WebLogic Server releases 6.1 and 8.1, for the source destination you could change the default bridge destination name to “61SourceDestination”. Then, when you create the corresponding target destination, you could name it as “81TargetDestination”. Once the bridge destinations are configured, these names are listed as options in the Source Destination and Target Destination attributes on the Bridges →General tab.
 - Specify the JNDI name of the resource adapter used to communicate with the messaging bridge destinations:

`eis.jms.WLSConnectionFactoryJNDIXA` (default) — QOS is *Exactly-once*

`eis.jms.WLSConnectionFactoryJNDINoTX` — QOS is *Atmost-once* or *Duplicate-Okay*

`eis.jms.WLS51ConnectionFactoryJNDINoTX` — used only with release 5.1 and only supports QOS of *Atmost-once* or *Duplicate-Okay*

For more information on which resource adapter name to use, see “Messaging Bridge Resource Adapters and JNDI Names” on page 289-4.

- Leave the Adapter Classpath field blank when connecting to source and target destinations that are both running on WebLogic Server 6.1 or later. When connecting to either a source or target destination that is running on WebLogic Server 6.0 or earlier, the Adapter Classpath field must indicate the location of the classes for the earlier WebLogic Server release. When connecting to a third-party JMS provider, the bridge destination must supply the provider’s CLASSPATH in the WebLogic Server CLASSPATH.
- Specify the URL of the JNDI provider used to look up the connection factory and destination.
- Specify the context factory used to get the JNDI context.
- Specify the JMS connection factory used to create a connection for the actual JMS destination being mapped to the JMS bridge destination.

Note: In order to use the Exactly-once QOS for transactions, the JMS connection factory has to be an XA connection factory. For more information about connection factory and QOS requirements, refer to the Attributes table in “Messaging Bridge --> Configuration --> General” on page 280-1.

- Specify the JNDI name of the actual JMS destination being mapped to the JMS bridge destination.
- Indicate whether the destination type is either a Queue or Topic.
- Optionally, enter the user name and password that the messaging bridge will give to the bridge adapter.

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 JMS destinations in order for the messaging bridge to work.

For more information about JMS bridge destination attributes, see “JMS Bridge Destination --> Configuration” on page 288-1.

5. Click Create to create an instance of 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.

When you finish defining attributes for a source JMS bridge destination, repeat these steps to configure a target JMS bridge destination, or vice versa. Then follow the instructions for [“Configuring a Messaging Bridge Instance” on page 289-11](#).

Configuring General Bridge Destinations

A general `BridgeDestination` instance defines a unique name for the actual source and target general bridge destinations within the WebLogic domain, the name of the adapter used to communicate with the specified destination, a list of properties to pass to the adapter, and, optionally, a user name and password.

Note: Although WebLogic JMS includes a “General Bridge Destination” framework for accessing non-JMS messaging products, WebLogic Server does not provide supported connector adapters for such products. Therefore, you must obtain a custom connector adapter from a third-party OEM vendor and consult their documentation for configuration instructions. You can also contact BEA Professional Services for information about obtaining a custom connector adapter.

You need to configure a `BridgeDestination` instance for each actual source and target destination to be mapped to a messaging bridge. Therefore, when you finish defining attributes for a source general bridge destination, repeat these steps to configure a target general bridge destination, or vice versa. You will designate the source and target general Bridge Destinations in “Configuring a Messaging Bridge Instance” on page 289-11.

To configure a source or target general bridge destination, follow these steps.

1. Expand the Services → Messaging Bridge node.
2. Click the General Bridge Destinations node to open the General Bridge Destinations page in the right pane.
3. 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, define the general configuration attributes for a general bridge destination:
 - Enter a general bridge destination name that is unique across a WebLogic Server domain. For example, if you are bridging between WebLogic Server releases 6.1 and 8.1, for the source destination you could change the default destination name to “61SourceDestination”. Then when you create the corresponding target destination, you could name it as “81TargetDestination”. Once the bridge destinations are configured, these names are listed as options in the Source Destination and Target Destination attributes on the Bridges → General tab.
 - Specify the JNDI name of the adapter used to communicate with the bridge destinations.

`eis.jms.WLSConnectionFactoryJNDIXA` (default) — QOS is *Exactly-once*

`eis.jms.WLSConnectionFactoryJNDINoTX` — QOS is *Atmost-once* or *Duplicate-Okay*

`eis.jms.WLS51ConnectionFactoryJNDINoTX` — used only with release 5.1 and only supports QOS of *Atmost-once* or *Duplicate-Okay*

For more information on which resource adapter name to use, see “Messaging Bridge Resource Adapters and JNDI Names” on page 289-4.

Note: 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 to obtain a custom adapter.

- Leave the Adapter Classpath field blank when connecting to source and target destinations that are both running on WebLogic Server 6.1 or later. When connecting to either a source or target destination that is running on WebLogic Server 6.0 or earlier, the Adapter Classpath field must indicate the location of the classes for the earlier WebLogic Server release. When connecting to a third-party JMS provider, the bridge destination must supply the provider’s CLASSPATH in the WebLogic Server CLASSPATH.
- Specify all the properties defined for a bridge destination. Each property must be separated by a semicolon (for example, `DestinationJNDIName=myTopic;DestinationType=topic;`). For a complete listing of the required properties for JMS implementations, refer to the Attributes table in “General Bridge Destination --> Configuration” on page 276-1.

Note: 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.

- Optionally, enter the user name and password that the messaging bridge will give to the bridge adapter.

Note: The User Name/Password for the source and target destinations must have permission to access the underlying source and target destinations in order for the Messaging Bridge to work.

For more information about the general bridge destination attributes, see “General Bridge Destination --> Configuration” on page 276-1.

5. Click Create to create an instance of 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.

When you finish defining attributes for a source general bridge destination, repeat these steps to configure a target general bridge destination, or vice versa. Then follow the instructions for “[Configuring a Messaging Bridge Instance](#)” on page 289-11.

Configuring a Messaging Bridge Instance

A messaging bridge instance communicates with the configured source and target bridge destinations. For each mapping of a source destination to a target destination, whether it is another WebLogic JMS implementation, a third-party JMS provider, or another non-JMS messaging product, you must configure a `MessagingBridge` instance. Each `MessagingBridge` instance defines the source and target destination for the mapping, a message filtering selector, a QOS, transaction semantics, and various reconnection parameters.

Before starting the procedure in this section, refer to the “Using the Messaging Bridge to Interoperate with Different WebLogic Server Releases and Domains” on page 289-17 or “Using the Messaging Bridge to Access a Third-Party Messaging Provider” on page 289-22 sections for specific configuration requirements and guidelines.

To configure a messaging bridge, follow these steps:

1. Expand the Services → Messaging Bridge node.
2. Expand the Bridges node to open the Messaging Bridges tab in the right pane.

3. Click the Configure a new Messaging Bridge link. A dialog shows the tabs associated with configuring a new messaging bridge.
4. On the Configuration General tab, define the general configuration attributes for a messaging bridge:

Table 289-2 Messaging Bridge Attributes on the General Tab

Attribute	Description
Name	Enter a messaging bridge name that is unique across a WebLogic Server domain.
Source Destination	Select the source destination <i>from which</i> messages are received by the messaging bridge. Following the example target JMS bridge destination name that was previously suggested for connecting WebLogic Server releases 6.1 and 8.1, you would select the “61SourceDestination” name that you created on the JMS Bridge Destination -> Configuration tab.
Target Destination	Select the target destination <i>to which</i> messages are sent from the messaging bridge. Following the example target JMS bridge destination name that was previously suggested for connecting WebLogic Server releases 6.1 and 8.1, you would select the “81TargetDestination” name that you created on the JMS Bridge Destination -> Configuration tab.
Selector	<p>Specify a selector 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.</p> <p>For more information on using selectors to filter messages, see “Developing a WebLogic JMS Application” in <i>Programming WebLogic JMS</i>.</p>

Table 289-2 Messaging Bridge Attributes on the General Tab

Attribute	Description
Quality Of Service (QOS)	<p>Select a quality-of-service guarantee for forwarding a message across a messaging bridge. The valid qualities of service are:</p> <p><i>Exactly-once</i> (default) — Each message will be sent exactly once. This is the highest quality of service. In order to use this QOS:</p> <ul style="list-style-type: none"> ■ Any WebLogic Server implementation must be release 6.1 or later. ■ The source and target JMS connection factories must be configured to use the <code>XAConnectionFactory</code>. ■ The transaction <code>jms-xa-adp.rar</code> adapter must be deployed and identified in the Adapter JNDI Name attribute as <code>"eis.jms.WLSConnectionFactoryJNDIXA"</code> for both the source and target destinations. <p><i>Atmost-once</i> — Each message is sent at most one time. Some messages may not be delivered to the target destination.</p> <p><i>Duplicate-okay</i> — Each message is sent at least one time. Duplicate messages can be delivered to the target destination.</p>
QOS Degradation Allowed	<p>Indicate whether the messaging bridge automatically degrades the requested QOS when the configured one is not available. If this occurs, a message is delivered to the WebLogic startup window (or log file). If QOS Degradation Allowed is not selected, and the messaging bridge cannot satisfy the requested QOS, it will result in an error and the messaging bridge will not start.</p>
Maximum Idle Time (seconds)	<p>For bridges running in asynchronous mode, specify the maximum amount of time the messaging bridge will sit idle before checking the health of its connections. For bridges running in synchronous mode, this determines the amount of time the messaging bridge can block on a receive call if no transaction is involved.</p>

Table 289-2 Messaging Bridge Attributes on the General Tab

Attribute	Description
Asynchronous Mode Enabled	<p>Indicate whether a messaging bridge works in asynchronous mode. Messaging bridges that work in asynchronous mode are driven by the source destination. The messaging bridge listens for messages and forwards them as they arrive. When Asynchronous Mode is disabled, the bridge works in synchronous mode, even if the source supports asynchronous receiving.</p> <p>Note: For a messaging bridge with a QOS of <i>Exactly-once</i> to work in asynchronous mode, the source destination has to support the <code>MDBTransaction</code> interface described in the weblogic.jms.extensions Javadoc. Otherwise, the bridge automatically switches to synchronous mode if it detects that <code>MDBTransactions</code> are not supported by the source destination. For more information about <code>MDBTransactions</code>, see “Using Message-Driven Beans” in <i>Programming WebLogic EJB</i>.</p>
Durability Enabled	<p>This is used only for JMS topics or for third-party destinations with similar characteristics as a JMS topic. By enabling durability, a messaging bridge creates a durable subscription for the source destination. 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 this attribute is not selected, messages that are sent to the source JMS topic while the bridge is down cannot be forwarded to the target destination.</p> <p>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 <i>Programming WebLogic JMS</i>.</p>

Table 289-2 Messaging Bridge Attributes on the General Tab

Attribute	Description
Started	<p>Indicates the initial state of the messaging bridge when it is configured and whenever the server is restarted. You can also use this field to dynamically start and stop the messaging bridge. To stop the bridge, clear the check box. Conversely, reselect the check box to restart the bridge.</p> <p>Note: Unless there is a configuration issue that prevents the messaging bridge from starting, this field indicates the expected run-time state of the messaging bridge. For information on monitoring all the configured messaging bridges in your domain, see “Monitoring All Messaging Bridges” on page 289-23.</p>

For more information about the general messaging bridge attributes, see the Attributes table in “Messaging Bridge --> Configuration --> General” on page 280-1.

- 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. If you selected the Started check box, the bridge will be in a running state once it is targeted.
- On the Target and Deploy tab, select an independent server instance, a cluster, or a migratable server target on which to deploy the messaging bridge. This must be the same target where the bridge’s resource adapter was deployed. You can also reconfigure deployment targets later if you wish.

For more information, see “Targeting a Messaging Bridge to a Server, a Cluster, or a Migratable Target” on page 289-16.

- Optionally, on the Connection Retry tab, change the attribute fields or accept the default values as assigned. Since the source and target destinations for a messaging bridge will not always be available, the messaging bridge must be able to reconnect to the destination at some periodic interval. These attributes govern the time between reconnection attempts. Then, click Apply to save your changes.

For more information about the bridge’s connection retry attributes, see the Attributes table in “Messaging Bridge --> Configuration --> Connection Retry” on page 279-1.

8. Optionally, on the Transactions tab, change the attribute fields or accept the default values as assigned. Then, click Apply to save your changes.

For more information about the bridge's transaction attributes, see the Attributes table in "Messaging Bridge --> Configuration --> Transactions" on page 281-1.

Targeting a Messaging Bridge to a Server, a Cluster, or a Migratable Target

You can choose the servers, clusters, or migratable targets in your domain on which you would like to deploy a messaging bridge. You can also reconfigure deployment targets later if you wish.

1. Expand the Messaging Bridge →Bridges node to show the list of messaging bridges defined in your domain.
2. Click the messaging bridge that you want to assign to a server, cluster, or migratable target. A dialog displays in the right pane showing the tabs associated with the messaging bridge instance.
3. Click the Target and Deploy tab to display the following targeting options.
 - Independent 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. You can also target an individual server or servers within a 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.

Note: This must be the same target where the bridge's resource adapter was deployed. For more information, see "Deploying the Bridge's Resource Adapters" on page 289-5.
4. Click Apply to save your assignments.

Using the Messaging Bridge to Interoperate with Different WebLogic Server Releases and Domains

The following interoperability guidelines apply when using the messaging bridge to access JMS destinations on different releases of WebLogic Server and in other WebLogic Server domains.

- “Naming Guidelines for WebLogic Servers and Domains” on page 289-17
- “Enabling Security Interoperability for WebLogic Domains” on page 289-18
- “Using the Messaging Bridge To Access Destinations In a Release 6.1 or Later Domain” on page 289-19
- “Using the Messaging Bridging To Access Destinations In a Release 6.0 Domain” on page 289-20
- “Using the Messaging Bridging To Access Destinations In a Release 5.1 Domain” on page 289-21

Note: When the messaging bridge is used to communicate between two domains running different releases of Weblogic Server, a best-practice recommendation is for the messaging bridge to be configured to run on the domain using the latest release of Weblogic Server.

Naming Guidelines for WebLogic Servers and Domains

Unique naming rules apply to all WebLogic Server deployments if more than one domain is involved. Therefore, make sure that:

- WebLogic Server instances and domain names are unique.
- WebLogic JMS server names are unique name across domains.

- If a JMS file store is being used for persistent messages, the JMS file store name must be unique across domains.

Enabling Security Interoperability for WebLogic Domains

Whenever the Exactly-once QOS (quality of service) is required for transactionally transferring messages across different WebLogic Server domains, you must establish a *trust relationship* between these domains by using a common security credential. This requirement applies to the source and target destination domains, as well as to the messaging bridge domain – if the messaging bridge is not running in the same domain as the source or target domains.

Follow these steps to establish a trusted relationship between release 6.1 or later WebLogic domains.

1. Configure the security for the 8.1 domain where the messaging bridge is running as follows:
 - a. Expand the Domains node (for example, Examples).
 - b. Click the View Domain-Wide Security Settings link on the General tab.
 - c. Select the Security Configuration →Advanced tab.
 - d. If necessary, clear the Enable Generated Credential check box.
 - e. Enter a password for the domain in the Credential field. This password must match the password used for the domain that you are interoperating with.
Note: When interoperating with a release 6.1 domain, the 7.0 or 8.1 Credential password must *exactly match* the “system” user password configured for the 6.1 domain.
 - f. Confirm the password in the Confirm field, and then click Apply.
2. When interoperating with a release 6.1 domain, make sure that “system” user is a member of the Administrators group in the 7.0 or 8.1 domain.

Note: For more information about WebLogic Server 7.0 or later domain interoperability security, see [“Enabling Trust Between WebLogic Domains”](#) in *Managing WebLogic Security*.

Using the Messaging Bridge To Access Destinations In a Release 6.1 or Later Domain

Use these guidelines when configuring a messaging bridge on a release 8.1 domain to provide “Exactly-once” transactional message communication between two release 6.1 or later domains.

Note: The *Exactly-once* quality of service for two-phase transactions is only supported for release 6.1 or later.

- If a JMS file store is being used for persistent messages, the JMS file store name must be unique across WebLogic domains, as described in “Naming Guidelines for WebLogic Servers and Domains” on page 289-17.
- Make sure that security interoperability between the domains is correctly configured, as described in “Enabling Security Interoperability for WebLogic Domains” on page 289-18.
- Make sure that the XA connection factory is enabled for the domains by selecting the XAConnection Factory Enabled check box on the Services →JMS →Connection Factories →Configuration →Transactions tab.
- Deploy the transaction resource adapter, `jms-xa-adj.rar`, on the 8.1 domain where the messaging bridge is running, as described in “Deploying the Bridge’s Resource Adapters” on page 289-5.
- When configuring the JMS bridge destinations, as described in “Configuring JMS Bridge Destinations” on page 289-7, do the following for both the source and target destinations:
 - In the Adapter JNDI Name field, identify the transaction adapter’s JNDI name, `eis.jms.WLSConnectionFactoryJNDIXA`.
 - Do not enter anything in the Adapter Classpath field.
- On the Messaging Bridge →Configuration →General tab, select a Quality Of Service of *Exactly-once*, as described in “Configuring a Messaging Bridge Instance” on page 289-11.

Using the Messaging Bridging To Access Destinations In a Release 6.0 Domain

When configuring a messaging bridge involves interoperability between WebLogic Server 8.1 and a release 6.0 domain, you must configure the following items on the Weblogic Server 8.1 domain that the bridge is running on:

Note: The *Exactly-once* QOS (quality of service) for transactions is not supported for WebLogic Server 6.0. For more information on the bridge QOS options, see the Attribute table in “Messaging Bridge --> Configuration --> General” on page 280-1.

- Deploy the non-transaction resource adapter, `jms-notran-adp.rar` on the 8.1 bridge domain, as described in “Deploying the Bridge’s Resource Adapters” on page 289-5.
- When configuring the JMS source and target destinations, as described in “Configuring JMS Bridge Destinations” on page 289-7, do the following:

In the Adapter JNDI Name field:

- For the source and target destinations, specify the non-transaction adapter’s JNDI name as `eis.jms.WLSConnectionFactoryJNDINOtx`.

In the Adapter Classpath field:

- For the 8.1 destination, leave the field blank.
- For the 6.0 destination, indicate the location of the classes for the WebLogic Server 6.0 release.

For example, if you have WebLogic Server 6.0 GA installed in a directory named `WL60_HOME`, then set the Adapter Classpath as follows for the 6.0 JMS bridge destination:

`WL60_HOME\lib\weblogic60.jar`

- On the Messaging Bridge → Configuration → General tab, select a Quality Of Service of *Atmost-once* or *Duplicate-okay*, as described in “Configuring a Messaging Bridge Instance” on page 289-11.

Using the Messaging Bridging To Access Destinations In a Release 5.1 Domain

When configuring a messaging bridge involves interoperability between WebLogic Server 8.1 and release 5.1, you must configure the following on the WebLogic Server 8.1 implementation that the bridge is running on:

Note: The *Exactly-once* QOS for transactions is not supported for WebLogic Server 5.1. For more information on the bridge QOS options, see the Attribute table in “Messaging Bridge --> Configuration --> General” on page 280-1.

- The `jms51-interop.jar` file in the `WL_HOME\server\lib` directory must be in the CLASSPATH of the WebLogic Server 8.1 implementation.
- The release 5.1 resource adapter (`jms-notran-adp51.rar`) and the non-transaction adapter (`jms-notran-adp.rar`) must be deployed on the 8.1 bridge domain, as described in “Deploying the Bridge’s Resource Adapters” on page 289-5.
- When configuring the JMS source and target destinations, as described in “Configuring JMS Bridge Destinations” on page 289-7, do the following:

In the Adapter JNDI Name field:

- For the 8.1 destination, specify the non-transaction adapter’s JNDI name as `eis.jms.WLSConnectionFactoryJNDINOtx`.
- For the 5.1 destination, specify the 5.1 adapter’s JNDI name as `eis.jms.WLS51ConnectionFactoryJNDINOtx`.

In the Adapter Classpath field:

- For the 8.1 destination, leave the field blank.
- For the 5.1 destination, indicate the location of the classes for the WebLogic Server 5.1 release, as well as the location of the `jms51-interop.jar` file for the 8.1 release.

For example, if you have WebLogic Server 5.1 GA installed in a directory named `WL51_HOME` and your WebLogic Server 8.1 release is installed in `WL81_HOME`, then set the Adapter Classpath as follows for the 5.1 destination:

```
WL51_HOME\classes;WL51_HOME\lib\weblogicaux.jar;  
WL81_HOME\server\lib\jms51-interop.jar
```

Note: If your implementation is using a 5.1 Service Pack, the corresponding *sp.jar* files must also be added to the Adapter Classpath field.

- On the Messaging Bridge → Configuration → General tab, select a Quality Of Service of *Atmost-once* or *Duplicate-okay*, as described in “Configuring a Messaging Bridge Instance” on page 289-11.

Using the Messaging Bridge to Access a Third-Party Messaging Provider

When configuring a messaging bridge involves interoperability with a third-party messaging provider, you must configure the following:

- Before starting WebLogic Server:
 - Supply the provider’s CLASSPATH in the WebLogic Server CLASSPATH.
 - Include the PATH of any native code required by the provider’s client-side libraries in the WebLogic Server system PATH. (This variable may vary depending on your operating system.)
- In the JMSBridgeDestination instance for the third-party messaging product being bridged, provide *vendor-specific* information in the following attributes:
 - Connection URL
 - Initial Context Factory
 - Connection Factory JNDI Name
 - Destination JNDI Name

Note: The messaging bridge cannot provide the “Exactly-once” quality of service when the source and target bridge destinations are located on the same resource manager (that is, when the bridge is forwarding a global transaction that is using the XA resource of the resource manager). For example, when using MQ Series, it is not possible to use the same Queue Manager for the source and target bridge destinations.

For more information on configuring the remaining attributes for a JMS Bridge Destination, see “Configuring JMS Bridge Destinations” on page 289-7.

Managing a Messaging Bridge

Once a messaging bridge is up and running, it can be managed from the console.

- “Monitoring All Messaging Bridges” on page 289-23
- “Stopping and Restarting a Messaging Bridge” on page 289-23
- “Configuring the Messaging Bridge Execute Thread Pool Size” on page 289-24

Monitoring All Messaging Bridges

To monitor the status of all configured messaging bridges in your domain:

1. Expand the Server node.
2. Select the server instance where the messaging bridges are configured. A dialog displays in the right pane showing the tabs associated with the selected server instance.
3. Select the Services → Bridge tab.
4. Click the Monitor all Messaging Bridge Runtimes text link.
5. A table displays showing all the messaging bridge instances for the server and their status (either as running or not running).

Stopping and Restarting a Messaging Bridge

To temporarily suspend and restart an active messaging bridge:

1. Expand the Messaging Bridge node.

2. Select the messaging bridge instance that you want to suspend.
3. On the Configuration →General tab, clear the Started check box to suspend the bridge.
4. To restart the bridge, select the Started check box.

Configuring the Messaging Bridge 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 a messaging bridge to use the WebLogic Server default thread pool.

1. Expand the Servers node.
2. Select the specific server instance where the messaging bridge is configured.
3. In the right pane, select the Services →Bridge tab.
4. Enter a new value in the Messaging Bridge Thread Pool Size field.
5. Click Apply to save your changes.

Attributes and Console Screen Reference for Messaging Bridge

For information about an Administration Console screen and the attributes you can configure, select one of the links in the following categories:

Messaging Bridge

[“Messaging Bridge” on page 284-1](#)

[“Messaging Bridge --> Configuration --> General” on page 280-1](#)

[“Messaging Bridge --> Configuration --> Connection Retry” on page 279-1](#)

[“Messaging Bridge --> Configuration --> Transactions” on page 281-1](#)

[“Messaging Bridge --> Target and Deploy” on page 283-1](#)

[“Messaging Bridge --> Notes” on page 282-1](#)

[“Messaging Bridge Runtime” on page 285-1](#)

JMS Bridge Destination

[“JMS Bridge Destination” on page 287-1](#)

[“JMS Bridge Destination --> Configuration” on page 288-1](#)

[“JMS Bridge Destination --> Notes” on page 286-1](#)

General Bridge Destination

[“General Bridge Destination” on page 278-1](#)

“General Bridge Destination --> Configuration” on page 276-1

“General Bridge Destination --> Notes” on page 277-1

Create a New ACL

Use this page to create a new access control lists (ACLs) for a WebLogic resource in the CompatibilityRealm. This page applies only to WebLogic Server deployments running Compatibility security.

- For more information, see [“Defining ACLs” on page 322-30](#).



ACL Permission

Use this page to specify what users or groups have the specified permission to the WebLogic resource. This page applies only to WebLogic Server deployments using Compatibility security.

- For more information, see [“Defining ACLs” on page 322-30](#).



ACL

This page lists all the ACLs defined in the CompatibilityRealm. This page applies only to WebLogic Server deployments using Compatibility security.

- For more information, see [“Defining ACLs” on page 322-30](#).



Caching Realm-->ACL

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure and enable the ACL cache for the Caching realm. To use the Caching realm, you need to use Compatibility security. The use of the Caching realm is deprecated in this release of WebLogic Server.

Tasks

[“Enabling the ACL Cache” on page 322-6](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 294-1

Attribute Label	Description	Value Constraints
Enable ACL Cache	Enables the ACL cache. By default,the ACL cache is enabled.	<i>Default:</i> true <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false
ACL Cache Size	Maximum number of ACL lookups to cache. This attribute should be a prime number for best lookup performance.	<i>Minimum:</i> 17 <i>Maximum:</i> 65537 <i>Default:</i> 211
ACL Cache Positive TTL	Number of seconds to retain the results of a successful ACL lookup.	<i>Minimum:</i> 1 <i>Maximum:</i> 2147483647 <i>Default:</i> 60
ACL Cache Negative TTL	Number of seconds to retain the results of an unsuccessful ACL lookup.	<i>Minimum:</i> 1 <i>Maximum:</i> 2147483647 <i>Default:</i> 10

Caching Realm --> Authentication

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure and enable the Authentication cache for the Caching realm. To use the Caching realm, you need to use Compatibility security. The use of the Caching realm is deprecated in this release of WebLogic Server.

Tasks

[“Enabling the Authentication Cache” on page 322-6](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 295-1

Attribute Label	Description	Value Constraints
Enable Authentication Cache	Enables the Authentication cache. By default, the Authentication cache is enabled.	<i>Default:</i> true <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false
Authentication Cache Size	Maximum number of Authentication requests to cache. This attribute should be a prime number for best lookup performance.	<i>Minimum:</i> 17 <i>Maximum:</i> 65537 <i>Default:</i> 211
Authentication Cache TTLPositive	Number of seconds to retain the results of a successful Authentication lookup.	<i>Minimum:</i> 1 <i>Maximum:</i> 2147483647 <i>Default:</i> 60
Authentication Cache TTLLegative	Number of seconds to retain the results of an unsuccessful Authentication lookup.	<i>Minimum:</i> 1 <i>Maximum:</i> 2147483647 <i>Default:</i> 10

Caching Realm --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure the Caching realm. The Caching realm works with the File realm, alternate security realms, or custom security realms to fulfill client requests with the proper authentication and authorization. The Caching realm manages separate caches for users, groups, permissions, access control lists (ACLs), and authentication requests. To use the Caching realm, you need to use Compatibility security. The use of the Caching realm is deprecated in this release of WebLogic Server.

Tasks

[“Configuring the Caching Realm” on page 322-5](#)

[“Enabling the ACL Cache” on page 322-6](#)

[“Enabling the Authentication Cache” on page 322-6](#)

[“Enabling the Group Cache” on page 322-7](#)

[“Enabling the User Cache” on page 322-7](#)

[“Enabling the Permission Cache” on page 322-7](#)

[“Adding a Note to the Caching Realm” on page 322-7](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 296-1

Attribute Label	Description	Value Constraints
Name	The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.	
Basic Realm	Name of the class for the alternate security realm or custom security realm to be used with the Caching Realm. The names of the configured realms appear in the Basic Realms attribute of the Caching Realm table. This attribute is required.	
Case Sensitive Cache	Defines whether the specified security realm is case-sensitive. By default, this attribute is enabled. To use a realm that is not case-sensitive (such as the Windows NT and LDAP security realms), disable this attribute.	<i>Default:</i> true <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false

Caching Realm --> Groups

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure and enable the Group cache for the Caching realm. To use the Caching realm, you need to use Compatibility security. The use of the Caching realm is deprecated in this release of WebLogic Server.

Tasks

[“Enabling the Group Cache” on page 322-7](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 297-1

Attribute Label	Description	Value Constraints
Enable Group Cache	Enables the Group cache. By default, the Group cache is enabled.	<i>Default:</i> true <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false
Group Cache Size	Maximum number of Group lookups to cache. This attribute should be a prime number for best performance.	<i>Minimum:</i> 17 <i>Maximum:</i> 65537 <i>Default:</i> 211
Group Cache TTLPositive	Number of seconds to retain the results of a successful Group lookup.	<i>Minimum:</i> 1 <i>Maximum:</i> 2147483647 <i>Default:</i> 60
Group Cache TTLLegative	Number of seconds to retain the results of an unsuccessful Group lookup.	<i>Minimum:</i> 1 <i>Maximum:</i> 2147483647 <i>Default:</i> 10
Group Membership Cache TTL	Number of seconds to store the members of a Group before updating it.	<i>Minimum:</i> 1 <i>Default:</i> 300

Caching Realm --> Permissions

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure and enable the Permissions cache for the Caching realm. To use the Caching realm, you need to use Compatibility security. The use of the Caching realm is deprecated in this release of WebLogic Server.

Tasks

[“Enabling the Permission Cache” on page 322-7](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security](#) page in the WebLogic Server documentation

Attributes

Table 298-1

Attribute Label	Description	Value Constraints
Enable Permission Cache	Enables the Permission cache. By default, the Permission cache is enabled.	<i>Default:</i> true <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false
Permission Cache Size	Maximum number of Permission lookups to cache. This attribute should be a prime number for best performance.	<i>Minimum:</i> 17 <i>Maximum:</i> 65537 <i>Default:</i> 211
Permission Cache TTLPositive	Number of seconds to retain the results of a successful Permission lookup.	<i>Minimum:</i> 1 <i>Maximum:</i> 2147483647 <i>Default:</i> 60
Permission Cache TTLLNegative	Number of seconds to retain the results of an unsuccessful Permission lookup.	<i>Minimum:</i> 1 <i>Maximum:</i> 2147483647 <i>Default:</i> 10

Caching Realm --> Users

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure and enable the Users cache for the Caching realm. To use the Caching realm, you need to run Compatibility security. The use of the Caching realm is deprecated in this release of WebLogic Server.

Tasks

[“Enabling the User Cache” on page 322-7](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 299-1

Attribute Label	Description	Value Constraints
Enable User Cache	Enables the User cache. By default, the User cache is enabled.	<i>Default:</i> true <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false
User Cache Size	Maximum number of User lookups to cache. This attribute should be a prime number for best performance.	<i>Minimum:</i> 17 <i>Maximum:</i> 65537 <i>Default:</i> 211
User Cache TTLPositive	Number of seconds to retain the results of a successful User lookup.	<i>Minimum:</i> 1 <i>Maximum:</i> 2147483647 <i>Default:</i> 60
User Cache TTLNegative	Number of seconds to retain the results of an unsuccessful User lookup.	<i>Minimum:</i> 1 <i>Maximum:</i> 2147483647 <i>Default:</i> 10

Caching Realm --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to store notes about the selected Caching realm. To use the Caching realm, you need to use Compatibility security. The use of the Caching realm is deprecated in this release of WebLogic Server.

Tasks

[“Adding a Note to the Caching Realm” on page 322-7](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security](#) page in the WebLogic Server documentation

Attributes

Table 300-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration.	<i>Dynamic:</i> yes

Caching Realm

This table lists the names of the Caching realms configured for this WebLogic Server deployment, the name of the realm configured as the Basic realm (for example, the security realm to be used with the Caching realm) for this WebLogic Server deployment, and the caches that are enabled for each Caching realm.

The use of the Caching realm is deprecated in this release of WebLogic Server.

To configure a new Caching realm, click on the [Configure a new Caching Realm...](#) link.

- For more information, see [“Configuring the Caching Realm”](#) on page 322-5.



Custom Realm --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Custom security realms allow you to use an existing data store such as a directory server or a database when authenticating and authorizing users to WebLogic Server. Use this page to configure a Custom security realm. To use a Custom security realm, you need to use Compatibility security. The use of Custom security realms is deprecated in this release of WebLogic Server.

Tasks

[“Installing a Custom Security Realm” on page 322-25](#)

[“Adding A Note To A Custom Security Realm” on page 322-26](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

[The Security topics in the WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 302-1

Attribute Label	Description	Value Constraints
Name	The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.	
Realm Class Name	The name of Java class that implements the custom security realm.	
Configuration Data	Information needed to connect to the security store of the custom security realm.	
Password	Password for the custom security realm. If a password is supplied, WebLogic Server encrypts it.	<i>Encrypted: yes</i>

Custom Realm --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to store notes about the selected custom security realm. To use a custom security realm, you need to use Compatibility security. The use of custom security realms is deprecated in this release of WebLogic Server.

Tasks

[“Adding A Note To A Custom Security Realm” on page 322-26](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 303-1

Attribute Label	Description	Value Constraints
Notes	The notes attribute of the ConfigurationMBean object	<i>Dynamic: yes</i>

Groups

This page lists all groups defined in the security realm.

To create a new group, click the [Configure a New Group...](#) link on this page.

To delete a group, click on the trash can icon in the Groups table.

- For more information, see [“Defining Groups” on page 322-28](#).



Group-->Group

Use this page to create a new group in the active security realm and add users and groups to the new group. Also, use this page to delete users from the group.

- For more information, see [“Defining Groups” on page 322-28](#).



LDAP Security

Realm-->Configuration-->General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure the LDAP security realm. LDAP security realm supports the use of Open LDAP, Netscape iPlanet, Microsoft Site Server, or Novell NDS LDAP servers.

When using the LDAP security realm, you must configure a Caching realm. When configuring the Caching realm, choose the LDAP security realm as the Basic realm. The Basic attribute defines the association between the Caching realm and the LDAP security realm.

To use the LDAP security realm, you need to use Compatibility security. The use of the LDAP security realm is deprecated in this release of WebLogic Server.

Tasks

[“Configuring an LDAP V1 Security Realm” on page 322-8](#)

[“Enabled Communication between the LDAP Server and WebLogic Server” on page 322-9](#)

[“Specifying How Users Are Located in the LDAP V1 Security Realm” on page 322-9](#)

[“Specifying How Groups Are Located in the LDAP V1 Security Realm” on page 322-10](#)

[“Configuring the Caching Realm” on page 322-5](#)

Related Topics

- [Introduction to WebLogic Security](#)
- [Managing WebLogic Security](#)
- [Securing WebLogic Resources](#)
- [Programming WebLogic Security](#)
- [Developing Security Providers for WebLogic Server](#)
- [Securing a Production Environment](#)
- The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)
- [Security FAQ](#)
- The [Security](#) page in the WebLogic Server documentation

Attributes

Table 306-1

Attribute Label	Description	Value Constraints
Name	The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.	
Realm Class Name	The realmClassName attribute of the BasicRealmMBean object	

LDAP Security Realm-->Groups

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to specify how groups are stored in the LDAP server. To use the LDAP security realm, you need to use Compatibility security. The use of the LDAP security realm is deprecated in this release of WebLogic Server.

Tasks

[“Configuring an LDAP V1 Security Realm” on page 322-8](#)

[“Enabled Communication between the LDAP Server and WebLogic Server” on page 322-9](#)

[“Specifying How Users Are Located in the LDAP V1 Security Realm” on page 322-9](#)

[“Specifying How Groups Are Located in the LDAP V1 Security Realm” on page 322-10](#)

[“Adding a Note to the LDAP V1 Security Realm” on page 322-10](#)

[“Configuring the Caching Realm” on page 322-5](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 307-1

Attribute Label	Description	Value Constraints
Group DN	The list of attributes that, when combined with the attribute named in the <code>groupNameAttribute</code> attribute, uniquely identifies a group in the LDAP directory.	<i>Default:</i>
Group Name Attribute	The name of a group in the LDAP directory. It is usually the common name.	<i>Default:</i> "cn"
Group Is Context	Specifies how group membership is recorded in the LDAP directory. Set to true if each group entry contains one user. Set to false if there is one group entry containing an attribute for each group member.	<i>Default:</i> true <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false
Group Username Attribute	The name of the attribute that contains a group member in a group entry.	<i>Default:</i> "member"

LDAP Security Realm-->LDAP Server

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure the LDAP directory server to enable communication between the LDAP directory server and WebLogic Server.

To use the LDAP security realm, you need to use Compatibility security. The use of the LDAP security realm is deprecated in this release of WebLogic Server.

Tasks

[“Configuring an LDAP V1 Security Realm” on page 322-8](#)

[“Enabled Communication between the LDAP Server and WebLogic Server” on page 322-9](#)

[“Specifying How Users Are Located in the LDAP V1 Security Realm” on page 322-9](#)

[“Specifying How Groups Are Located in the LDAP V1 Security Realm” on page 322-10](#)

[“Adding a Note to the LDAP V1 Security Realm” on page 322-10](#)

[“Configuring the Caching Realm” on page 322-5](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 308-1

Attribute Label	Description	Value Constraints
LDAPURL	<p>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 the LDAP server is listening.</p> <p>If you want WebLogic Server to connect to the LDAP server using the SSL protocol, use the SSL port of the LDAP server in the server URL.</p>	<i>Default:</i> "ldap://ldapservice:389"
Principal	<p>The distinguished name of the LDAP user that WebLogic Server uses to connect to the LDAP server. This user must be able to list LDAP users and groups.</p>	<i>Default:</i> null
Credential	<p>The password that authenticates the LDAP user defined in the Principal attribute. This password is automatically encrypted.</p>	<i>Default:</i> null <i>Encrypted:</i> yes

Table 308-1

Attribute Label	Description	Value Constraints
Enable SSL	<p>Enables the use of the SSL protocol to protect communications between the LDAP server and WebLogic Server. Keep in mind the following:</p> <ul style="list-style-type: none">■ Disable this attribute if the LDAP server is not configured to use the SSL protocol.■ If you set the User Authentication attribute to <code>external</code>, this attribute must be enabled.	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false
Auth Protocol	<p>The type of authentication used to authenticate the LDAP server. Set the attribute to one of the following:</p> <ul style="list-style-type: none">■ None for no authentication.■ Simple for password authentication.■ CRAM-MD5 for certificate authentication. <p>Netscape Directory Server supports CRAM-MD5. Microsoft Site Server and Novell NDS support simple.</p>	<p><i>Default:</i> "none"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "none"■ "simple"■ "CRAM-MD5"



LDAP Security Realm-->Users

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to define how users are stored in the LDAP server. To use the LDAP security realm, you need to use Compatibility security. The use of the LDAP security realm is deprecated in this release of WebLogic Server.

Tasks

[“Configuring an LDAP V1 Security Realm” on page 322-8](#)

[“Enabled Communication between the LDAP Server and WebLogic Server” on page 322-9](#)

[“Specifying How Users Are Located in the LDAP V1 Security Realm” on page 322-9](#)

[“Specifying How Groups Are Located in the LDAP V1 Security Realm” on page 322-10](#)

[“Adding a Note to the LDAP V1 Security Realm” on page 322-10](#)

[“Configuring the Caching Realm” on page 322-5](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 309-1

Attribute Label	Description	Value Constraints
User Authentication	<p>Determines the method for allowing or denying a user the right to communicate with WebLogic Server. Set this attribute to one of the following values:</p> <ul style="list-style-type: none">■ Bind--The LDAP security realm retrieves user data, including the password for the LDAP server, and checks the password in WebLogic Server.■ External--The LDAP security realm authenticates a user by attempting to bind to the LDAP server with the username and password supplied by the WebLogic client.■ Local--The LDAP security realm authenticates a user by looking up the UserPassword attribute in the LDAP directory and checking its value against a set of passwords in WebLogic Server.	<p><i>Default:</i> "bind"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "bind"■ "external"■ "local"

Table 309-1

Attribute Label	Description	Value Constraints
User Password Attribute	If the <code>userAuthentication</code> attribute is set to <code>local</code> , this attribute finds the attribute in the LDAP user objects that contains the passwords of the LDAP users.	<i>Default:</i> "userpassword"
User DN	<p>A list of attributes that, when combined with the attribute named in the <code>userNameAttribute</code> attribute, uniquely identifies a user in the LDAP directory.</p> <p>When specifying this attribute, use the following format:</p> <p><code>ou=Barb.Klock, u=acme.com</code></p>	<i>Default:</i>
User Name Attribute	The login name of a user for the LDAP directory. The value of this attribute can be the common name of a user in the LDAP directory. However, it is generally an abbreviated string, such as a User ID.	



LDAP Security Realm-->Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to store notes about the selected LDAP security realm. To use the LDAP security realm, you need to use Compatibility security. The use of the LDAP security realm is deprecated in this release of WebLogic Server.

Tasks

[“Adding a Note to the LDAP V1 Security Realm” on page 322-10](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 310-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration.	<i>Dynamic:</i> yes

Windows NT Realm-->Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The Windows NT security realm uses account information defined for a Windows NT domain to authenticate users and groups. Use this page to define a name for the realm and the computer on which the Windows NT domain is running.

When using the Windows NT security realm, you must configure a Caching realm. When configuring the Caching realm, select the Windows NT security realm as the Basic realm. The Basic attribute defines the association between the Caching realm and the Windows NT security realm.

To use the Windows NT security realm, you need to use Compatibility security. The use of the Windows NT security realm is deprecated in this release of WebLogic Server.

Tasks

[“Configuring the Windows NT Security Realm” on page 322-16](#)

[“Adding a Note to the Windows NT Security Realm” on page 322-19](#)

[“Configuring the Caching Realm” on page 322-5](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security](#) page in the WebLogic Server documentation

Attributes

Table 311-1

Attribute Label	Description	Value Constraints
Name	The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.	
Primary Domain	The host name of the Windows NT Primary Domain Controller where users and groups are defined for the domain. If entering multiple sets of host names, delimit them with commas. If the local computer(where WebLogic Server executes) is the Primary Controller, you set set the attribute to period (".").	<i>Default:</i> null
Realm Class Name	The realmClassName attribute of the BasicRealmMBean object	

Windows NT Realm-->Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to store notes about the selected Windows NT security realm. To use the Windows NT security realm, you need to use Compatibility security. The use of the Windows NT security realm is deprecated in this release of WebLogic Server.

Tasks

[“Adding a Note to the Windows NT Security Realm” on page 322-19](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security](#) page in the WebLogic Server documentation

Attributes

Table 312-1

Attribute Label	Description	Value Constraints
Notes	The notes attribute of the ConfigurationMBean object	<i>Dynamic: yes</i>

RDBMS Realm-->Database

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to define attributes for the JDBC driver being used to connect to the database in the RDBMS security realm. To use the RDBMS security realm, you need to use Compatibility security. The use of the RDBMS security realm is deprecated in this release of WebLogic Server.

Tasks

[“Configuring the RDBMS Security Realm” on page 322-22](#)

[“Defining Database Attributes for the RDBMS Security Realm” on page 322-23](#)

[“Defining Database Schema for the RDBMS Security Realm” on page 322-24](#)

[“Configuring the Caching Realm” on page 322-5](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 313-1

Attribute Label	Description	Value Constraints
Driver	The Java class name for the database driver used with the RDBMS security realm.	<i>Default:</i> ""
URL	The location of the database. Change the URL to the name of the computer on which the database is running and the number of the port at which the database is listening.	<i>Default:</i> ""
User Name	The username used to login into the database.	
Password	The password required to log into the database.	<i>Encrypted:</i> yes

RDBMS Realm-->General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The RDBMS security realm is a BEA-provided custom security realm that stores users, groups, and access control lists (ACLs) in a relational database. Use this page to define a name for the realm and specify the name of the WebLogic class that implements the RDBMS security realm. The Java class needs to be in the CLASSPATH of WebLogic Server.

When using the RDBMS security realm, you must configure a Caching realm. When configuring the Caching realm, select the RDBMS security realm as the Basic realm. The Basic attribute defines the association between the Caching realm and the RDBMS security realm.

To use the RDBMS security realm, you need to use Compatibility security. The use of the RDBMS security realm is deprecated in WebLogic Server 7.0.

Tasks

[“Configuring the RDBMS Security Realm” on page 322-22](#)

[“Defining Database Attributes for the RDBMS Security Realm” on page 322-23](#)

[“Defining Database Schema for the RDBMS Security Realm” on page 322-24](#)

[“Configuring the Caching Realm” on page 322-5](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 314-1

Attribute Label	Description	Value Constraints
Name	The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.	
Realm Class	The name of the Java class that implements the RDBMS security realm. This class should be included in the CLASSPATH of WebLogic Server.	<i>Default:</i> ""

RDBMS Realm-->Schema

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to define the database schema used to store users, groups, and access control lists (ACLs) in the database in the RDBMS security realm. To use the RDBMS security realm, you need to use Compatibility security. The use of the RDBMS security realm is deprecated in this release of WebLogic Server.

Tasks

[“Configuring the RDBMS Security Realm” on page 322-22](#)

[“Defining Database Attributes for the RDBMS Security Realm” on page 322-23](#)

[“Defining Database Schema for the RDBMS Security Realm” on page 322-24](#)

[“Configuring the Caching Realm” on page 322-5](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security](#) page in the WebLogic Server documentation

Attributes

Table 315-1

Attribute Label	Description	Value Constraints
Schema Properties (key=value)	The schema properties (the prepared statements) for manipulating the database. Specify an open-ended properties list so that additional properties can be added to the code for the RDBMS security realm.	

RDBMS Realm-->Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to store notes about the selected RDBMS security realm. To use the RDBMS security realm, you need to use Compatibility security. The use of the RDBMS security realm is deprecated in this release of WebLogic Server.

Tasks

[“Adding A Note to the RDBMS Security Realm” on page 322-25](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 316-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration.	<i>Dynamic:</i> yes

Basic Realm

Use this page to specify the name of the security realm (for example, an alternative security realm or a custom security realm) to be using with the Caching realm. The available realms appear in the window.

To configure a security realm, click on the `Configure a New Realm...` link.

- For more information, see [“Configuring the Caching Realm” on page 322-5](#).



Unix Realm-->Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The UNIX security realm executes a small native program, `wlauth`, to look up users and groups and to authenticate users on the basis of their UNIX login names and passwords. The UNIX security realm can only be used on the Solaris and Linux platforms. Use this page to define a name for the realm and the name of the program used to authenticate users in the UNIX security realm.

When using the UNIX security realm, you must configure a Caching realm. When configuring the Caching realm, select the UNIX security realm as the Basic realm. The Basic attribute defines the association between the Caching realm and the UNIX security realm.

To use the UNIX security realm, you need to use Compatibility security. The use of the UNIX security realm is deprecated in this release of WebLogic Server.

Tasks

[“Configuring the `wlauth` Program for the UNIX Security Realm” on page 322-19](#)

[“Adding a Note to the UNIX Security Realm” on page 322-21](#)

[“Configuring the UNIX Security Realm” on page 322-21](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 318-1

Attribute Label	Description	Value Constraints
Name	The name of this configuration.	
Auth Program	<p>The name of the program used to authenticate users in the UNIX security realm. In most cases, the name of the program is <code>wlauth</code>. The authentication program must run <code>setuid root</code>.</p> <ul style="list-style-type: none">■ If the program name is <code>wlauth</code> and is in the <code>CLASSPATH</code>, you need not explicitly set this attribute; leave the attribute blank.■ If the program name is different than <code>wlauth</code>, or if it is not in the <code>CLASSPATH</code> of WebLogic Server, specify this attribute.	<i>Default:</i> "wlauth"
Realm Classname	The <code>realmClassName</code> attribute of the <code>BasicRealmMBean</code> object.	

Unix Realm-->Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to store notes about the selected UNIX security realm. To use the UNIX security realm, you need to use Compatibility security. The use of the UNIX security realm is deprecated in this release of WebLogic Server.

Tasks

[“Configuring the UNIX Security Realm” on page 322-21](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 319-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration.	<i>Dynamic:</i> yes

Unlock User Accounts

[Overview](#) [Tasks](#) [Related Topics](#)

Overview

WebLogic Server defines a set of attributes to protect user accounts from intruders. In the default security configuration, these attributes are set for maximum protection.

As a system administrator, you have the option of turning off all the attributes, increasing the number of login attempts before a user account is locked, increasing the time period in which invalid login attempts are made before locking the user account, and changing the amount of time a user account is locked. Remember that changing the attributes lessens security and leaves user accounts vulnerable to security attacks.

If a user account exceeds the values set for the attributes on this page, the user account becomes locked and the table on the Users page has the word `Details` in the table row for the user account. If a user account is locked repeatedly, further investigation is required. Repeated login failures could be a sign that a hacker is trying to break into the system.

Note: The User Lockout attributes apply to the security realm and all its security providers. If you are using an Authentication provider that has its own mechanism for protecting user accounts, disable the Lockout Enabled attribute.

If a user account becomes locked and you delete the user account and add another user account with the same name and password, the UserLockout attribute will not be reset.

Tasks

[“Unlocking A User Account” on page 322-27](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security](#) page in the WebLogic Server documentation

Users

Use this page to add users to a manageable security realm (for example, the File realm) in the CompatibilityRealm.

To define a new user in the security realm, click the `Configure a new User...` link.

To delete a user, click on the trash can icon in the Users table.

- For more information, see [“Defining Users” on page 322-26](#).



1 Compatibility Security

[“Attributes and Console Screen Reference for Compatibility Security” on page 323-1]

This topic describes configuring and managing security when using Compatibility security. For more information, see [Using Compatibility Security](#) in *Managing WebLogic Security*. For information about using the security features in WebLogic Server, see “Security” on page 428-1 in the Administration Console online help and [Managing WebLogic Security](#).

Tasks

Setting Up Compatibility Security: Main Steps

To set up Compatibility security:

1. Make a back-up copy of your 6.x WebLogic domain (including your `config.xml` file) before using Compatibility security.
2. Add the following to the 6.x `config.xml` file if it does not exist:

```
<Security Name="mydomain" Realm="mysecurity" />
<Realm Name="mysecurity" FileRealm="myrealm" />
<FileRealm Name="myrealm" />
```
3. Install WebLogic Server in a new directory location. Do not overwrite your existing 6.x installation directory. For more information, see the [WebLogic Server Installation Guide](#).
4. Modify the start script for your 6.x server to point to the new WebLogic Server installation. Specifically, you need to modify:

- The classpath to point to the `weblogic.jar` file in the new WebLogic Server installation.
 - The `JAVA_HOME` variable to point to the new WebLogic Server installation.
5. Use the start script for your 6.x server to boot WebLogic Server.

To verify whether you are correctly running Compatibility security, do the following:

1. In the WebLogic Server Administration Console, expand the Domain node.
2. Click on your WebLogic Server domain (referred to as the domain).
3. Click the View the Domain Log link.

The following message appears in the log:

```
Security initializing using realm CompatibilityRealm
```

In addition, a `CompatibilitySecurity` node will appear in the WebLogic Server Administration Console.

Configuring the Identity Assertion Provider in the Realm Adapter Authentication Provider

The Realm Adapter Authentication provider includes an Identity Assertion provider. The Identity Assertion provider provides backward compatibility for implementations of the `weblogic.security.acl.CertAuthenticator` class. The identity assertion is performed on X.509 tokens. By default, the Identity Assertion provider is not enabled in the Realm Adapter Authentication provider.

To enable identity assertion in the Realm Adapter Authentication provider:

1. Expand the Security-->Realms nodes.
2. Click the `CompatibilityRealm`.
3. Expand the Providers node.
4. Click Authentication Providers.
5. Click the Realm Adapter Authenticator link in the Realms table.

The General tab appears.

6. Enter X.509 in the Active Types list box.

This step enables the use of 6.x Cert Authenticators.

7. Click Apply.
8. Reboot WebLogic Server.

Configuring a Realm Adapter Auditing Provider

The Realm Adapter Auditing provider allows you to use implementations of the `weblogic.security.audit.AuditProvider` class when using Compatibility security. In order for the Realm Adapter Auditing provider to work properly, the implementation of the `weblogic.security.audit.AuditProvider` class must have been defined in the Audit Provider class attribute on the Domain-->Security-->Compatibility-->General tab.

To configure a Realm Adapter Auditing provider:

1. Expand the Compatibility Security-->Realms nodes.
2. Expand the Providers node.
3. Click Auditors.
4. Click Configure a Realm Adapter Auditor... link.

The General tab appears

5. Click Create to save your changes.
6. Reboot WebLogic Server.

Changing the System Password

During installation, WebLogic Server does the following to the File realm in *mydomain*:

1. Adds the username and password supplied during installation to the File realm.
2. Sets the system password to password specified during installation.

These steps ensure that a `system` user is defined in the compatibility version of the File realm.

When using the Configuration Wizard to create a new WebLogic Server domain, WebLogic Server sets the system password in the File realm in *mydomain* to the password of the first user defined in the Admin role. If the Admin role is mapped only to the Administrators group, the system password is the password of the first alphabetical user in the Administrators group.

To improve security, BEA recommends frequently changing the system password that was set during installation. Each WebLogic Server deployment must have a unique password.

1. In the console for the Administration Server, expand the Compatibility Security node.
2. Select the Users tab.
3. In the User Configuration window, under Change a User's Password, enter `system` in the Name attribute.
4. In the Old Password attribute, enter 6.x password.
5. Enter a new password in the New Password attribute.
6. Enter the new password again in the Confirm the Password attribute.

When you use an Administration Server and Managed Servers in a domain, the Managed Server must always use the password for the Administration Server in the domain. Always change the password for the Administration Server through the WebLogic Server Administration Console. When WebLogic Server is rebooted, the new password is propagated to all the Managed Servers in the domain.

Configuring the File Realm

To configure the File realm:

1. Expand the Domain node (for example, `mydomain`).
2. Click the View Domain-Wide Security Settings link at the bottom of the General tab.

3. Select the Compatibility-->File Realm tab.
4. Enter values in the attribute fields on the File Realm tab.
5. Click Apply to save your changes.

All user and group data for the File realm is stored in the `fileRealm.properties` file. If the `fileRealm.properties` file becomes corrupted or is destroyed, you must reconfigure the security information for WebLogic Server. Compatibility security cannot run without a `fileRealm.properties` file. Even if you write a custom security realm, you still need a `fileRealm.properties` file to boot WebLogic Server. Therefore, BEA recommends that you take the following steps:

1. Make a backup copy of the `fileRealm.properties` file and put it in a secure place.
2. Set the permissions on the `fileRealm.properties` file such that the administrator of the WebLogic Server deployment has write and read privileges and no other users have any privileges.

Note: Also make a backup copy of the `SerializedSystemIni.dat` file for the File realm.

Configuring the Caching Realm

To configure the Caching realm:

1. Configure the alternate or custom security realm with which you will use the Caching realm. See the appropriate realm configuration procedures in the following sections:
 - “Configuring an LDAP V1 Security Realm” on page 322-8
 - “Configuring an LDAP Realm V2” on page 322-10
 - “Configuring the Windows NT Security Realm” on page 322-16
 - “Configuring the UNIX Security Realm” on page 322-21
 - “Configuring the RDBMS Security Realm” on page 322-22
 - “Installing a Custom Security Realm” on page 322-25
2. Expand the Compatibility Security-->Caching Realms nodes.

3. Click the Configure a new Caching Realm... link.
4. Enter values in the attribute fields on the [Caching Realm --> General](#) page.
5. Click Create.
6. Enable the caches you want to use with the Caching realm. For more information, see:
 - [Enabling the ACL Cache](#)
 - [Enabling the Authentication Cache](#)
 - [Enabling the Group Cache](#)
 - [Enabling the User Cache](#)
 - [Enabling the Permission Cache](#)
7. When you finish enabling caches for the Caching realm, reboot WebLogic Server.

Enabling the ACL Cache

To enable the ACL cache:

1. Click the ACL tab under the Caching Realm tab.
2. Configure and enable the ACL cache by defining values for the attributes shown on the [Caching Realm-->ACL](#) page.
3. Click Apply to save your changes.

Enabling the Authentication Cache

To enable the Authentication cache:

1. Click the Authentication tab under the Caching Realm tab.
2. Configure and enable the Authentication cache by defining values for the attributes shown on the [Caching Realm --> Authentication](#) page.
3. Click Apply to save your changes.

Enabling the Group Cache

To enable the Group cache:

1. Click the Group tab under the Caching Realm tab.
2. Configure and enable the Group cache by defining values for the attributes shown on the [Caching Realm --> Groups](#) page.
3. Click Apply to save your changes.

Enabling the User Cache

To enable the User cache:

1. Click on the User tab under the Caching Realm tab.
2. Configure and enable the User cache by defining values for the attributes shown on the [Caching Realm --> Users](#) page.
3. Click Apply to save your changes.

Enabling the Permission Cache

To enable the Permission cache:

1. Click on the Permission tab under the Caching Realm tab.
2. Configure and enable the Permission cache by defining values for the attributes shown on the [Caching Realm --> Permissions](#) page.
3. Click Apply to save your changes.

Adding a Note to the Caching Realm

To add a note to the caching realm:

1. Click on the Notes tab under the Caching Realm tab.
2. Write any pertinent information in the Notes field.
3. Click Apply to save your changes.

Configuring an LDAP V1 Security Realm

The Lightweight Directory Access Protocol (LDAP) V1 security realm provides authentication through users and groups stored in an LDAP directory. This server allows you to manage all the users for your organization in one place: the LDAP directory. The LDAP V1 security realm supports Open LDAP, Netscape iPlanet, Microsoft Site Server, and Novell NDS directory servers.

To use the LDAP V1 security realm instead of the File realm:

1. Expand the Compatibility Security-->Realms nodes.
2. Click the Configure a New LDAP Realm V1... link to display the name of the class that implements the LDAP V1 security realm.
3. Click Create.
4. Define attributes for the LDAP directory server and specify how users and groups are located in the LDAP V1 security realm. For more information:
 - [“Enabled Communication between the LDAP Server and WebLogic Server” on page 322-9](#)
 - [“Specifying How Users Are Located in the LDAP V1 Security Realm” on page 322-9](#)
 - [“Specifying How Groups Are Located in the LDAP V1 Security Realm” on page 322-10](#)
5. When you have finished defining all the attributes, reboot WebLogic Server.
6. Configure the Caching realm. For more information, see [“Configuring the Caching Realm” on page 322-5](#)

When configuring the Caching realm, select the LDAP Realm V1 option from the pull-down menu for the Basic Realm attribute on the General page. The Basic Realm attribute defines the association between the Caching realm and the alternate security realm (in this case, the LDAP V1 security realm).

7. Expand the Domains node.
8. Click the View Domain-Wide Security Settings link on the Domain-->General page.
9. Select the Compatibility-->File Realm tab.
10. In the Caching Realm attribute, choose the name of the Caching realm to be used with the LDAP V1 security realm. A list of configured Caching realms appears on the pull-down menu.
11. Reboot WebLogic Server.

Enabled Communication between the LDAP Server and WebLogic Server

To enable communication between the LDAP server and WebLogic Server:

1. Click the LDAP Realm V1 tab.
2. Define values for the attributes on the [LDAP Security Realm-->LDAP Server](#) page.
3. Click Apply to save your changes.

Specifying How Users Are Located in the LDAP V1 Security Realm

To specify how users are located in the LDAP V1 security realm:

1. Click the Users tab under the LDAP Realm V1 tab.
2. Define the attributes shown on the [LDAP Security Realm-->Users](#) page.
3. Click Apply to save your changes.

Specifying How Groups Are Located in the LDAP V1 Security Realm

To specify how groups are located in the LDAP V1 security realm:

1. Click on the Groups tab under the LDAP Realm V1 tab.
2. Define the attributes shown on the [LDAP Security Realm-->Groups](#) page.
3. Click Apply to save your changes.

Adding a Note to the LDAP V1 Security Realm

To add a note to the LDAP V1 security realm:

1. Click on the Notes tab under the LDAP Realm V1 tab.
2. Write any pertinent information in the Notes field.
3. Click Apply to save your changes.

Configuring an LDAP Realm V2

In Compatibility security, the LDAP realm V2 is configured as a custom security realm.

The LDAP tree and schema is different for every LDAP server. The “[Supported Server Templates](#)” on page 322-13 has templates for the supported LDAP servers. These templates specify default configuration information used to represent users and groups in each of the supported LDAP servers.

To use a LDAP realm V2:

1. Expand the Compatibility Security-->Realms nodes.
2. Click the Configure a new Custom Realm... link.
3. Set attributes on the Configuration tab.

-
4. The following table describes the attributes you set on the Custom Security Realm Configuration window.

Table 322-1 Custom Security Realm Attributes

Attribute	Description
Name	Name of the LDAP realm V2, such as defaultLDAPRealmForNetscapeDirectoryServer.

Table 322-1 Custom Security Realm Attributes

Attribute	Description
Realm Class Name	Name of the WebLogic class that implements the LDAP V2 realm such as <code>weblogic.security.ldaprealmv2.LDAPRealm</code> . This class needs to be in the CLASSPATH of WebLogic Server.
Configuration Data	<p>Specify information specific to your LDAP configuration for the following:</p> <p><code>server.host</code>—The host name of the LDAP server.</p> <p><code>server.port</code>—The port number on which the LDAP server listens.</p> <p><code>useSSL</code>—Specifies whether or not to use SSL to protect communications between the LDAP server and WebLogic Server. Set the value to <code>true</code> to enable the use of SSL.</p> <p><code>server.principal</code>—The LDAP user used by WebLogic Server to connect to the LDAP server.</p> <p><code>server.credential</code>—The password of the LDAP user user by WebLogic Server to connect to the LDAP server.</p> <p><code>user.dn</code>—The base DN of the tree in the LDAP directory that contains users.</p> <p><code>user.filter</code>—The LDAP search filter for finding a user given the name of the user.</p> <p><code>group.dn</code>—The base DN of the tree in the LDAP directory that contains groups.</p> <p><code>group.filter</code>—The LDAP search filter for finding a group given the name of the group.</p> <p><code>membership.filter</code>—The LDAP search filter for finding the members of a group given the name of the group.</p> <p>See “Supported Server Templates” on page 322-13 for sample values for the supported LDAP servers.</p>

5. Click Apply to save your changes.

6. Configure the Caching realm as described in [“Configuring the Caching Realm” on page 322-5](#).

When configuring the Caching realm, select the LDAP realm V2 from the pull-down menu for the Basic Realm attribute on the General tab. The Basic Realm attribute defines the association between the Caching realm and the alternate security realm (in this case, the LDAP realm V2).

7. Expand the Domains node.
8. Click the View Domain-Wide Security Settings link on the Domain-->General page.
9. Select the Compatibility-->File Realm tab.
10. In the Caching Realm attribute, choose the name of the Caching realm to be used with the LDAP V2 security realm. A list of configured Caching realms appears on the pull-down menu.
11. Reboot WebLogic Server.

Supported Server Templates

Listing 322-1 through Listing 322-1 are templates used to configure LDAP servers supported in the LDAP realm V2. Copy these templates directly into the `config.xml` file for your application.

Warning: Each line in the following code examples must appear on a single line. The examples in the code examples have been formatted to fit the margins of this document and some lines have been broken to facilitate that formatting. If you paste this text into the `config.xml` file, be sure to concatenate the lines that are broken so that they appear on a single line in your code.

Listing 322-1 Default Netscape Directory Server Template

```
<CustomRealmName="defaultLDAPRealmForNetscapeDirectoryServer"
RealmClassName="weblogic.security.ldaprealmv2.LDAPRealm"
ConfigurationData=
"server.host=ldapserver.example.com;
server.port=700;
useSSL=true;
```

```
server.principal=uid=admin,
ou=Administrators,ou=TopologyManagement,o=NetscapeRoot;
server.credential=*secret*;
user.dn=ou=people,o=beasys.com;
user.filter=(&uid=%u)(objectclass=person));
group.dn=ou=groups,o=beasys.com;
group.filter=(&cn=%g)(objectclass=groupofuniquenames));
membership.filter=(&(uniquemember=%M)
(objectclass=groupofuniquenames));
```

```
"Notes="Before enabling the LDAP V2 security realm, edit the
configuration parameters for your environment."/>
```

Listing 322-2 Default Microsoft Site Server Template

```
<CustomRealmName="defaultLDAPRealmForMicrosoftSiteServer"
RealmClassName="weblogic.security.ldaprealmv2.LDAPRealm"
ConfigurationData=
"server.host=ldapserver.example.com;
server.port=700;
useSSL=true;
server.principal=cn=Administrator,ou=Members,
o=ExampleMembershipDir;
server.credential=*secret*
user.dn=ou=Members, o=ExampleMembershipDir;
user.filter=(&(cn=%u)(objectclass=member)
(!userAccountControl:1.2.840.113556.1.4.803:=2));
group.dn=ou=Groups, o=ExampleMembershipDir;
group.filter=(&(cn=%g)(objectclass=mgroup));
membership.scope.depth=1;microsoft.membership.scope=sub;
membership.filter=(|(&(memberobject=%M)
(objectclass=memberof))(&(groupobject=%M)
(objectclass=groupmemberof)));
membership.search=true;
```

```
"Notes="Before enabling the LDAP V2 security realm, edit the
configuration parameters for your environment."/>
```

Listing 322-3 Default Novell Directory Services Template

```
<CustomRealmName="defaultLDAPRealmForNovellDirectoryServices"
RealmClassName="weblogic.security.ldaprealmv2.LDAPRealm"
```

```

ConfigurationData=
"server.host=ldapserver.example.com;
server.port=700;
useSSL=true;
server.principal=cn=Admin, DC=BEASYS
server.credential= *secret*;
user.dn=ou=people,o=example.com;
user.filter=(&!(cn=%u)(objectclass=person));
group.dn=ou=groups,o=example.com;
group.filter=(&!(cn=%g)(objectclass=groupofuniquenames));
membership.filter=(&!(member=%M)
(objectclass=groupofuniquenames));"

"Notes="Before enabling the LDAP V2 security realm, edit the
configuration parameters for your environment."/>

```

Listing 322-4 Default Open LDAP Directory Services Template

```

<CustomRealmName="defaultLDAPRealmForOpenLDAPDirectoryServices"
RealmClassName="weblogic.security.ldaprealmv2.LDAPRealm"
ConfigurationData=
"server.host=ldapserver.example.com;
server.port=700;
useSSL=true;
server.principal=cn=Manager, dc=example, dc=com;
server.credential= *secret*;
user.dn=ou=people, dc=example,dc=com;
user.filter=(&!(uid=%u)(objectclass=person));
group.dn=ou=groups,dc=example,c=com;
group.filter=(&!(cn=%g)(objectclass=groupofuniquenames));
membership.filter=(&!(uniquemember=%M)
(objectclass=groupofuniquenames));"

"Notes="Before enabling the LDAP V2 security realm, edit the
configuration parameters for your environment."/>

```

Adding a Note to the LDAP V2 Security Realm

To add a note to the LDAP V2 security realm:

1. Click on the Notes tab under the configuration window for the chosen LDAP server.
2. Write any pertinent information in the Notes field.
3. Click Apply to save your changes.

Configuring the Windows NT Security Realm

To configure the Windows NT security realm:

1. Expand the Compatibility Security-->Realms node.
2. Click the Configure a New NT Realm... link.
3. Set attributes on the [Windows NT Realm-->Configuration](#) page that define a name for the Windows NT realm and the computer on which the Windows NT domain is running.
4. Click Apply to save your changes.
5. Configure the Caching realm. For more information, see [“Configuring the Caching Realm” on page 322-5](#).

When configuring the Caching realm, select your Windows NT security realm from the pull-down menu for the Basic Realm attribute on the General page. The Basic Realm attribute defines the association between the Caching realm and the alternate security realm (in this case, the Windows NT security realm).

6. Expand the Domains node.
7. Click the View Domain-Wide Security Settings link on the Domain-->General page.
8. Click the Compatibility-->File Realm tab.
9. In the Caching Realm attribute, choose the name of the Caching realm to be used with the Windows NT security realm. A list of configured Caching realms appears on the pull-down menu.
10. Reboot WebLogic Server.

Use the following command to verify that you have the correct privileges to run WebLogic Server as the specified Windows NT user:

```
java weblogic.security.ntrealm.NTRrealm username password
```

where *username* and *password* are the username and password of the Windows NT account under which WebLogic Server runs.

The output from this command indicates if the specified username and password authenticated properly.

Table 322-2 Windows NT Authentication Verification

Command Output	Meaning
auth?poppy	The entered username and password authenticated correctly.
auth?null	The entered username and password did not authenticate properly.

If the test comes up with an immediate failure stating that the client or user running WebLogic Server does not have the privileges to run the Windows NT Security realm, you need to update the permissions (referred to as rights) for the Windows user running WebLogic Server. For more information, see [“Updating Users Permissions for Windows NT and Windows 2000” on page 322-17](#).

Updating Users Permissions for Windows NT and Windows 2000

To update the rights in Windows NT:

1. On the Start menu, select Programs—Administrative Tools.
2. Select User Manager.
3. Under the Policies menu, choose the User Rights option.
4. Check the Show Advanced Users Rights option.
5. Give the following rights to the Windows user running WebLogic Server:

6. Act as part of the operating system
7. Create a token object
8. Replace a process level token
9. Verify that the Windows user running WebLogic Server is a member of the Administrators group.
10. Reboot Windows NT to ensure all the modifications take effect.
11. Verify that the Logon as System Account option is checked. Note that the Allow System to Interact with Desktop option does not need to be checked. Running the Windows NT Security realm under a specific Windows NT user account does not work.

To update the rights in Windows 2000:

1. On the Start menu, select Programs—~~Administrative Tools~~.
2. Select Local Security Policy.
3. Go to Local Policies—~~User Rights Assignment~~.
4. Give the following rights to the Windows user running WebLogic Server:
 - Act as part of the operating system
 - Create a token object
 - Replace a process level token
5. Verify that the Windows user running WebLogic Server is a member of the Administrators group.
6. Reboot Windows 2000 to ensure all the modifications take effect.
7. Verify that the Logon as System Account option is checked. Note that the Allow System to Interact with Desktop option does not need to be checked. Running the Windows NT Security realm under a specific Windows NT user account does not work.

The following are common Windows NT error codes that occur when using the Windows NT Security realm:

Table 322-3 Windows NT Error Codes

Error Code	Meaning
1326	The host machine running the security realm does not have a trust relationship with the primary domain controller. The host machine may not be a member of the domain or the domain may not trust the host machine.
53	A network error has indicates that the path to the primary domain controller could not be located. This error can occur if the domain name is misspelled or if the domain name is specified rather than the host name of the primary domain controller.

A full explanation of the Windows NT error codes is found in the `winerror.h` file.

Adding a Note to the Windows NT Security Realm

To add a note to the Windows NT security realm:

1. Click on the [Windows NT Realm-->Notes](#) tab under the Configuration tab.
2. Write any pertinent information in the Notes field.
3. Click Apply to save your changes.

Configuring the wlauth Program for the UNIX Security Realm

The `wlauth` program runs `setuid root`. You need root permissions to modify the ownership and file attributes on the `wlauth` program and to set up the PAM configuration file for `wlauth`.

To set up the `wlauth` program for the UNIX security realm:

1. If WebLogic Server is installed on a network drive, copy the `wlauth` file to a file system on the computer that executes WebLogic Server, for example, the `/usr/sbin` directory. The `wlauth` file is in the `weblogic/lib/arch` directory, where *arch* is the name of your platform.
2. As the root user, run the following commands to change the `wlauth` owner and permissions:

```
# chown root wlauth
# chmod +xs wlauth
```

3. Set up the PAM configuration for `wlauth`.

Solaris—Add the following lines to your `/etc/pam.conf` file:

```
# Setup for WebLogic authentication on Solaris machines
#
wlauth auth required      /usr/lib/security/pam_unix.so.1
wlauth password required  /usr/lib/security/pam_unix.so.1
wlauth account required   /usr/lib/security/pam_unix.so.1
```

Linux—Create a file called `/etc/pam.d/wlauth` containing the following:

```
##PAM-1.0
#
# File name:
# /etc/pam.d/wlauth
#
# If you do not use shadow passwords, delete "shadow".
auth required      /lib/security/pam_pwdb.so shadow
account required   /lib/security/pam_pwdb.so
```

Note: Omit `shadow` if you are not using shadow passwords.

If `wlauth` is not in the WebLogic Server class path or if you have given the program a name other than `wlauth`, you must add a Java command-line property when you start WebLogic Server. Edit the script you use to start WebLogic Server and add the following option after the `java` command:

```
-Dweblogic.security.unixrealm.authProgram=wlauth_prog
```

Replace *wlauth_prog* with the name of the `wlauth` program, including the full path if the program is not in the search path. Start WebLogic Server. If the `wlauth` program is in the WebLogic Server path and is named `wlauth`, this step is not needed.

Configuring the UNIX Security Realm

Note: The UNIX Security realm runs only on the Solaris and Linux platforms.

To configure the Unix security realm:

1. Expand the Compatibility Security-->Realms nodes.
2. Click the Configure a New Unix Realm... link.
3. Set attributes on the [Unix Realm-->Configuration](#) page that define a name for the realm and the program that provides authentication services for the UNIX Security realm.
4. Click Create.
5. Configure the Caching realm. For more information, see [“Configuring the Caching Realm” on page 322-5](#).

When configuring the Caching realm, select your UNIX security realm from the pull-down menu for the Basic Realm attribute on the General page. The Basic Realm attribute defines the association between the Caching realm and the alternate security realm (in this case, the UNIX security realm).

6. Expand the Domains node.
7. Click the View Domain-Wide Security Settings link on the Domain-->General page.
8. Click the Compatibility-->File Realm tab.
9. In the Caching Realm attribute, choose the name of the Caching realm to be used with the UNIX security realm. A list of configured Caching realms appears on the pull-down menu.
10. Reboot WebLogic Server.

Adding a Note to the UNIX Security Realm

1. Click on the [Unix Realm-->Notes](#) tab under the Configuration tab.
2. Write any pertinent information in the Notes field.

3. Click Apply to save your changes.

Configuring the RDBMS Security Realm

The RDBMS security realm is a BEA-provided custom security realm that stores users, groups and ACLs in a relational database. The RDBMS security realm is an example and is not meant to be used in a production environment.

Notes: The RDBMS example does not work with databases that have an autocommit feature enabled. If you use the RDBMS example as a starting point for your RDBMS implementation, use explicit commit statements in your code and make sure the autocommit feature in the database you are using is disabled.

If your implementation of the RDBMS security realm uses the `getActiveDomain()` method, you need to edit and recompile your `RDBMSDelegate` class in order to use the RDBMS security realm with Compatibility security. Replace the `getActiveDomain()` method with the `getSecurityConfig()` method in the `weblogic.server` package.

To configure an RDBMS security realm:

1. Expand the Compatibility Security-->Realms node.
2. Choose the database you want to use with WebLogic Server. The following templates are available:
 - defaultRDBMSRealmForOracle
 - defaultRDBMSRealmForMSSQLServerType4
 - defaultRDBMSRealmForCloudScape
 - defaultRDBMSRealmForODBC

A configuration window for the chosen database appears.

3. Set attributes on the [RDBMS Realm-->General](#) page that define a name for the realm and the class that implements the RDBMS security realm.
4. Click Create.
5. Define attributes for connecting to the database and the database schema. For more information:

- [“Defining Database Attributes for the RDBMS Security Realm” on page 322-23](#)
 - [“Defining Database Schema for the RDBMS Security Realm” on page 322-24](#)
6. Configure the Caching realm. For more information, see [“Configuring the Caching Realm” on page 322-5](#).

When configuring the Caching realm, select the RDBMS security realm from the pull-down menu for the Basic Realm attribute on the General page. The Basic Realm attribute defines the association between the Caching realm and the alternate security realm (in this case, the RDBMS security realm).

7. Expand the Domains node.
8. Click the View Domain-Wide Security Settings link on the Domain-->General page.
9. Click the Compatibility-->File Realm tab.
10. In the Caching Realm attribute, choose the name of the Caching realm to be used with the RDBMS security realm. A list of configured Caching realms appears on the pull-down menu.
11. Reboot WebLogic Server.

Defining Database Attributes for the RDBMS Security Realm

To define attributes for the JDBC driver that connects to the database in the RDBMS security realm:

1. Click the [RDBMS Realm-->Database](#) tab.
2. Define attributes for the JDBC driver being used to connect to the database.
3. Click Apply to save your changes.

Defining Database Schema for the RDBMS Security Realm

To define attribute for the database schema used by the RDBMS security realm:

1. Click the [RDBMS Realm-->Schema](#) tab.
2. Define the schema used to store Users, Groups, and ACLs in the database in the Schema Properties box on the Schema page.

Listing 322-1 contains the database statements entered in the Schema properties for the RDBMS code example shipped with WebLogic Server in the `/samples/examples/security/rdbmsrealm` directory.

Listing 322-1 Sample Schema for RDBMS Security Realm

```
"getGroupNewStatement=true;getUser=SELECT U_NAME, U_PASSWORD FROM
users WHERE U_NAME = ?;
getGroupMembers=SELECT GM_GROUP, GM_MEMBER from groupmembers WHERE
GM_GROUP = ?;
getAclEntries=SELECT A_NAME, A_PRINCIPAL, A_PERMISSION FROM
aclentries WHERE A_NAME = ? ORDER BY A_PRINCIPAL;
getUsers=SELECT U_NAME, U_PASSWORD FROM users;
getGroups=SELECT GM_GROUP, GM_MEMBER FROM groupmembers;
getAcls=SELECT A_NAME, A_PRINCIPAL, A_PERMISSION FROM aclentries
ORDER BY A_NAME, A_PRINCIPAL;
getPermissions=SELECT DISTINCT A_PERMISSION FROM aclentries;
getPermission=SELECT DISTINCT A_PERMISSION FROM aclentries WHERE
A_PERMISSION = ?;
newUser=INSERT INTO users VALUES ( ? , ? );
addGroupMember=INSERT INTO groupmembers VALUES ( ? , ? );
removeGroupMember=DELETE FROM groupmembers WHERE GM_GROUP = ? AND
GM_MEMBER = ?;
deleteUser1=DELETE FROM users WHERE U_NAME = ?;
deleteUser2=DELETE FROM groupmembers WHERE GM_MEMBER = ?;
deleteUser3=DELETE FROM aclentries WHERE A_PRINCIPAL = ?;
deleteGroup1=DELETE FROM groupmembers WHERE GM_GROUP = ?;
deleteGroup2=DELETE FROM aclentries WHERE A_PRINCIPAL = ?"
```

3. Click Apply to save your changes.

Adding A Note to the RDBMS Security Realm

To add a note to the RDBMS security realm:

1. Click on the [RDBMS Realm-->Notes](#) tab under the Configuration tab.
2. Write any pertinent information in the Notes field.
3. Click Apply to save your changes.

Installing a Custom Security Realm

You can create a custom security realm that draws from an existing store of users such as directory server on the network. To use a custom security realm, you create an implementation of the `weblogic.security.acl.AbstractListableRealm` interface or the `weblogic.security.acl.AbstractManageableRealm` interface and then use the Administration Console to install your implementation.

To install a custom security realm:

1. Expand the Compatibility Security-->Realms node.
2. Click the Configure a New Custom Realm... link.
3. Set attributes on the [Custom Realm --> Configuration](#) page that define a name for the custom security realm, specify the interface that implements the realm, and define how the users, groups, and optionally ACLs are stored in the custom security realm.
4. Click Create.
5. Configure the Caching realm. For more information, see [“Configuring the Caching Realm” on page 322-5](#).

When configuring the Caching realm, select the custom security realm from the pull-down menu for the Basic Realm attribute on the General page. The Basic Realm attribute defines the association between the Caching realm and the custom security realm.

6. Expand the Domains node.

7. Click the View Domain-Wide Security Settings link on the Domain-->General page.
8. Click the Compatibility-->File Realm tab.
9. In the Caching Realm attribute, choose the name of the Caching realm to be used with the custom security realm. A list of configured Caching realms appears on the pull-down menu.
10. Reboot WebLogic Server.

Adding A Note To A Custom Security Realm

To add a note to a custom security realm:

1. Click on the [Custom Realm --> Notes](#) tab under the Configuration tab.
2. Write any pertinent information in the Notes field.
3. Click Apply to save your changes.

Defining Users

Note: This section explains how to add users to a manageable security realm (for example, the File realm) in the *CompatibilityRealm*. If you are using a security realm that is not manageable through the WebLogic Server Administration Console, you must use the administration tools provided in that realm to define a user.

To define a user:

1. Expand the Compatibility Security node.
2. Click Users.
3. In the User Configuration window, enter the name of the user in the Name attribute.
4. Enter a password for the user in the Password attribute.

5. Enter the password again in the Confirm Password attribute.
6. Click Create.

Deleting Users

To delete a user:

1. Expand the Compatibility Security node.
2. Click Users.
3. In the User Configuration window, enter the name of the user in the Delete Users box.
4. Click Delete.

Changing the Password of a User

1. Expand the Compatibility Security node.
2. Click Users.
The User Configuration window appears.
3. Enter the name of the user in the Name attribute on the User Configuration window.
4. Enter the old password in the Old Password attribute.
5. Enter the new password in the New Password attribute.
6. Enter the new password again to confirm the password change.

Unlocking A User Account

To unlock a user account:

1. Expand the Compatibility Security node.

2. Click Users.
3. In the User Configuration window, click the Unlock Users link.
4. Enter the names of the user accounts you want to unlock in the Users to Unlock field.
5. Choose the servers on which you want the user accounts unlocked.
6. Click Unlock.

Disabling the Guest User

For a more secure deployment, BEA recommends running WebLogic Server with the `guest` account disabled.

To disable the Guest user:

1. Expand the Domains node.
2. Click the View Domain-Wide Security Settings link on the Domain-->General page.
3. Click the Compatibility-->General tab.
4. Check the Guest Disable checkbox.
5. Reboot WebLogic Server.

Disabling the `guest` account just disables the ability to log in into the account `guest`; it does not disable the ability for unauthenticated users to access a WebLogic Server deployment.

Defining Groups

Note: This section explains how to add groups to a manageable security realm (for example, the File realm) in the *CompatibilityRealm*. If you are using a security realm that is not manageable through the WebLogic Server Administration Console, you must use the administration tools provided in that realm to define a group.

To define a group in the Compatibility realm:

1. Expand the Compatibility Security node.
2. Click Groups.
3. Click the Create a New Group... link.
4. In the Groups window, enter the name of the group in the Name attribute. BEA recommends naming groups in the plural. For example, Administrators instead of Administrator.
5. Click the Users attribute and select the WebLogic Server users you want to add to the group.
6. Click the Groups attribute and select the WebLogic Server groups you want to add to the group.
7. Click Apply to create a new Group.

Removing Users from a Group

To remove a user from a group:

1. Expand the Compatibility Security node.
2. Click Groups.
3. Select the group from which you want to delete a user.
4. In the Groups window, check the users you want to remove from the group.
5. Click Apply.

Deleting Groups

To delete a groups:

1. Expand the Compatibility Security node.
2. Click Groups.

The Groups table appears. This table displays the names of all groups defined in the Compatibility realm.

3. To delete a group, enter the name of the group in the Remove These Groups list box.
4. Click Remove.

Defining ACLs

Compatibility security provides backward compatibility for ACLs and should not be considered a long-term security solution. The steps in this section should only be used if you corrupt an existing 6.x security realm and you have no choice but to restore it. Instead of ACLs, use security roles and security policies to protect WebLogic resources.

Note: ACLs on MBeans are not supported in this release of WebLogic Server. For more information, see "[Layered Security Scheme for Server Resources](#)" in *Securing WebLogic Resources*.

When you specify an ACL for a JDBC connection pool, you must specifically define access to the JDBC connection pool for the `system` user in the `filerealm.properties` file. For example:

```
acl.reserve.poolforsecurity=system
acl.reset.poolforsecurity=system
```

To create ACLs for WebLogic resources:

1. Expand the Compatibility Security node.
2. Click the ACLs tab.
3. Click the Create a New ACL... link.
4. In the ACL Configuration window in the New ACL Name attribute, specify the name of WebLogic Server resource that you want to protect with an ACL.

For example, create an ACL for a JDBC connection pool named `demopool1`.

5. Click Create.
6. Click on the Add a New Permission link.

7. Specify a permission for the resource.

Either create separate ACLs for each permission available for a resource or one ACL that grants all the permissions for a resource. For example, you can create three ACLs for the JDBC connection pool, `demopool`: one with `reserve` permission, one with `reset` permission, and one with `shrink` permission. Or you can create one ACL with `reserve`, `reset`, and `shrink` permissions.

8. Specify Weblogic users or groups that have the specified permission to the resource.
9. Click Apply.

Protecting User Accounts

To protect user accounts in your WebLogic Server domain:

1. Expand the Domains node.
2. Click the View Domain-Wide Security Settings link on the Domain-->General tab.
3. Click the Compatibility-->Passwords tabs.
4. Set attributes on the page by entering values at the appropriate prompts and selecting the required checkboxes.
5. Click Apply.
6. Reboot WebLogic Server.

Installing an Audit Provider

If your WebLogic Server 6.x security configuration uses an implementation of the `weblogic.security.audit.AuditProvider` class, the Auditor is not automatically configured in Compatibility security. Configure a Realm Adapter Auditing provider in the Compatibility realm to access the 6.x Auditor.

To configure a Realm Adapter Auditing provider:

1. Start WebLogic Server.
2. Start the admin command line tool
3. Enter the following commands:

```
java weblogic.Admin -url t3://localhost:7001 -username  
adminusername -password adminpassword CREATE -mbean Security:  
Name=CompatibilityRealmRealmAdapterAuditor -type  
weblogic.security.providers.realmadapter.RealmAdapterAuditor  
commotype
```

```
java weblogic.Admin -url t3://localhost:7001 -username  
adminusername -password adminpassword SET -mbean Security:  
Name=CompatibilityRealmRealmAdapterAuditor -property Realm  
Security:Name=CompatibilityRealm commotype
```

```
java weblogic.Admin -url t3://localhost:7001 -username  
adminusername -password adminpassword SET -mbean Security  
Name=CompatibilityRealm -property Auditors  
Security:Name=CompatibilityRealmRealmAdapterAuditor commotype
```

4. Reboot WebLogic Server.

Attributes and Console Screen Reference for Compatibility Security

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“Domain-->Compatibility Security-->General” on page 365-1](#)

[“Domain -->Compatibility Security-->File Realm” on page 366-1](#)

[“Domain-->Compatibility Security-->Advanced” on page 367-1](#)

[“Domain-->Compatibility Security-->Passwords” on page 364-1](#)

[“ACL” on page 293-1](#)

[“Create a New ACL” on page 291-1](#)

[“ACL Permission” on page 292-1](#)

[“Basic Realm” on page 317-1](#)

[“Caching Realm” on page 301-1](#)

[“Caching Realm-->ACL” on page 294-1](#)

[“Caching Realm --> Authentication” on page 295-1](#)

[“Caching Realm --> General” on page 296-1](#)

[“Caching Realm --> Groups” on page 297-1](#)

[“Caching Realm --> Permissions” on page 298-1](#)

[“Caching Realm --> Users” on page 299-1](#)

[“Caching Realm --> Notes” on page 300-1](#)

[“Custom Realm --> Configuration” on page 302-1](#)

[“Custom Realm --> Notes” on page 303-1](#)

[“Groups” on page 304-1](#)

“Group-->Group” on page 305-1

“LDAP Security Realm-->Configuration-->General” on page 306-1

“LDAP Security Realm-->Groups” on page 307-1

“LDAP Security Realm-->LDAP Server” on page 308-1

“LDAP Security Realm-->Users” on page 309-1

“RDBMS Realm-->General” on page 314-1

“RDBMS Realm-->Database” on page 313-1

“RDBMS Realm-->Schema” on page 315-1

“RDBMS Realm-->Notes” on page 316-1

“Realm Adapter Adjudication Provider-->General” on page 406-1

“Realm Adapter Adjudication Provider-->Details” on page 407-1

“Realm Adapter Authentication Provider-->General” on page 409-1

“Realm Adapter Authentication Provider-->Details” on page 408-1

“Realm Adapter Authorization Provider-->General” on page 410-1

“Realm Adapter Authorization Provider-->Details” on page 411-1

“Unix Realm-->Configuration” on page 318-1

“Unix Realm-->Notes” on page 319-1

“Users” on page 321-1

“Windows NT Realm-->Configuration” on page 311-1

“Windows NT Realm-->Notes” on page 312-1

Active Directory Authentication Provider-->Active Directory

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure the Active Directory LDAP server to enable communication between the LDAP server and WebLogic Server.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 324-1

Attribute Label	Description	Value Constraints
Host	The host name or IP address of the LDAP server. <i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean <i>Attribute:</i> Host	<i>Default:</i> "localhost"
Port	The port number on which the LDAP server is listening. <i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean <i>Attribute:</i> Port	<i>Default:</i> new java.lang.Integer(389)
SSLEnabled	The option to enable the use of the SSL protocol when connecting to the LDAP server. <i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean <i>Attribute:</i> SSLEnabled	<i>Default:</i> new java.lang.Boolean(false) <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false
Principal	Distinguished Name (DN) of the LDAP user used by WebLogic Server to connect to the LDAP server. <i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean <i>Attribute:</i> Principal	

Table 324-1

Attribute Label	Description	Value Constraints
Credential	<p>The credential (generally a password) used to authenticate the LDAP user defined in the Principal attribute.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> Credential</p>	<p><i>Encrypted:</i> yes</p>
Cache Enabled	<p>Enables the use of a cache with the LDAP server. This is a cache of the LDAP requests.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> CacheEnabled</p>	<p><i>Default:</i> new java.lang.Boolean(true)</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false
Cache Size	<p>The size of the cache in K.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> CacheSize</p>	<p><i>Default:</i> new java.lang.Integer(32)</p>
Cache TTL	<p>The time-to-live (TTL) of the cache in seconds.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> CacheTTL</p>	<p><i>Default:</i> new java.lang.Integer(60)</p>



Active Directory Authentication Provider-->General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure attributes for the Active Directory Authentication provider.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 325-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> Name</p>	
Description	<p>A short description of the LDAP Authentication provider.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> Description</p>	<p><i>Default:</i> "Provider that performs LDAP authentication"</p>
Version	<p>The version number of the LDAP Authentication provider.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> Version</p>	<p><i>Default:</i> "1.0"</p>

Table 325-1

Attribute Label	Description	Value Constraints
Control Flag	<p>Determines how the login sequence uses the Authentication provider.</p> <p>A REQUIRED value specifies this LoginModule must succeed. Even if it fails, authentication proceeds down the list of LoginModules for the configured Authentication providers.</p> <p>A REQUISITE value specifies this LoginModule must succeed. If other Authentication providers are configured and this LoginModule succeeds, authentication proceeds down the list of LoginModules. Otherwise, control is return to the application.</p> <p>A SUFFICIENT value specifies this LoginModule need not succeed. If it does succeed, return control to the application. If it fails and other Authentication providers are configured, authentication proceeds down the LoginModule list.</p> <p>An OPTIONAL value specifies this LoginModule need not succeed. Whether it succeeds or fails, authentication proceeds down the LoginModule list. This setting is the default.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> ControlFlag</p>	<i>Default:</i> "REQUIRED"



Active Directory Authentication Provider-->Groups

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to specify how groups are stored and located in the Active Directory LDAP directory.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 326-1

Attribute Label	Description	Value Constraints
Group Base DN	<p>The base distinguished name (DN) of the tree in the LDAP directory that contains groups.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> GroupBaseDN</p>	<p><i>Default:</i> "ou=ou=WLSGROUPS,dc=example,dc=com"</p>
Group Search Scope	<p>Specifies how deep in the LDAP directory tree to search for groups. Valid values are subtree and onelevel.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> GroupSearchScope</p>	<p><i>Default:</i> "subtree"</p>
Group From Name Filter	<p>LDAP search filter for finding a group given the name of the group. If the attribute is not specified (that is, if the attribute is null or empty), a default search filter is created based on the group schema.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> GroupFromNameFilter</p>	<p><i>Default:</i> "(&(cn=%g)(objectclass=groupofuniquenames))"</p>

Table 326-1

Attribute Label	Description	Value Constraints
All Groups Filter	<p>An LDAP search filter for finding all groups beneath the base group distinguished name (DN). If the attribute is not specified (that is, if the attribute is null or empty), a default search filter is created based on the Group schema.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> AllGroupsFilter</p>	
Static Group Object Class	<p>The name of the LDAP object class that stores static groups.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> StaticGroupObjectClass</p>	<i>Default:</i> "groupofuniqueNames"
Static Group Name Attribute	<p>The attribute of a static LDAP group object that specifies the name of the group.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> StaticGroupNameAttribute</p>	<i>Default:</i> "cn"



Active Directory Authentication Provider-->Membership

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to specify how groups are stored and located in the Active Directory.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 327-1

Attribute Label	Description	Value Constraints
Static Member DNAttribute	<p>The attribute of the LDAP static group object that specifies the distinguished names (DNs) of the members of the group.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> StaticMemberDNAttribute</p>	<p><i>Default:</i> "uniquemember"</p>
Static Group DN's from Member DNFilter	<p>LDAP search filter that, given the distinguished name (DN) of a member of a group, returns the DN's of the static LDAP groups that contain that member.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> StaticGroupDN'sfromMemberDNFilter</p>	
Dynamic Group Object Class	<p>The LDAP object class that stores dynamic groups.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> DynamicGroupObjectClass</p>	

Table 327-1

Attribute Label	Description	Value Constraints
Dynamic Group Name Attribute	The attribute of a dynamic LDAP group object that specifies the name of the group. <i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean <i>Attribute:</i> DynamicGroupNameAttribute	
Dynamic Member URLAttribute	The attribute of the dynamic LDAP group object that specifies the URLs of the members of the dynamic group. <i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean <i>Attribute:</i> DynamicMemberURLAttribute	



Active Directory Authentication Provider-->Users

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to specify how user information is stored in the Active Directory LDAP directory.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 328-1

Attribute Label	Description	Value Constraints
User Object Class	The LDAP object class that stores users. <i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean <i>Attribute:</i> UserObjectClass	<i>Default:</i> "person"
User Name Attribute	The attribute of the LDAP User object that specifies the name of the user. <i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean <i>Attribute:</i> UserNameAttribute	<i>Default:</i> "uid"
User Dynamic Group DNAttribute	The attribute of an LDAP user object that specifies the distinguished names (DNs) of dynamic groups to which this user belongs. If such an attribute does not exist, WebLogic Server determines if a user is a member of a group by evaluating the URLs on the dynamic group. If a group contains other groups, WebLogic Server evaluates the URLs on any of the descendents (indicates parent relationship) of the group. <i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean <i>Attribute:</i> UserDynamicGroupDNAttribute	

Table 328-1

Attribute Label	Description	Value Constraints
User Base DN	<p>The base distinguished name (DN) of the tree in the LDAP directory that contains users.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> UserBaseDN</p>	<p><i>Default:</i> "ou=people, o=example.com"</p>
User Search Scope	<p>Specifies how deep in the LDAP directory tree to search for Users. Valid values are subtree and onelevel.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> UserSearchScope</p>	<p><i>Default:</i> "subtree"</p>
User From Name Filter	<p>An LDAP search filter for finding a user given the name of the user. If the attribute (user name attribute and user object class) is not specified (that is, if the attribute is null or empty), a default search filter is created based on the user schema.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> UserFromNameFilter</p>	<p><i>Default:</i> "(&(cn=%u)(objectclass=user))"</p>

Table 328-1

Attribute Label	Description	Value Constraints
All Users Filter	<p>An LDAP search filter for finding all users beneath the base user distinguished name (DN). If the attribute (user object class) is not specified (that is, if the attribute is null or empty), a default search filter is created based on the user schema.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.ActiveDirectoryAuthenticatorMBean</p> <p><i>Attribute:</i> AllUsersFilter</p>	

Active Directory Authentication Provider-->Details

[Tasks](#) [Related Topics](#)

Overview

This page has additional attributes for the Active Directory Authentication provider.

- **Follow Referrals**—Specifies that a search for a user or group within the Active Directory Authentication provider will follow referrals to other LDAP servers or branches within the LDAP directory. By default, this attribute is enabled.
- **Bind Anonymously On Referrals**—By default, the Active Directory Authentication provider uses the same DN and password used to connect to the LDAP server when following referrals during a search. If you want to connect as an anonymous user, enable this attribute. Contact your LDAP system administrator for more information.
- **Results Time Limit**—The maximum number of milliseconds for the LDAP server to wait for results before timing out. If this attribute is set to 0, there is not maximum time limit. The default is 0.
- **Connect Timeout**—The maximum time in seconds to wait for the connection to the LDAP server to be established. If this attribute is set to 0, there is not a maximum time limit. The default is 0.
- **Parallel Connect Delay**—The delay in seconds when making concurrent attempts to attempt to multiple LDAP servers. If this attribute is set to 0, connection attempts are serialized. An attempt is made to connect to the first server in the list. The next entry in the list is tried only if the attempt to connect to the current host fails. If this attribute is not set and an LDAP server is unavailable, an application may be blocked for a long time. If this attribute is greater than 0, another connection is started after the specified time.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Adjudication Provider

This page displays the name of the Adjudication provider configured for the default security realm (for example, myrealm). Use the Replace with a new Default Adjudicator... link to configure a different Adjudication provider as the default.

- For more information, see [“Configuring the WebLogic Adjudication Provider” on page 428-34.](#)



Auditing Provider

This page displays the name of the Auditing provider configured in the default security realm. By default, the WebLogic Auditing provider is configured as the default Auditing provider. Use the [Configure a new Default Auditor...](#) link to configure a different auditing provider as the default.

- For more information, see “Configuring a WebLogic Auditing Provider” on page 428-34.



Authentication Providers

This page lists the names of the Authentication providers available for a security realm. By default, the WebLogic Authentication provider is configured as the default Authentication provider for the default security realm (`myrealm`). Choose from the [Configure...](#) links to configure new or additional Authentication providers for a security realm. To delete an Authentication provider, click the trash can icon in the corresponding row of the Authentication providers table.

The way you configure multiple Authentication providers can affect the overall outcome of the authentication process, which is especially important for multipart authentication (that is, an authentication sequence that uses more than one Authentication provider). Authentication providers are called in the order in which they are configured. To change the ordering of Authentication providers in a security realm, click the [Re-order the Configured Authentication Providers...](#) link. Be aware the way each Authentication provider's Control Flag attribute is set effects the outcome of the authentication process.

For more information, see:

- [“Configuring an Authentication Provider: Main Steps” on page 428-21](#)
- [“Setting the JAAS Control Flag” on page 428-23](#)
- [“Configuring a WebLogic Identity Assertion Provider” on page 428-32](#)



Authorization Provider

This page displays the name of the Authorization provider configured in the default security realm (for example, `myrealm`). By default, the WebLogic Authorization provider is configured as the default Authorization provider. Use the [Configure a new Default Authorizer...](#) link to configure a different Authorization provider as the default.

- For more information, see “Configuring the WebLogic Authorization Provider” on page 428-30.



Change Password

[Tasks](#) [Related Topics](#)

Overview

Use this page to change the password of a user.

Tasks

“Changing the Password of a User” on page 428-5

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation



Credential Mapping Provider

This page displays the name of the Credential Mapping provider configured in the default security realm (for example, `myrealm`). By default, the WebLogic Credential Mapping provider is configured as the default credential mapping provider. Use the [Configure a new Default Credential Mapper...](#) link to configure a different Credential Mapping provider as the default.

- For more information, see “Configuring the WebLogic Credential Mapping Provider” on page 428-31.



WebLogic Adjudication Provider-->General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure a WebLogic Adjudication provider for a security realm.

Note: The WebLogic Server Administration Console refers to the WebLogic Adjudication provider as the Default Adjudicator.

When multiple Authorization providers are configured in a security realm, each may return a different answer to the “is access allowed” question for a given resource. This answer may be `PERMIT`, `DENY`, or `ABSTAIN`. Determining what to do if multiple Authorization providers do not agree on the answer is the primary function of the Adjudication provider. Adjudication providers resolve authorization conflicts by weighting each Authorization provider’s answer and returning a final decision.

By default, the WebLogic Adjudication provider is configured. The WebLogic Adjudication provider behaves as follows:

- If all Authorization providers return `PERMIT`, then `PERMIT`.
- If any Authorization providers return `DENY`, then `DENY`.
- If some Authorization providers return `ABSTAIN` and others return `PERMIT`, then `PERMIT` if unanimous permit is not required, `DENY` otherwise.

You can use a custom Adjudication provider instead of the WebLogic Adjudication provider. For a custom Adjudication provider to be available in the WebLogic Server Administration Console, the MBean JAR file for the provider must be in the `WL_HOME\lib\mbeantypes` directory.

Tasks

“Configuring a New Security Realm” on page 428-18

[“Configuring the WebLogic Adjudication Provider” on page 428-34](#)

[“Configuring a Custom Security Provider” on page 428-35](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 336-1

Attribute Label	Description	Value Constraints
Name	The name of this WebLogic Adjudication provider.	
Description	A short description of this WebLogic Adjudication provider. <i>MBean:</i> weblogic.security.providers.authorization.DefaultAdjudicator <i>Attribute:</i> Description	

Table 336-1

Attribute Label	Description	Value Constraints
Version	The version of this WebLogic Adjudication provider. <i>MBean:</i> weblogic.security. providers.authorization. DefaultAdjudicator <i>Attribute:</i> Version	



WebLogic Adjudication Provider-->Details

When multiple Authorization providers are configured in a security realm, each may return a different answer to the “is access allowed” question for a given resource. This answer may be `PERMIT`, `DENY`, or `ABSTAIN`. The WebLogic Adjudication provider behaves as follows:

- If all Authorization providers return `PERMIT`, then `PERMIT`.
- If any Authorization providers return `DENY`, then `DENY`.
- If some Authorization providers return `ABSTAIN` and others return `PERMIT`, then `PERMIT` if unanimous permit is not required, `DENY` otherwise.

Set the Require Unanimous Permit attribute to specify that all Authorization providers are required to vote `PERMIT` in order for this WebLogic Adjudication provider to vote `PERMIT`.



WebLogic Auditing Provider-->General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure a WebLogic Auditing provider for a security realm.

Note: The WebLogic Server Administration Console refers to the WebLogic Auditing provider as the Default Auditor.

Auditing is the process whereby information about operating requests and the outcome of those requests are collected, stored, and disturbed for the purposes of nonrepudiation. In other words, auditing provides an electronic trail of computer activity. In the WebLogic Server security architecture, an Auditing provider is used to provide auditing services.

If configured, the WebLogic Server Security Framework will call through to an Auditing provider before and after security operations (such as authentication or authorization) have been performed, enabling audit event recording. The decision to audit a particular event is made by the Auditing provider itself and can be based on specific audit criteria and/or severity levels. The records containing the audit information may be written to output repositories such as an LDAP back-end, database, and a simple file.

All auditing information recorded by the WebLogic Auditing provider is saved in `WL_HOME\mydomain\myserver\DefaultAuditRecorder.log`. Auditing is configured for the security realm but each server instance writes to its own file.

You can use a custom Auditing provider instead of the WebLogic Auditing provider. For a custom Auditing provider to be available through the WebLogic Server Administration Console, the MBean JAR file for the provider must be in the `WL_HOME\lib\mbeantypes` directory.

Tasks

“Configuring a WebLogic Auditing Provider” on page 428-34

Related Topics

- [Introduction to WebLogic Security](#)
- [Managing WebLogic Security](#)
- [Securing WebLogic Resources](#)
- [Programming WebLogic Security](#)
- [Developing Security Providers for WebLogic Server](#)
- [Securing a Production Environment](#)
- The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)
- [Security FAQ](#)
- The [Security](#) page in the WebLogic Server documentation

Attributes

Table 338-1

Attribute Label	Description	Value Constraints
Name	The name of this WebLogic Auditing provider. <i>MBean:</i> weblogic.security.providers.audit.DefaultAuditorMBean <i>Attribute:</i> Name	

Table 338-1

Attribute Label	Description	Value Constraints
Description	A short description of this WebLogic Auditing provider. <i>MBean:</i> weblogic.security.providers.audit.DefaultAuditorMBean <i>Attribute:</i> Description	<i>Default:</i> "WebLogic Auditing Provider"
Version	The version of this WebLogic Auditing provider. <i>MBean:</i> weblogic.security.providers.audit.DefaultAuditorMBean <i>Attribute:</i> Version	<i>Default:</i> "1.0"



WebLogic Auditing Provider-->Details

Set the Severity attribute to determine the severity level at which auditing is initiated by this WebLogic Auditing provider.



WebLogic Authentication Provider-->Details

Set the Minimum Password Length attribute to specify the minimum number of characters required in a password processed by this WebLogic Authentication provider. This password is the password used to define users in the embedded LDAP server used by the WebLogic Authentication provider to store user and group information.



Weblogic Authentication Provider-->General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure a WebLogic Authentication provider for a security realm.

Authentication is the process whereby the identity of users or system processes are proved or verified. Authentication also involves remembering, transporting, and making identity information available to various components of a system when that information is needed.

By default, the WebLogic Authentication provider is configured in the default security realm (`myrealm`). The WebLogic Authentication provider allows you to edit, list, and manage users, groups, and group membership. User and group information is stored in the embedded LDAP server.

The Control Flag attribute is the Java Authentication and Authorization Service (JAAS) control flag that determines how the LoginModule for the WebLogic Authentication provider is used in the login sequence

You can use a custom Authentication provider instead of the WebLogic Authentication provider. For a custom Authentication provider to be available in the WebLogic Server Administration Console, the MBean JAR file for the provider must be in the `WL_HOME\lib\mbeantypes` directory.

Tasks

“Configuring an Authentication Provider: Main Steps” on page 428-21

“Configuring the WebLogic Authentication Provider” on page 428-24

Related Topics

- [Introduction to WebLogic Security](#)
- [Managing WebLogic Security](#)
- [Securing WebLogic Resources](#)
- [Programming WebLogic Security](#)
- [Developing Security Providers for WebLogic Server](#)
- [Securing a Production Environment](#)
- The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)
- [Security FAQ](#)
- The [Security page](#) in the WebLogic Server documentation

Attributes

Table 341-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this WebLogic Authentication provider.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.DefaultAuthenticatorMBean</p> <p><i>Attribute:</i> Name</p>	
Description	<p>A short description of this WebLogic Authentication provider.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.DefaultAuthenticatorMBean</p> <p><i>Attribute:</i> Description</p>	<p><i>Default:</i> "WebLogic Authentication Provider"</p>

Table 341-1

Attribute Label	Description	Value Constraints
Version	<p>The version number of this WebLogic Authentication provider.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.DefaultAuthenticatorMBean</p> <p><i>Attribute:</i> Version</p>	<i>Default:</i> "1.0"
Control Flag	<p>Determines how the login sequence uses the Authentication provider.</p> <p>A REQUIRED value specifies this LoginModule must succeed. Even if it fails, authentication proceeds down the list of LoginModules for the configured Authentication providers.</p> <p>A REQUISITE value specifies this LoginModule must succeed. If other Authentication providers are configured and this LoginModule succeeds, authentication proceeds down the list of LoginModules. Otherwise, control is return to the application.</p> <p>A SUFFICIENT value specifies this LoginModule need not succeed. If it does succeed, return control to the application. If it fails and other Authentication providers are configured, authentication proceeds down the LoginModule list.</p> <p>An OPTIONAL value specifies this LoginModule need not succeed. Whether it succeeds or fails, authentication proceeds down the LoginModule list. This setting is the default.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.DefaultAuthenticatorMBean</p> <p><i>Attribute:</i> ControlFlag</p>	<i>Default:</i> "REQUIRED"



WebLogic Authentication Provider-->Export

[Tasks](#) [Related Topics](#)

Overview

Use this page to export user and group data from a WebLogic Authentication provider. Use the Import page to then import the user and group data into a WebLogic Authentication provider in another security realm.

Tasks

“Importing and Exporting Security Data from Security Providers” on page 428-40

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security](#) page in the WebLogic Server documentation



WebLogic Authentication Provider-->Import

[Tasks](#) [Related Topics](#)

Overview

Use this page to import user and group data from a WebLogic Authentication provider in one security realm into a WebLogic Authentication provider in another security realm.

Tasks

“Importing and Exporting Security Data from Security Providers” on page 428-40

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security](#) page in the WebLogic Server documentation



WebLogic Authorization Provider-->Details

Authorization providers have the choice of implementing the `DeployableAuthorizationProvider` Security Service Provider Interface (SSPI) or the `AuthorizationProvider` SSPI. Authorization providers that support deploying security policies on behalf of a Web application or Enterprise JavaBean (EJB) need to implement the `DeployableAuthorizationProvider` SSPI and enable the Policy Deployment Enabled attribute on this page. The Policy Deployment Enabled attribute is enabled by default for the WebLogic Authorization provider.



Weblogic Authorization Provider-->General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure a WebLogic Authorization provider for a security realm.

Note: The Administration Console refers to the WebLogic Authorization provider as the Default Authorizer.

Authorization is the process whereby the interactions between users and resources are limited to ensure integrity, confidentiality, and availability. In other words, authorization is responsible for controlling access to resources based on user identity or other information. In the WebLogic Server Security architecture, an Authorization provider is used to provide authorization services.

By default, the WebLogic Authorization provider is configured in the default security realm (`myrealm`). You can use a custom Authorization provider instead of the WebLogic Authorization provider. For a custom Authorization provider to be available through the WebLogic Server Administration Console, the MBean JAR file for the provider must be in the `WL_HOME\lib\mbeantypes` directory.

Tasks

“Configuring a New Security Realm” on page 428-18

“Configuring the WebLogic Authorization Provider” on page 428-30

“Configuring a Custom Security Provider” on page 428-35

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 345-1

Attribute Label	Description	Value Constraints
Name	The name of this WebLogic Authorization provider. <i>MBean:</i> weblogic.security.providers.authorization.DefaultAuthorizerMBean <i>Attribute:</i> Name	
Description	A short description of this WebLogic Authorization provider. <i>MBean:</i> weblogic.security.providers.authorization.DefaultAuthorizerMBean <i>Attribute:</i> Description	<i>Default:</i> "Weblogic Authorization Provider"

Table 345-1

Attribute Label	Description	Value Constraints
Version	The version number of this WebLogic Authorization provider. <i>MBean:</i> weblogic.security.providers.authorization.DefaultAuthorizerMBean <i>Attribute:</i> Version	<i>Default:</i> "1.0"
Policy Deployment Enabled	Specifies whether this WebLogic Authorization provider stores security policies that are created while deploying a Web application or an Enterprise JavaBean (EJB).	<i>Default:</i> Enabled



WebLogic Authorization Provider-->Export

[Tasks](#) [Related Topics](#)

Overview

Use this page to export security policies from a WebLogic Authorization provider. Use the Import page to then import the security policies into a WebLogic Authorization provider in another security realm.

Tasks

“Importing and Exporting Security Data from Security Providers” on page 428-40

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security](#) page in the WebLogic Server documentation



WebLogic Authorization Provider-->Import

[Tasks](#) [Related Topics](#)

Overview

Use this page to import security policies from a WebLogic Authorization provider in one security realm into a WebLogic Authorization provider in another security realm.

Tasks

“Importing and Exporting Security Data from Security Providers” on page 428-40

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security](#) page in the WebLogic Server documentation



WebLogic Credential Mapping Provider-->Details

This page is not used in this release of WebLogic Server.



WebLogic Credential Mapping-->General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure a WebLogic Credential Mapping provider for a security realm.

Note: The WebLogic Server Administration Console refers to the WebLogic Credential Mapping provider as the Default Credential Mapper.

Credential mapping is the process whereby a remote system's authentication and authorization mechanisms are used to obtain an appropriate set of credentials to authenticate users to a target resource. In the WebLogic Server security architecture, a Credential Mapping provider is used to provide credential mapping services and bring new types of credentials into the WebLogic Server environment.

By default, the WebLogic Credential Mapping provider is configured in the default security realm (`myrealm`). You can use a Custom Credential Mapping provider instead of the WebLogic Credential Mapping provider. For a Custom Credential Mapping provider to be available in the WebLogic Server Administration Console, the MBean JAR file for the provider must be in the `WL_HOME\lib\mbeantypes` directory.

The Credential Mapping Deployment Enabled attribute specifies whether or not this Credential Mapping provider imports credential maps from the `weblogic-ra.xml` deployment descriptor.

In order to support the Credential Mapping Deployment Enabled attribute, a Credential Mapping provider must implement the `DeployableCredentialProvider` SSPI. By default, the WebLogic Credential Mapping provider has this attribute enabled. Therefore, information from a `weblogic-ra.xml` deployment descriptor file is automatically loaded into the WebLogic Credential Mapping provider when the resource adapter is deployed.

It is important to understand that once information from a `weblogic-ra.xml` deployment descriptor file is loaded into the embedded LDAP server, the original resource adapter remains unchanged. Therefore, if you redeploy the original resource

adapter (which will happen if you redeploy it through the WebLogic Server Administration Console, modify it on disk, or restart WebLogic Server), the data will once again be imported from the `weblogic-ra.xml` deployment descriptor file and credential mapping information may be lost.

To avoid overwriting new credential mapping information with old information in a `weblogic-ra.xml` deployment descriptor file, enable the Ignore Security Data in Deployment Descriptors attribute on the security realm.

Tasks

“Configuring a New Security Realm” on page 428-18

“Configuring the WebLogic Credential Mapping Provider” on page 428-31

“Configuring a Custom Security Provider” on page 428-35

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security](#) page in the WebLogic Server documentation

Attributes

Table 349-1

Attribute Label	Description	Value Constraints
Name	The name of this WebLogic Credential Mapping provider.configuration <i>MBean:</i> weblogic.security.providers.credentials.DefaultCredentialMapperMBean <i>Attribute:</i> Name	
Description	A short description of this WebLogic Credential Mapping provider. <i>MBean:</i> weblogic.security.providers.credentials.DefaultCredentialMapperMBean <i>Attribute:</i> Description	<i>Default:</i> "WebLogic Credential Mapping Provider"
Version	The version number of this WebLogic Credential Mapping provider. <i>MBean:</i> weblogic.security.providers.credentials.DefaultCredentialMapperMBean <i>Attribute:</i> Version	<i>Default:</i> "1.0"
Credential Mapping Deployment Enabled	Specifies whether this WebLogic Credential Mapping provider stores credential maps that are created while deploying a Resource Adapter (RA). <i>MBean:</i> weblogic.security.providers.credentials.DefaultCredentialMapperMBean <i>Attribute:</i> CredentialMappingDeploymentEnabled	<i>Default:</i> new java.lang.Boolean(true) <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false



WebLogic Credential Mapping Provider-->Export

[Tasks](#) [Related Topics](#)

Overview

Use this page to export credential maps from a WebLogic Credential Mapping provider. Use the Import page to then import the credential maps into a WebLogic Credential Mapping provider in another security realm.

Tasks

“Importing and Exporting Security Data from Security Providers” on page 428-40

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation



WebLogic Credential Mapping Provider-->Import

[Tasks](#) [Related Topics](#)

Overview

Use this page to import credential maps from a WebLogic Credential Mapping provider in one security realm into a WebLogic Credential Mapping provider in another security realm.

Tasks

“Importing and Exporting Security Data from Security Providers” on page 428-40

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation



Weblogic Identity Assertion Provider-->General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure a WebLogic Identity Assertion provider for a security realm.

Note: The WebLogic Server Administration Console refers to the WebLogic Identity Assertion provider as the Default Identity Asserter.

If you are using perimeter authentication, you need to use an Identity Assertion provider. In perimeter authentication, a system outside of WebLogic Server establishes trust via tokens (as opposed to simple authentication, where WebLogic Server establishes trust via usernames and passwords). An Identity Assertion provider verifies the tokens and performs whatever actions are necessary to establish validity and trust in the token. Each Identity Assertion provider is designed to support one or more token formats.

Multiple Identity Assertion providers can be configured in a security realm, but none are required. Identity Assertion providers can support more than one token type, but only one token type per Identity Assertion provider can be active at a given time. When using the WebLogic Identity Assertion provider, configure the active token type. The WebLogic Identity Assertion provider supports identity assertion using X509 certificates and CORBA Common Secure Interoperability version 2 (CSI v2).

You can use a custom Identity Assertion provider instead of the WebLogic Identity Assertion provider. For a custom Identity Assertion provider to be available in the WebLogic Server Administration Console, the MBean JAR file for the provider must be in the `WL_HOME\lib\mbeantypes` directory.

When using 2-way SSL, WebLogic Server verifies the digital certificate of the Web browser or Java client when establishing an SSL connection. However, the digital certificate does not identify the Web browser or Java client as a user in the WebLogic

Server security realm. If the Web browser or Java client requests a WebLogic Server resource protected by a security policy, WebLogic Server requires the Web browser or Java client to have an identity. The WebLogic Identity Assertion provider allows you to define a user name mapper that maps the digital certificate of a client to a user in a WebLogic Server security realm.

This user name mapper is a class that implements the `weblogic.security.providers.authentication.UserNameMapper` interface. You can either write your own implementation and configure it in the Administration Console or use the default implementation provided by WebLogic Server.

- Specify customer implementations of the `weblogic.security.providers.authentication.UserNameMapper` interface on this page.
- Use the Details page to enable the use of the default user name mapper and to configure attributes for that user name mapper.

Tasks

“Configuring an Authentication Provider: Main Steps” on page 428-21

“Configuring a WebLogic Identity Assertion Provider” on page 428-32

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

Attributes

Table 352-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this WebLogic Identity Assertion provider.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.DefaultIdentityAsserterMBean</p> <p><i>Attribute:</i> Name</p>	
Description	<p>A short description of this WebLogic Identity Assertion provider.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.DefaultIdentityAsserterMBean</p> <p><i>Attribute:</i> Description</p>	<p><i>Default:</i> "WebLogic Identity Assertion provider"</p>
Version	<p>The version number of this WebLogic Identity Assertion provider.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.DefaultIdentityAsserterMBean</p> <p><i>Attribute:</i> Version</p>	<p><i>Default:</i> "1.0"</p>

Table 352-1

Attribute Label	Description	Value Constraints
User Name Mapper Class Name	<p>The name of the Java class that maps X.509 digital certificates and X.501 distinguished names to WebLogic user names.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.DefaultIdentityAsserterMBean</p> <p><i>Attribute:</i> UserNameMapperClassName</p>	
Trusted Client Principals	<p>The list of trusted client principals to use in CSIv2 identity assertion. The wildcard character (*) can be used to specify all principals are trusted. If a client is not listed as a trusted client principal, the CSIv2 identity assertion fails and the invoke is rejected.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.DefaultIdentityAsserterMBean</p> <p><i>Attribute:</i> TrustedClientPrincipals</p>	

Table 352-1

Attribute Label	Description	Value Constraints
Supported Types	<p>The list of token types supported by the Identity Assertion provider. To see a list of default token types, refer the Javadoc for <code>weblogic.security.spi.IdentityAsserter</code>.</p> <p><i>MBean:</i> <code>weblogic.security.providers.authentication.DefaultIdentityAsserterMBean</code></p> <p><i>Attribute:</i> <code>SupportedTypes</code></p>	<p><i>Default:</i> <code>new String[] { weblogic.security.spi. IdentityAsserter.AU_TYPE, weblogic.security.spi. IdentityAsserter.X509_TYPE, weblogic.security.spi. IdentityAsserter. CSI_PRINCIPAL_TYPE, weblogic.security.spi. IdentityAsserter. CSI_ANONYMOUS_TYPE, weblogic.security.spi. IdentityAsserter. CSI_X509_CERTCHAIN_TYPE, weblogic.security.spi. IdentityAsserter. CSI_DISTINGUISHED_NAME_T YPE }</code></p>
Active Types	<p>Specifies what type of token is currently being used by the Identity Assertion provider.</p> <p><i>MBean:</i> <code>weblogic.security.providers.authentication.DefaultIdentityAsserterMBean</code></p> <p><i>Attribute:</i> <code>ActiveTypes</code></p>	



WebLogic Identity Assertion Provider-->Details

[Tasks](#) [Related Topics](#)

Overview

When using 2-way SSL, WebLogic Server verifies the digital certificate of the Web browser or Java client when establishing an SSL connection. However, the digital certificate does not identify the Web browser or Java client as a user in the WebLogic Server security realm. If the Web browser or Java client requests a WebLogic Server resource protected by a security policy, WebLogic Server requires the Web browser or Java client to have an identity. The WebLogic Identity Assertion provider allows you to define a user name mapper that maps the digital certificate of a client to a user in a WebLogic Server security realm.

Use this page to activate the default user name mapper and specify which attributes in a digital certificates are used to create the username. The attributes on the page are defined as follows:

- **Default User Name Mapper Attribute Type**—The attribute from the subject Distinguished Name (DN) which this WebLogic Identity Assertion provider should use when mapping from the X.509 digital certificate or X500 name token used to a username. Valid values are:
 - **C**—Country code.
 - **CN**—Common name.
 - **E**—Email address. (This is the default value).
 - **L**—Name of the city or town.
 - **O**—Organization name.
 - **OU**—Organization unit name (for example, the name of the division or group within a company).
 - **S**—State

-
- `Street`—The name of the stree.
 - `Default User Name Mapper Attribute Delimiter`—The delimiter that ends the attribute value when mapping from the X509 digital certificate or X500 name to the user name.

If the authentication type in a Web application is set to `CLIENT-CERT`, the Web Application Container in WebLogic Server performs identity assertion on values from request headers and cookies. If the header name or cookie name matches the active token type for the configured Identity Assertion provider, the value is passed to the provider.

The Base64 Decoding Required attribute determines whether the request header value or cookie value must be Base64 Decoded before sending it to the Identity Assertion provider. The setting is enabled by default for purposes of backward compatibility, however, most Identity Assertion providers will disable this attribute.

Tasks

“Configuring a WebLogic Identity Assertion Provider” on page 428-32

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation





Weblogic Keystore Provider-->Details

This page is not used in this release of WebLogic Server.

Note: The WebLogic Keystore provider is deprecated in this release of WebLogic Server.



WebLogic Keystore Provider-->General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure a WebLogic Keystore provider for a security realm. A keystore is a mechanism designed to store password-protected store private keys and trusted CA certificates. In the WebLogic Server security architecture, the WebLogic Keystore provider is used to access keystores. You cannot use a custom Keystore provider with WebLogic Server.

Note: The WebLogic Keystore provider is deprecated in this release of WebLogic Server.

Tasks

“Configuring Keystores and SSL” on page 428-43

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security](#) page in the WebLogic Server documentation

Attributes

Table 355-1

Attribute Label	Description	Value Constraints
Name	The name of this WebLogic Keystore provider. <i>MBean:</i> weblogic.security.providers.pk.DefaultKeyStoreMBean <i>Attribute:</i> Name	
Version	The version of this WebLogic Keystore provider. <i>MBean:</i> weblogic.security.providers.pk.DefaultKeyStoreMBean <i>Attribute:</i> Version	<i>Default:</i> "1.0"
Description	A short description of this WebLogic Keystore provider. <i>MBean:</i> weblogic.security.providers.pk.DefaultKeyStoreMBean <i>Attribute:</i> Description	<i>Default:</i> "WebLogic Keystore Security provider that obtains secured private keys and trusted CA certificates from a keystore."

Table 355-1

Attribute Label	Description	Value Constraints
Private Keystore Location	<p>The location of a JKS keystore that contains private keys for WebLogic Server. That is, a directory and filename location that is either a complete file path, or is relative to the server's root directory.</p> <p><i>MBean:</i> weblogic.security.providers.pk.DefaultKeyStoreMBean</p> <p><i>Attribute:</i> PrivateKeyStoreLocation</p>	<p><i>Default:</i> "wlDefaultKeyStore.jks"</p>
Private Keystore Pass Phrase	<p>The password for the private keystore that is specified in the Private Keystore Location field. If you set a null value, no password is required to access the private Keystore.</p> <p><i>MBean:</i> weblogic.security.providers.pk.DefaultKeyStoreMBean</p> <p><i>Attribute:</i> PrivateKeyStorePassPhrase</p>	<p><i>Default:</i> empty</p> <p><i>Encrypted:</i> yes</p>
Root CA Keystore Location	<p>The location of a JKS keystore that contains trusted CAs for WebLogic Server. The configured keystore implementation determines the input requirements for this attribute.</p> <p>For more information about legal values, refer to the documentation supplied by the keystore security vendor.</p> <p><i>MBean:</i> weblogic.security.providers.pk.DefaultKeyStoreMBean</p> <p><i>Attribute:</i> RootCAKeyStoreLocation</p>	<p><i>Default:</i> empty</p>

Table 355-1

Attribute Label	Description	Value Constraints
Root CA Keystore Pass Phrase	<p>The password for the keystore that is specified by the <code>RootCAKeyStoreLocation</code> attribute. If you set a null value, no password is required to access the keystore that contains trusted CAs. This behavior may be overridden by the configured keystore implementation.</p> <p>For more information about legal values, refer to the documentation supplied by the keystore security vendor.</p> <p><i>MBean:</i> <code>weblogic.security.providers.pk.DefaultKeyStoreMBean</code></p> <p><i>Attribute:</i> <code>RootCAKeyStorePassPhrase</code></p>	<p><i>Default:</i> empty</p> <p><i>Encrypted:</i> yes</p>
Type	<p>The type of keystore.</p> <p><i>MBean:</i> <code>weblogic.security.providers.pk.DefaultKeyStoreMBean</code></p> <p><i>Attribute:</i> <code>Type</code></p>	<p><i>Default:</i> "jks"</p>

WebLogic Role Mapping Provider-->Details

This page is not used in this release of WebLogic Server.



WebLogic Role Mapping Provider-->General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure a WebLogic Role Mapping provider.

Note: The WebLogic Server Administration Console refers to the WebLogic Role Mapping provider as the Default Role Mapper.

Role Mapping providers support security policies by obtaining a computed set of roles granted to a requestor for a given resource. Role Mapping providers supply Authorization providers with this role information so that the Authorization provider can answer the “is access allowed?” question for WebLogic resources that use role-based security.

The WebLogic Security Framework uses business logic and the current operation parameters (obtained from the J2EE and WebLogic deployment descriptor files) to determine which roles (if any) apply to the particular Subject at the moment in which access is required for a given resource. If multiple Role Mapping providers are configured, the set of roles returned by all Role Mapping providers will be unioned by the WebLogic Security Framework.

By default, the WebLogic Role Mapping provider is configured in the default security realm (`myrealm`). You can use a Custom Role Mapping provider instead of the WebLogic Role Mapping provider. For a Custom Role Mapping provider to be available through the WebLogic Server Administration Console, the MBean JAR file for the provider must be in the `WL_HOME\lib\mbeantypes` directory.

A Role Mapping provider that stores security roles that are created when deploying Web application or Enterprise JavaBean (EJB) deployments needs to implement the `DeployableRoleProvider` Security Service Provider Interface (SSPI) instead of the `RoleProvider` SSPI. You also need to enable the `Role Deployment Enabled` attribute on this page. The `Role Deployment Enabled` attribute is enabled by default for the WebLogic Role Mapping provider.

During application deployment, WebLogic Server reads security roles from the `weblogic.xml` and `weblogic-ejb-jar.xml` files. This information is used to populate the WebLogic Role Mapping provider. Any changes made to the security roles through the WebLogic Server Administration Console are not persisted to the `weblogic.xml` and `weblogic-ejb-jar.xml` files. Before you deploy the application again (which will happen if you redeploy it through the WebLogic Server Administration Console, modify it on disk, or restart WebLogic Server), you need enable the Check Roles and Policies and Future Redeploys options on the General page for a security realm.

Tasks

“Configuring a New Security Realm” on page 428-18

“Configuring the WebLogic Role Mapping Provider” on page 428-32

“Configuring a Custom Security Provider” on page 428-35

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 357-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this WebLogic Role Mapping provider.</p> <p><i>MBean:</i> weblogic.security.providers.authorization.DefaultRoleMapperMBean</p> <p><i>Attribute:</i> Name</p>	
Description	<p>A short description of this WebLogic Role Mapping provider.</p> <p><i>MBean:</i> weblogic.security.providers.authorization.DefaultRoleMapperMBean</p> <p><i>Attribute:</i> Description</p>	<p><i>Default:</i> "Weblogic Role Mapping Provider"</p>
Version	<p>The version number of this WebLogic Role Mapping provider.</p> <p><i>MBean:</i> weblogic.security.providers.authorization.DefaultRoleMapperMBean</p> <p><i>Attribute:</i> Version</p>	<p><i>Default:</i> "1.0"</p>
Role Deployment Enabled	<p>Specifies whether this WebLogic Role Mapping provider stores security roles that are created while deploying a Web application or an Enterprise JavaBean (EJB).</p> <p><i>MBean:</i> weblogic.security.providers.authorization.DefaultRoleMapperMBean</p> <p><i>Attribute:</i> RoleDeploymentEnabled</p>	<p><i>Default:</i> new java.lang.Boolean(true)</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false



WebLogic Role Mapping-->Export

[Tasks](#) [Related Topics](#)

Overview

Use this page to export security roles from a WebLogic Role Mapping provider. Use the Import page to then import the security roles into a WebLogic Role Mapping provider in another security realm.

Tasks

“Importing and Exporting Security Data from Security Providers” on page 428-40

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation



WebLogic Role Mapping Provider-->Import

[Tasks](#) [Related Topics](#)

Overview

Use this page to import security roles from a WebLogic Role Mapping provider in one security realm into a WebLogic Role Mapping provider in another security realm.

Tasks

“Importing and Exporting Security Data from Security Providers” on page 428-40

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security](#) page in the WebLogic Server documentation



Domain-->Security-->General

[Tasks](#) [Related Topics](#)

Overview

Use this page to change the default security realm for the WebLogic domain. All available security realms are listed on the pull-down menu. If you configure a new security realm, but do not configure any security providers or all the required security providers, the security realm will not be available from the pull-down menu. In order for a security realm to be valid, you must configure an Authentication provider, an Authorization provider, an Adjudication provider, a Credential Mapping provider, and a Role Mapping provider.

The Anonymous Admin Lookup Enabled attribute specifies whether anonymous, read-only access to WebLogic Server MBeans should be allowed from the MBeanHome API. With this anonymous access, you can see the value of any MBean attribute that is not explicitly marked as protected by the Weblogic Server MBean authorization process. This attribute is checked by default to sure backward compatibility. Unchecking this attribute will make the server more secure; however, it might cause applications that require anonymous access to MBeans to stop working properly.

Tasks

“Configuring a New Security Realm” on page 428-18

“Changing the Default Security Realm” on page 428-41

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Domain-->Security-->Filter

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Connection filters add an additional layer of security by filtering unwanted network connections. For example, you can deny any non-SSL connections originating outside of your corporate network. Use this page to configure a connection filter for the WebLogic domain.

Tasks

“Configuring Connection Filtering” on page 428-47

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 361-1

Attribute Label	Description	Value Constraints
Connection Logger Enabled	<p>Specifies whether this WebLogic domain should log accepted connections.</p> <p><i>MBean:</i> weblogic.management.configuration.SecurityMBean</p> <p><i>Attribute:</i> ConnectionLoggerEnabled</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false <p><i>Dynamic:</i> yes</p>
Connection Filter	<p>The name of the Java class that implements a connection filter. The connection filter must be an implementation of the weblogic.security.net.ConnectionFilter interface. WebLogic Server provides a default implementation.</p> <p><i>MBean:</i> weblogic.management.configuration.SecurityMBean</p> <p><i>Attribute:</i> ConnectionFilter</p>	<p><i>Default:</i> null</p>

Table 361-1

Attribute Label	Description	Value Constraints
Connection Filter Rules	<p>The list of rules for the system connection filter. If none are specified, all connections are accepted.</p> <p>The syntax of the rules is as follows:</p> <ul style="list-style-type: none">■ Each rule must be written on a single line in the source code.■ Tokens in a rule are separated by white space.■ A pound sign (#) is the comment character. Everything after a pound sign on a line is ignored.■ Whitespace before or after a rule is ignored.■ Lines consisting only of whitespace or comments are skipped. <p>All rules have the following format:</p> <pre>target localAddress localPort action protocols</pre> <p>where</p> <p><code>target</code> specifies one or more servers to filter.</p> <p><code>localAddress</code> defines the host address of the server. (If you specify an asterisk (*), the match returns all local IP addresses.)</p> <p><code>localPort</code> defines the port on which the server is listening. (If you specify an asterisk, the match returns all available ports on the server).</p> <p><code>action</code> specifies the action to perform. The value must be <code>allow</code> or <code>deny</code>.</p> <p><code>protocols</code> is the list of protocol names to match. (One of the following protocols must be specified <code>http</code>, <code>https</code>, <code>t3</code>, <code>t3s</code>, <code>giop</code>, <code>giops</code>, <code>dcom</code>, or <code>ftp</code>.) If no protocol is defined, all protocols will match a rule.</p>	<p><i>Default:</i> null</p> <p><i>Dynamic:</i> yes</p>

Table 361-1

Attribute Label	Description	Value Constraints
	<p>Two kinds of rules are recognized:</p> <ul style="list-style-type: none">■ A fast rule applies to a hostname or IP address with optional netmask. If a host name corresponds to multiple IP addresses, multiple rules are generated.■ A slow rule applies to part of a domain name. Since a rule requires a connect-time DNS lookup to perform a match, slow rules impact performance. <p><i>MBean:</i> weblogic.management.configuration.SecurityMBean</p> <p><i>Attribute:</i> ConnectionFilterRules</p>	

Domain-->Security-->Embedded LDAP

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The embedded LDAP server contains user, group, group membership, security role, security policy, and credential map information. By default, each WebLogic Server domain has an embedded LDAP server configured with the default values set for each attribute. The WebLogic Authentication, Authorization, Credential Mapping, and Role Mapping providers use the embedded LDAP server as their database. If you use any of these providers in a new security realm, you may want to change the default values for the embedded LDAP server to optimize its use in your environment.

Tasks

“Configuring the Embedded LDAP Server” on page 428-17

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security](#) page in the WebLogic Server documentation

Attributes

Attribute Label	Description	Value Constraints
Credential	The credential (usually a password) used to connect to the embedded LDAP server. If this password has not been set, WebLogic Server generates a password at startup, initializes the attribute, and saves the configuration to the <code>config.xml</code> file. If you want to connect to the embedded LDAP server using an external LDAP browser and the embedded LDAP administrator account (<code>cn=Admin</code>), change this attribute from the generated value.	<i>Default:</i> null <i>Configurable:</i> yes <i>Encrypted:</i> yes <i>Readable:</i> yes <i>Writable:</i> yes
Backup Hour	The hour at which to backup the embedded LDAP server data files. This attribute is used in conjunction with the Backup Minute attribute to determine the time at which the embedded LDAP server data files are backed up. At the specified time, WebLogic Server suspends writes to the embedded LDAP server, backs up the data files into a zip files in the <code>ldap/backup</code> directory, and then resumes writes. The default is 23.	<i>Minimum:</i> 0 <i>Maximum:</i> 23 <i>Default:</i> 23 <i>Configurable:</i> yes <i>Readable:</i> yes <i>Writable:</i> yes
Backup Minute	The minute at which to backup the embedded LDAP server data files. This attribute is used in conjunction with the Back Up Hour attribute to determine the time at which the embedded LDAP server data files are backed up. The default is 5 minutes.	<i>Minimum:</i> 0 <i>Maximum:</i> 59 <i>Default:</i> 05 <i>Configurable:</i> yes <i>Readable:</i> yes <i>Writable:</i> yes

Attribute Label	Description	Value Constraints
Backup Copies	The number of backup copies of the embedded LDAP server data files. This value limits the number of zip files in the <code>ldap/backup</code> directory. The default is 7.	<i>Minimum:</i> 0 <i>Maximum:</i> 65534 <i>Default:</i> 7 <i>Configurable:</i> yes <i>Readable:</i> yes <i>Writable:</i> yes
Cache Enabled	Specifies whether or not a cache is used with the embedded LDAP server. This cache is used when a managed server is reading or writing to the master embedded LDAP server that is running on the Administration server.	<i>Default:</i> true <i>Readable:</i> yes <i>Writable:</i> yes
Cache Size	The size of the cache (in K) that is used with the embedded LDAP server. The default is 32K.	<i>Minimum:</i> 0 <i>Default:</i> 32 <i>Configurable:</i> yes <i>Readable:</i> yes <i>Writable:</i> yes
Cache TTL	The time-to-live (TTL) of the cache in seconds. The default is 60 seconds.	<i>Minimum:</i> 0 <i>Default:</i> 60 <i>Configurable:</i> yes <i>Readable:</i> yes <i>Writable:</i> yes
Refresh Replica At Startup	Specifies whether or not a Managed server should refresh all replicated data at boot time. This attribute is useful if you have made a large number of changes while the Managed server was not active and you want to download the entire replica instead of having the Administration server push each change to the Managed server. The default is false.	<i>Default:</i> false <i>Readable:</i> yes <i>Writable:</i> yes

Attribute Label	Description	Value Constraints
Master First	Specifies that connections to the master LDAP server (running on the Administration server) should always be made instead of connections to the local replicated embedded LDAP server. This causes the Managed server to retrieve security data from the embedded LDAP server in the Administration server instead of going to the local embedded LDAP server that contains a replica of the information in the Administration server.	<i>Default:</i> false <i>Readable:</i> yes <i>Writable:</i> yes

Domain-->Security-->Advanced

[Tasks](#) [Related Topics](#)

Overview

Use this page to establish trust between WebLogic domains. A trust relationship is established when principals in a Subject from one WebLogic domain are accepted as principals in the local domain.

The domain credential is generated when WebLogic Server is started. This process ensures that by default no two WebLogic Server domains have the same credential. To enable trust between two WebLogic Server domains, you must explicitly specify the same value for the credential in both WebLogic Server domains.

Tasks

“Enabling Trust Between WebLogic Domains” on page 428-47

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Domain-->Compatibility Security-->Passwords

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

WebLogic Server provides a set of attributes to protect user accounts from intruders. By default, these attributes are set for maximum protection. As a system administrator, you have the option of turning off all the attributes, increasing the number of login attempts before a user account is locked, increasing the time period in which invalid login attempts are made before locking the user account, and changing the amount of time a user account is locked. Use this page to change these attributes. This page applies to WebLogic Server deployments using Compatibility security. Remember that changing the attributes on this page lessens security and leaves user accounts vulnerable to security attacks.

Tasks

“Protecting User Accounts” on page 322-31

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

Securing a Production Environment

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

Security FAQ

The [Security page](#) in the WebLogic Server documentation

Attributes

Attribute Label	Description	Value Constraints
Minimum Password Length	The minimum number of characters required in any domain password.	<i>Minimum:</i> 0 <i>Default:</i> 8 <i>Configurable:</i> yes <i>Readable:</i> yes <i>Writable:</i> yes
Lockout Enabled	Requests the locking of a user account after invalid attempts to log in to that account exceed the specified Lockout Threshold. By default, this attribute is enabled.	<i>Default:</i> true <i>Readable:</i> yes <i>Writable:</i> yes
Lockout Threshold	Number of failed user password entries that can be tried before that user 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. Invalid login attempts must be made within a span defined by the Lockout Reset Duration attribute. The default is 5.	<i>Minimum:</i> 1 <i>Maximum:</i> 99999 <i>Default:</i> 5 <i>Configurable:</i> yes <i>Readable:</i> yes <i>Writable:</i> yes

Attribute Label	Description	Value Constraints
Lockout Duration	Number of minutes that a user's account remains inaccessible after being locked in response to several invalid login attempts within the amount of time specified by the Lockout Reset Duration attribute. The default is 30 minutes.	<i>Minimum:</i> 0 <i>Maximum:</i> 999999 <i>Default:</i> 30 <i>Configurable:</i> yes <i>Readable:</i> yes <i>Writable:</i> yes
Lockout Reset Duration	<p>Number of minutes within which invalid login attempts must occur in order for the user's account to be locked.</p> <p>An account is locked if the number of invalid login attempts defined in the Lockout Threshold attribute happens within the amount of time defined by this attribute. For example, if the value in Lockout Reset Duration attribute is 5 minutes, the Lockout Threshold is 3, and 3 invalid login attempts are made within a 6 minute interval, then the account is not locked. If 3 invalid login attempts are made within a 5 minute period, however, then the account is locked.</p> <p>The default is 5 minutes.</p>	<i>Minimum:</i> 1 <i>Maximum:</i> 99999 <i>Default:</i> 5 <i>Configurable:</i> yes <i>Readable:</i> yes <i>Writable:</i> yes
Lockout Cache Size	Specifies the intended cache size of unused and invalid login attempts. The default is 5.	<i>Minimum:</i> 0 <i>Maximum:</i> 99999 <i>Default:</i> 5 <i>Configurable:</i> yes <i>Readable:</i> yes <i>Writable:</i> yes



Domain-->Compatibility Security-->General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure an Auditing provider for the CompatibilityRealm and to disable the `guest` user for a WebLogic domain. This page applies to WebLogic Server deployments using Compatibility Security.

The Auditing provider installed from this page must be an implementation of the `weblogic.security.audit` interface. This interface is deprecated in this release of WebLogic Server.

The `guest` user is automatically provided in Compatibility security. When authorization is not required, WebLogic Server assigns the `guest` identity to an client, thus giving the client access to any resources that are available to the `guest` user. A client can log in as the `guest` user by entering `guest` as the username and password when prompted by a Web browser or in a Java client. By default, the `guest` account is disabled. For a more secure deployment, BEA recommends running WebLogic Server with the `guest` account disabled.

Tasks

“Setting Up Compatibility Security: Main Steps” on page 322-1

“Disabling the Guest User” on page 322-28

“Installing an Audit Provider” on page 322-31

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Attribute Label	Description	Value Constraints
Audit Provider Class	<i>MBean:</i> weblogic.management.configuration.SecurityMBean <i>Attribute:</i> AuditProviderClassName	<i>Configurable:</i> yes <i>Readable:</i> yes <i>Writable:</i> yes
Guest Disabled	Specifies whether or not guest logins can be used to access WebLogic Server resources. <i>MBean:</i> weblogic.management.configuration.SecurityMBean <i>Attribute:</i> GuestDisabled	<i>Default:</i> true <i>Readable:</i> yes <i>Writable:</i> yes

Domain -->Compatibility Security-->File Realm

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The File realm stores user and group data for the purpose of authentication. By default, the File realm is set as the default security realm when using Compatibility security. Use this page to define several attributes that govern the use of the File realm. The File realm is deprecated in this release of WebLogic Server.

Tasks

“Setting Up Compatibility Security: Main Steps” on page 322-1

“Configuring the File Realm” on page 322-4

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Attribute Label	Description	Value Constraints
Caching Realm	The name of an alternate security realm to be used in this WebLogic server domain. If none is specified, only the File realm can be used.	<i>Configurable:</i> yes <i>Readable:</i> yes <i>Writable:</i> yes
Max Users	The maximum number of users (1-1000) supported by File realm.	<i>Minimum:</i> 1 <i>Maximum:</i> 10000 <i>Default:</i> 1000 <i>Configurable:</i> yes <i>Readable:</i> yes <i>Writable:</i> yes
Max Groups	The maximum number of groups (1-1000) supported by the File realm.	<i>Minimum:</i> 1 <i>Maximum:</i> 10000 <i>Default:</i> 1000 <i>Configurable:</i> yes <i>Readable:</i> yes <i>Writable:</i> yes
Max ACLs	The maximum number of positive access control lists (ACLs) (1-10000) supported by the File realm. A warning is issued when this number is reached.	<i>Minimum:</i> 1 <i>Maximum:</i> 10000 <i>Default:</i> 1000 <i>Configurable:</i> yes <i>Readable:</i> yes <i>Writable:</i> yes

Domain-->Compatibility Security-->Advanced

[Related Topics](#)

Overview

Use this page to determine how user, group, and ACL information is returned in a thread and enable the logging of all security checks. This page applies to WebLogic Server deployments using Compatibility security.

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation



EJB Policies and Roles

Use this page to assign scoped roles and security policies to an EJB JAR, an individual EJB, or a method of an EJB.

- For more information, see [Securing WebLogic Resources](#).



Groups

This page lists all the groups defined in the default security realm. The Provider column indicates the Authentication provider in which the group is defined. In order to use the WebLogic Server Administration Console to define groups in an Authentication provider, the provider must implement in the GroupReader SSPI.

To define a new group in the security realm, click the [Configure a new Group...](#) link on this page.

If you have a large number of groups, use the Filter By field to retrieve and list only the groups that match your search criteria. The Filter By field uses the asterisk (*) as the wildcard character.

To delete an existing group, click the trash can icon that is located in the same row as the group you want to delete.

- For more information, see [“Defining Groups” on page 428-2](#).



Groups-->General

[Tasks](#) [Related Topics](#)

Overview

Use this page to add groups to an Authentication provider. The WebLogic Server Administration Console detects when an Authentication provider implements the GroupReader MBean and automatically loads group information into the provider. The Provider column indicates the Authentication provider in which the group is defined.

A group is an abstract, logical collection of users which has static membership. Groups can be used to define security policies for WebLogic resources. By default, WebLogic Server has the following groups:

- **Administrators**—View and modify all resource attributes and perform start and stop operations. By default, the user from WebLogic Server is booted is included in this group.
- **Operators**—View all resource attributes and perform server lifecycle operations. By default, this group is empty.
- **Deployers**—View all resource attributes and deploy applications such as EJBs. By default, this group is empty.
- **Monitors**—View all resource attributes, modify resource attributes, and perform operations that are not restricted by a role. By default, this group is empty.

You do not have to use the default groups provided by WebLogic Server. BEA recommends creating groups that more closely reflect your own business structure and practices.

Note: Group and user names must be unique.

If you have a large number of groups, use the Filter By field to retrieve and list only the groups that match your search criteria. The Filter By field uses the asterisk (*) as the wildcard character.

To delete an existing group, click the trash can icon that is located in the same row as the group you want to delete.

Tasks

“Defining Groups” on page 428-2

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Group-->Details

This page is not used in this release of WebLogic Server.



Group-->Membership

Use this page to add a group to one of the groups available in configured Authentication provider.

- For more information, see [“Defining Groups” on page 428-2](#).



iPlanet Authentication Provider-->Details

[Tasks](#) [Related Topics](#)

Overview

This page displays additional MBean attributes for the iPlanet Authentication provider.

- **Follow Referrals**—Specifies that a search for a user or group within the iPlanet Authentication provider will follow referrals to other LDAP servers or branches within the LDAP directory. By default, this attribute is enabled.
- **Bind Anonymously On Referrals**—By default, the iPlanet Authentication provider uses the same DN and password used to connect to the LDAP server when following referrals during a search. If you want to connect as an anonymous user, enable this attribute. Contact your LDAP system administrator for more information.
- **Results Time Limit**—The maximum number of milliseconds for the LDAP server to wait for results before timing out. If this attribute is set to 0, there is not maximum time limit. The default is 0.
- **Connect Timeout**—The maximum time in seconds to wait for the connection to the LDAP server to be established. If this attribute is set to 0, there is not a maximum time limit. The default is 0.
- **Parallel Connect Delay**—The delay in seconds when making concurrent attempts to attempt to multiple LDAP servers. If this attribute is set to 0, connection attempts are serialized. An attempt is made to connect to the first server in the list. The next entry in the list is tried only if the attempt to connect to the current host fails. If this attribute is not set and an LDAP server is unavailable, an application may be blocked for a long time. If this attribute is greater than 0, another connection is started after the specified time.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

iPlanet Authentication Provider-->General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure attributes for the iPlanet Authentication provider.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 374-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> Name</p>	
Description	<p>A short description of the LDAP Authentication provider.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> Description</p>	<p><i>Default:</i> "Provider that performs LDAP authentication"</p>
Version	<p>The version number of the LDAP Authentication provider.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> Version</p>	<p><i>Default:</i> "1.0"</p>

Table 374-1

Attribute Label	Description	Value Constraints
Control Flag	<p>Determines how the login sequence uses the Authentication provider.</p> <p>A REQUIRED value specifies this LoginModule must succeed. Even if it fails, authentication proceeds down the list of LoginModules for the configured Authentication providers.</p> <p>A REQUISITE value specifies this LoginModule must succeed. If other Authentication providers are configured and this LoginModule succeeds, authentication proceeds down the list of LoginModules. Otherwise, control is return to the application.</p> <p>A SUFFICIENT value specifies this LoginModule need not succeed. If it does succeed, return control to the application. If it fails and other Authentication providers are configured, authentication proceeds down the LoginModule list.</p> <p>An OPTIONAL value specifies this LoginModule need not succeed. Whether it succeeds or fails, authentication proceeds down the LoginModule list. This setting is the default.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> ControlFlag</p>	<i>Default:</i> "REQUIRED"



iPlanet Authentication Provider-->Groups

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to specify how groups are stored in the iPlanet LDAP directory.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 375-1

Attribute Label	Description	Value Constraints
Group Base DN	The base distinguished name (DN) of the tree in the LDAP directory that contains groups. <i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean <i>Attribute:</i> GroupBasedDN	<i>Default:</i> "ou=groups, o=example.com"
Group Search Scope	Specifies how deep in the LDAP directory tree to search for groups. Valid values are subtree and onelevel. <i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean <i>Attribute:</i> GroupSearchScope	<i>Default:</i> "subtree"
Group From Name Filter	An LDAP search filter for finding a group given the name of the group. If the attribute is not specified (that is, if the attribute is null or empty), a default search filter is created based on the group schema. <i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean <i>Attribute:</i> GroupFromNameFilter	<i>Default:</i> "(&(cn=%g)(objectclass=groupofuniqueNames))"

Table 375-1

Attribute Label	Description	Value Constraints
All Groups Filter	<p>An LDAP search filter for finding all groups beneath the base group distinguished name (DN). If the attribute is not specified (that is, if the attribute is null or empty), a default search filter is created based on the Group schema.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> AllGroupsFilter</p>	
Static Group Object Class	<p>The name of the LDAP object class that stores static groups.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> StaticGroupObjectClass</p>	<i>Default:</i> "groupofuniqueNames"
Static Group Name Attribute	<p>The attribute of a static LDAP group object that specifies the name of the group.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> StaticGroupNameAttribute</p>	<i>Default:</i> "cn"



iPlanet Authentication Provider-->iPlanet LDAP

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure the iPlanet LDAP server to enable communication between the LDAP server and WebLogic Server.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 376-1

Attribute Label	Description	Value Constraints
Host	The host name or IP address of the LDAP server. <i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean <i>Attribute:</i> Host	<i>Default:</i> "localhost"
Port	The port number on which the LDAP server is listening. <i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean <i>Attribute:</i> Port	<i>Default:</i> new java.lang.Integer(389)
SSLEnabled	The option to enable the use of the SSL protocol when connecting to the LDAP server. <i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean <i>Attribute:</i> SSLEnabled	<i>Default:</i> new java.lang.Boolean(false) <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false
Principal	Distinguished Name (DN) of the LDAP user used by WebLogic Server to connect to the LDAP server. <i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean <i>Attribute:</i> Principal	

Table 376-1

Attribute Label	Description	Value Constraints
Credential	<p>The credential (generally a password) used to authenticate the LDAP user defined in the Principal attribute.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> Credential</p>	<p><i>Encrypted:</i> yes</p>
Cache Enabled	<p>Enables the use of a cache with the LDAP server. This is a cache of the LDAP requests.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> CacheEnabled</p>	<p><i>Default:</i> new java.lang.Boolean(true)</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false
Cache Size	<p>The size of the cache in K.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> CacheSize</p>	<p><i>Default:</i> new java.lang.Integer(32)</p>
Cache TTL	<p>The time-to-live (TTL) of the cache in seconds.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> CacheTTL</p>	<p><i>Default:</i> new java.lang.Integer(60)</p>



iPlanet Authentication Provider-->Membership

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to specify how groups are stored and located in the iPlanet directory.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 377-1

Attribute Label	Description	Value Constraints
Static Member DNAttribute	<p>The attribute of an LDAP static group object that specifies the distinguished names (DNs) of the members of the group.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> StaticMemberDNAttribute</p>	<p><i>Default:</i> "uniquemember"</p>
Static Group DN's from Member DNFilter	<p>An LDAP search filter that, given the distinguished name (DN) of a member of a group, returns the DN's of the static LDAP groups that contain that member. If the attribute is not specified (that is, if the attribute is null or empty), a default search filter is created based on the group schema.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> StaticGroupDN'sfromMemberDNFilter</p>	<p><i>Default:</i> "(&(uniquemember=%M)(objectclass=groupofuniquenames))"</p>
Dynamic Group Object Class	<p>The LDAP object class that stores dynamic groups.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> DynamicGroupObjectClass</p>	<p><i>Default:</i> "groupofURLs"</p>

Table 377-1

Attribute Label	Description	Value Constraints
Dynamic Group Name Attribute	<p>The attribute of the dynamic LDAP group object that specifies the name of the group.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> DynamicGroupNameAttribute</p>	<i>Default:</i> "cn"
Dynamic Member URLAttribute	<p>The attribute of the dynamic LDAP group object that specifies the URLs of the members of the dynamic group.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> DynamicMemberURLAttribute</p>	<i>Default:</i> "memberURL"



iPlanet Authentication Provider-->Users

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to specify how users are stored in the iPlanet LDAP directory.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 378-1

Attribute Label	Description	Value Constraints
User Object Class	The LDAP object class that stores users. <i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean <i>Attribute:</i> UserObjectClass	<i>Default:</i> "person"
User Name Attribute	The attribute of an LDAP user object that specifies the name of the user. <i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean <i>Attribute:</i> UserNameAttribute	<i>Default:</i> "uid"
User Dynamic Group DNAttribute	The attribute of an LDAP user object that specifies the distinguished names (DNs) of dynamic groups to which this user belongs. If such an attribute does not exist, WebLogic Server determines if a user is a member of a group by evaluating the URLs on the dynamic group. If a group contains other groups, WebLogic Server evaluates the URLs on any of the descendents (indicates parent relationship) of the group. <i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean <i>Attribute:</i> UserDynamicGroupDNAttribute	

Table 378-1

Attribute Label	Description	Value Constraints
User Base DN	<p>The base distinguished name (DN) of the tree in the LDAP directory that contains users.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> UserBaseDN</p>	<i>Default:</i> "ou=people, o=example.com"
User Search Scope	<p>Specifies how deep in the LDAP directory tree to search for Users. Valid values are subtree and onelevel.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> UserSearchScope</p>	<i>Default:</i> "subtree"
User From Name Filter	<p>An LDAP search filter for finding a user given the name of the user. If the attribute (user name attribute and user object class) is not specified (that is, if the attribute is null or empty), a default search filter is created based on the user schema.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> UserFromNameFilter</p>	<i>Default:</i> "(&(uid=%u)(objectclass=person))"
All Users Filter	<p>An LDAP search filter for finding all users beneath the base user distinguished name (DN). If the attribute (user object class) is not specified (that is, if the attribute is null or empty), a default search filter is created based on the user schema.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.IPlanetAuthenticatorMBean</p> <p><i>Attribute:</i> AllUsersFilter</p>	



Keystore Provider

This page displays the name of the Keystore provider configured for the default security realm (for example, myrealm).

Note: The WebLogic Keystore provider is deprecated in this release of WebLogic Server.



Novell Authentication Provider-->General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure attributes for the Novell Authentication provider.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 380-1

Attribute Label	Description	Value Constraints
Name	The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration. <i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean <i>Attribute:</i> Name	
Description	A short description of the LDAP Authentication provider. <i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean <i>Attribute:</i> Description	<i>Default:</i> "Provider that performs LDAP authentication"
Version	The version number of the LDAP Authentication provider. <i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean <i>Attribute:</i> Version	<i>Default:</i> "1.0"

Table 380-1

Attribute Label	Description	Value Constraints
Control Flag	<p>Determines how the login sequence uses the Authentication provider.</p> <p>A <code>REQUIRED</code> value specifies this LoginModule must succeed. Even if it fails, authentication proceeds down the list of LoginModules for the configured Authentication providers.</p> <p>A <code>REQUISITE</code> value specifies this LoginModule must succeed. If other Authentication providers are configured and this LoginModule succeeds, authentication proceeds down the list of LoginModules. Otherwise, control is return to the application.</p> <p>A <code>SUFFICIENT</code> value specifies this LoginModule need not succeed. If it does succeed, return control to the application. If it fails and other Authentication providers are configured, authentication proceeds down the LoginModule list.</p> <p>An <code>OPTIONAL</code> value specifies this LoginModule need not succeed. Whether it succeeds or fails, authentication proceeds down the LoginModule list. This setting is the default.</p> <p><i>MBean:</i> <code>weblogic.security.providers.authentication.NovellAuthenticatorMBean</code></p> <p><i>Attribute:</i> <code>ControlFlag</code></p>	<i>Default:</i> "REQUIRED"



Novell Authentication Provider-->Details

[Tasks](#) [Related Topics](#)

Overview

This page displays additional MBean attributes for the Novell Authentication provider.

- **Follow Referrals**—Specifies that a search for a user or group within the Novell Authentication provider will follow referrals to other LDAP servers or branches within the LDAP directory. By default, this attribute is enabled.
- **Bind Anonymously On Referrals**—By default, the Novell Authentication provider uses the same DN and password used to connect to the LDAP server when following referrals during a search. If you want to connect as an anonymous user, enable this attribute. Contact your LDAP system administrator for more information.
- **Results Time Limit**—The maximum number of milliseconds for the LDAP server to wait for results before timing out. If this attribute is set to 0, there is not maximum time limit. The default is 0.
- **Connect Timeout**—The maximum time in seconds to wait for the connection to the LDAP server to be established. If this attribute is set to 0, there is not a maximum time limit. The default is 0.
- **Parallel Connect Delay**—The delay in seconds when making concurrent attempts to attempt to multiple LDAP servers. If this attribute is set to 0, connection attempts are serialized. An attempt is made to connect to the first server in the list. The next entry in the list is tried only if the attempt to connect to the current host fails. If this attribute is not set and an LDAP server is unavailable, an application may be blocked for a long time. If this attribute is greater than 0, another connection is started after the specified time.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Novell Authentication Provider-->Groups

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to specify how groups are stored in the Novell NDS directory.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 382-1

Attribute Label	Description	Value Constraints
Group Base DN	The base distinguished name (DN) of the tree in the LDAP directory that contains groups. <i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean <i>Attribute:</i> GroupBasedDN	<i>Default:</i> "ou=groups, o=example.com"
Group Search Scope	Specifies how deep in the LDAP directory tree to search for groups. Valid values are subtree and onelevel. <i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean <i>Attribute:</i> GroupSearchScope	<i>Default:</i> "subtree"
Group From Name Filter	An LDAP search filter for finding a group given the name of the group. If the attribute is not specified (that is, if the attribute is null or empty), a default search filter is created based on the group schema. <i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean <i>Attribute:</i> GroupFromNameFilter	<i>Default:</i> "(&(cn=%g)(objectclass=groupofuniqueNames))"

Table 382-1

Attribute Label	Description	Value Constraints
All Groups Filter	<p>An LDAP search filter for finding all groups beneath the base group distinguished name (DN). If the attribute is not specified (that is, if the attribute is null or empty), a default search filter is created based on the Group schema.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean</p> <p><i>Attribute:</i> AllGroupsFilter</p>	
Static Group Object Class	<p>The name of the LDAP object class that stores static groups.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean</p> <p><i>Attribute:</i> StaticGroupObjectClass</p>	<i>Default:</i> "groupofnames"
Static Group Name Attribute	<p>The attribute of a static LDAP group object that specifies the name of the group.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean</p> <p><i>Attribute:</i> StaticGroupNameAttribute</p>	<i>Default:</i> "cn"



Novell Authentication Provider-->Membership

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to define how group members are stored and located in the Novell NDS directory.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 383-1

Attribute Label	Description	Value Constraints
Static Member DNAttribute	<p>The attribute of a static LDAP group object that specifies the distinguished names (DNs) of the members of the group.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean</p> <p><i>Attribute:</i> StaticMemberDNAttribute</p>	<p><i>Default:</i> "uniquemember"</p>
Static Group DN's from Member DNFilter	<p>An LDAP search filter that, given the distinguished name (DN) of a member of a group, returns the DN's of the static LDAP groups that contain that member. If the attribute is not specified (that is, if the attribute is null or empty), a default search filter is created based on the group schema.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean</p> <p><i>Attribute:</i> StaticGroupDN'sfromMemberDNFilter</p>	<p><i>Default:</i> "(&(uniquemember=%M)(objectclasses=groupofnames))"</p>
Dynamic Group Object Class	<p>The LDAP object class that stores dynamic groups.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean</p> <p><i>Attribute:</i> DynamicGroupObjectClass</p>	

Table 383-1

Attribute Label	Description	Value Constraints
Dynamic Group Name Attribute	<p>The attribute of a dynamic LDAP group object that specifies the name of the group.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean</p> <p><i>Attribute:</i> DynamicGroupNameAttribute</p>	
Dynamic Member URLAttribute	<p>The attribute of the dynamic LDAP group object that specifies the URLs of the members of the dynamic group.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean</p> <p><i>Attribute:</i> DynamicMemberURLAttribute</p>	



Novell Authentication Provider-->Novell LDAP

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure the Novell NDS LDAP server to enable communication between the LDAP server and WebLogic Server.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 384-1

Attribute Label	Description	Value Constraints
Host	The host name or IP address of the LDAP server. <i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean <i>Attribute:</i> Host	<i>Default:</i> "localhost"
Port	The port number on which the LDAP server is listening. <i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean <i>Attribute:</i> Port	<i>Default:</i> new java.lang.Integer(389)
SSLEnabled	The option to enable the use of the SSL protocol when connecting to the LDAP server. <i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean <i>Attribute:</i> SSLEnabled	<i>Default:</i> new java.lang.Boolean(false) <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false
Principal	Distinguished Name (DN) of the LDAP user used by WebLogic Server to connect to the LDAP server. <i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean <i>Attribute:</i> Principal	

Table 384-1

Attribute Label	Description	Value Constraints
Credential	<p>The credential (generally a password) used to authenticate the LDAP user defined in the Principal attribute.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean</p> <p><i>Attribute:</i> Credential</p>	<p><i>Encrypted:</i> yes</p>
Cache Enabled	<p>Enables the use of a cache with the LDAP server. This is a cache of the LDAP requests.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean</p> <p><i>Attribute:</i> CacheEnabled</p>	<p><i>Default:</i> new java.lang.Boolean(true)</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false
Cache Size	<p>The size of the cache in K.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean</p> <p><i>Attribute:</i> CacheSize</p>	<p><i>Default:</i> new java.lang.Integer(32)</p>
Cache TTL	<p>The time-to-live (TTL) of the cache in seconds.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean</p> <p><i>Attribute:</i> CacheTTL</p>	<p><i>Default:</i> new java.lang.Integer(60)</p>



Novell Authentication Provider-->Users

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to specify how users are stored in the Novell NDS directory.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 385-1

Attribute Label	Description	Value Constraints
User Object Class	The LDAP object class that stores users. <i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean <i>Attribute:</i> UserObjectClass	<i>Default:</i> "person"
User Name Attribute	The attribute of an LDAP user object that specifies the name of the user. <i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean <i>Attribute:</i> UserNameAttribute	<i>Default:</i> "uid"
User Dynamic Group DNAttribute	The attribute of an LDAP user object that specifies the distinguished names (DNs) of dynamic groups to which this user belongs. If such an attribute does not exist, WebLogic Server determines if a user is a member of a group by evaluating the URLs on the dynamic group. If a group contains other groups, WebLogic Server evaluates the URLs on any of the descendents (indicates parent relationship) of the group. <i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean <i>Attribute:</i> UserDynamicGroupDNAttribute	

Table 385-1

Attribute Label	Description	Value Constraints
User Base DN	<p>The base distinguished name (DN) of the tree in the LDAP directory that contains users.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean</p> <p><i>Attribute:</i> UserBaseDN</p>	<i>Default:</i> "ou=people, o=example.com"
User Search Scope	<p>Specifies how deep in the LDAP directory tree to search for Users. Valid values are subtree and onelevel.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean</p> <p><i>Attribute:</i> UserSearchScope</p>	<i>Default:</i> "subtree"
User From Name Filter	<p>An LDAP search filter for finding a user given the name of the user. If the attribute is not specified (that is, if the attribute is null or empty), a default search filter is created based on the user schema.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean</p> <p><i>Attribute:</i> UserFromNameFilter</p>	<i>Default:</i> "(&(uid=%u)(objectclass=person))"
All Users Filter	<p>An LDAP search filter for finding all users beneath the base user distinguished name (DN). If the attribute (user object class) is not specified (that is, if the attribute is null or empty), a default search filter is created based on the user schema.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.NovellAuthenticatorMBean</p> <p><i>Attribute:</i> AllUsersFilter</p>	



Open LDAP Authentication Provider-->General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure attributes for the Open LDAP Authentication provider.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 386-1

Attribute Label	Description	Value Constraints
Name	The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration. <i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean <i>Attribute:</i> Name	
Description	A short description of the LDAP Authentication provider. <i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean <i>Attribute:</i> Description	<i>Default:</i> "Provider that performs LDAP authentication"
Version	The version number of the LDAP Authentication provider. <i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean <i>Attribute:</i> Version	<i>Default:</i> "1.0"

Table 386-1

Attribute Label	Description	Value Constraints
Control Flag	<p>Determines how the login sequence uses the Authentication provider.</p> <p>A REQUIRED value specifies this LoginModule must succeed. Even if it fails, authentication proceeds down the list of LoginModules for the configured Authentication providers.</p> <p>A REQUISITE value specifies this LoginModule must succeed. If other Authentication providers are configured and this LoginModule succeeds, authentication proceeds down the list of LoginModules. Otherwise, control is return to the application.</p> <p>A SUFFICIENT value specifies this LoginModule need not succeed. If it does succeed, return control to the application. If it fails and other Authentication providers are configured, authentication proceeds down the LoginModule list.</p> <p>An OPTIONAL value specifies this LoginModule need not succeed. Whether it succeeds or fails, authentication proceeds down the LoginModule list. This setting is the default.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean</p> <p><i>Attribute:</i> ControlFlag</p>	<i>Default:</i> "REQUIRED"



Open LDAP Authentication Provider-->Details

[Tasks](#) [Related Topics](#)

Overview

This page displays additional MBean attributes for the Open LDAP Authentication provider.

- **Follow Referrals**—Specifies that a search for a user or group within the Open LDAP Authentication provider will follow referrals to other LDAP servers or branches within the LDAP directory. By default, this attribute is enabled.
- **Bind Anonymously On Referrals**—By default, the Open LDAP Authentication provider uses the same DN and password used to connect to the LDAP server when following referrals during a search. If you want to connect as an anonymous user, enable this attribute. Contact your LDAP system administrator for more information.
- **Results Time Limit**—The maximum number of milliseconds for the LDAP server to wait for results before timing out. If this attribute is set to 0, there is not maximum time limit. The default is 0.
- **Connect Timeout**—The maximum time in seconds to wait for the connection to the LDAP server to be established. If this attribute is set to 0, there is not a maximum time limit. The default is 0.
- **Parallel Connect Delay**—The delay in seconds when making concurrent attempts to attempt to multiple LDAP servers. If this attribute is set to 0, connection attempts are serialized. An attempt is made to connect to the first server in the list. The next entry in the list is tried only if the attempt to connect to the current host fails. If this attribute is not set and an LDAP server is unavailable, an application may be blocked for a long time. If this attribute is greater than 0, another connection is started after the specified time.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Open LDAP Authentication Provider-->Groups

[Tasks](#) [Related Topics](#)

Overview

Use this page to specify how groups are stored in the Open LDAP directory.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 388-1

Attribute Label	Description	Value Constraints
Group Base DN	The base distinguished name (DN) of the tree in the LDAP directory that contains groups. <i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean <i>Attribute:</i> GroupBasedDN	<i>Default:</i> "ou=groups, dc=example, dc=com"
Group Search Scope	Specifies how deep in the LDAP directory tree to search for groups. Valid values are subtree and onelevel. <i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean <i>Attribute:</i> GroupSearchScope	<i>Default:</i> "subtree"
Group From Name Filter	An LDAP search filter for finding a group given the name of the group. If the attribute is not specified (that is, if the attribute is null or empty), a default search filter is created based on the group schema. <i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean <i>Attribute:</i> GroupFromNameFilter	<i>Default:</i> "(&(cn=%g)(objectclass=groupofnames))"

Table 388-1

Attribute Label	Description	Value Constraints
All Groups Filter	<p>An LDAP search filter for finding all groups beneath the base group distinguished name (DN). If the attribute is not specified (that is, if the attribute is null or empty), a default search filter is created based on the Group schema.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean</p> <p><i>Attribute:</i> AllGroupsFilter</p>	
Static Group Object Class	<p>The name of the LDAP object class that stores static groups.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean</p> <p><i>Attribute:</i> StaticGroupObjectClass</p>	<i>Default:</i> "groupofuniqueNames"
Static Group Name Attribute	<p>The attribute of a static LDAP group object that specifies the name of the group.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean</p> <p><i>Attribute:</i> StaticGroupNameAttribute</p>	<i>Default:</i> "cn"



Open LDAP Authentication Provider-->Membership

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to define how group members are stored and located in the Open LDAP directory.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 389-1

Attribute Label	Description	Value Constraints
Static Member DNAttribute	<p>The attribute of an LDAP static group object that specifies the distinguished names (DNs) of the members of the group.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean</p> <p><i>Attribute:</i> StaticMemberDNAttribute</p>	<p><i>Default:</i> "uniquemember"</p>
Static Group DNs from Member DNFilter	<p>An LDAP search filter that, given the distinguished name (DN) of a member of a group, returns the DNs of the static LDAP groups that contain that member.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean</p> <p><i>Attribute:</i> StaticGroupDNsfromMemberDNFilter</p>	<p><i>Default:</i> "(&(uniquemember=%M)(objectclass=groupofuniquenames))"</p>
Dynamic Group Object Class	<p>The LDAP object class that stores dynamic groups.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean</p> <p><i>Attribute:</i> DynamicGroupObjectClass</p>	

Table 389-1

Attribute Label	Description	Value Constraints
Dynamic Group Name Attribute	<p>The attribute of a dynamic LDAP group object that specifies the name of the group.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean</p> <p><i>Attribute:</i> DynamicGroupNameAttribute</p>	
Dynamic Member URLAttribute	<p>The attribute of the dynamic LDAP group object that specifies the URLs of the members of the dynamic group.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean</p> <p><i>Attribute:</i> DynamicMemberURLAttribute</p>	



Open LDAP Authentication Provider-->Open LDAP

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure the Open LDAP server to enable communication between the LDAP server and WebLogic Server.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 390-1

Attribute Label	Description	Value Constraints
Host	The host name or IP address of the LDAP server. <i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean <i>Attribute:</i> Host	<i>Default:</i> "localhost"
Port	The port number on which the LDAP server is listening. <i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean <i>Attribute:</i> Port	<i>Default:</i> new java.lang.Integer(389)
SSLEnabled	The option to enable the use of the SSL protocol when connecting to the LDAP server. <i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean <i>Attribute:</i> SSLEnabled	<i>Default:</i> new java.lang.Boolean(false) <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false
Principal	Distinguished Name (DN) of the LDAP user used by WebLogic Server to connect to the LDAP server. <i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean <i>Attribute:</i> Principal	

Table 390-1

Attribute Label	Description	Value Constraints
Credential	<p>The credential (generally a password) used to authenticate the LDAP user defined in the Principal attribute.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean</p> <p><i>Attribute:</i> Credential</p>	<p><i>Encrypted:</i> yes</p>
Cache Enabled	<p>Enables the use of a cache with the LDAP server. This is a cache of the LDAP requests.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean</p> <p><i>Attribute:</i> CacheEnabled</p>	<p><i>Default:</i> new java.lang.Boolean(true)</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false
Cache Size	<p>The size of the cache in K.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean</p> <p><i>Attribute:</i> CacheSize</p>	<p><i>Default:</i> new java.lang.Integer(32)</p>
Cache TTL	<p>The time-to-live (TTL) of the cache in seconds.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean</p> <p><i>Attribute:</i> CacheTTL</p>	<p><i>Default:</i> new java.lang.Integer(60)</p>



Open LDAP Authentication Provider-->Users

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to how users are stored in Open LDAP directory.

Tasks

“Configuring an LDAP Authentication Provider” on page 428-25

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 391-1

Attribute Label	Description	Value Constraints
User Object Class	The LDAP object class that stores users. <i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean <i>Attribute:</i> UserObjectClass	<i>Default:</i> "person"
User Name Attribute	The attribute of an LDAP user object that specifies the name of the user. <i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean <i>Attribute:</i> UserNameAttribute	<i>Default:</i> "uid"
User Dynamic Group DNAttribute	The attribute of an LDAP user object that specifies the distinguished names (DNs) of dynamic groups to which this user belongs. If such an attribute does not exist, WebLogic Server determines if a user is a member of a group by evaluating the URLs on the dynamic group. If a group contains other groups, WebLogic Server evaluates the URLs on any of the descendents (indicates parent relationship) of the group. <i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean <i>Attribute:</i> UserDynamicGroupDNAttribute	

Table 391-1

Attribute Label	Description	Value Constraints
User Base DN	<p>The base distinguished name (DN) of the tree in the LDAP directory that contains users.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean</p> <p><i>Attribute:</i> UserBaseDN</p>	<i>Default:</i> "ou=people, o=example.com"
User Search Scope	<p>Specifies how deep in the LDAP directory tree to search for Users. Valid values are subtree and onelevel.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean</p> <p><i>Attribute:</i> UserSearchScope</p>	<i>Default:</i> "subtree"
User From Name Filter	<p>An LDAP search filter for finding a user given the name of the user. If the attribute (user name attribute and user object class) is not specified (that is, if the attribute is null or empty), a default search filter is created based on the user schema.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean</p> <p><i>Attribute:</i> UserFromNameFilter</p>	<i>Default:</i> "(&(uid=%u)(objectclass=person))"
All Users Filter	<p>An LDAP search filter for finding all users beneath the base user distinguished name (DN). If the attribute (user object class) is not specified (that is, if the attribute is null or empty), a default search filter is created based on the user schema.</p> <p><i>MBean:</i> weblogic.security.providers.authentication.OpenLDAPAuthenticatorMBean</p> <p><i>Attribute:</i> AllUsersFilter</p>	



Define Policy

Use this page to create a security policy for a WebLogic resource. Security policies answer the question "who has access" to a WebLogic resource. A security policy is created when you define an association between a WebLogic resource and a user, group, or role. You can also optionally define a time constraint to a security policy. This association protects the resource the way access control lists (ACLs) protected resources in previous releases of WebLogic Server. A WebLogic resource has no protection until you assign it a security policy.

- For more information, see [Securing WebLogic Resources](#).



Security Realm-->User Management

This tab provides shortcuts to the pages used for configuring users, groups, and global roles for a security realm.



Security Realm-->General

[Tasks](#) [Related Topics](#)

Overview

A security realm provides all the auditing, authentication, authorization, credential mapping, and role mapping services to a WebLogic Server deployment. You can configure multiple security realms within a single WebLogic Server deployment. Use this page to configure a new security realm.

Only one security realm is designated as the default security realm. If you want your newly configured security realm to be the default security realm, click the View Domain-Wide Security Settings link on the General page on the Domain node. Then click the General tab. For more information, see [“Changing the Default Security Realm” on page 428-41](#).

For any security realm to be valid, configure each of the following types of security providers (in any order):

- Authentication
- Authorization
- Adjudication
- Credential Mapping
- Role Mapping

At least one Authorization, Credential Mapping, and Role Mapping provider in the security realm must implement the `DeployableAuthorizationProvider`, `DeployableCredentialProvider`, and `DeployableRoleProvider` Security Service Provider Interface (SSPI). This SSPI allows the providers to store (rather than retrieve) information from deployment descriptors.

To give you control over performance, the WebLogic Server Administration Console requires you to specify how the WebLogic Security Service should perform security checks. You specify this preference using the Check Roles and Policies attribute on the security realm.

When the value of the Check Roles and Policies setting is: Web Applications and EJBs Protected in DD, the WebLogic Security Service only performs security checks on URL and EJB resources that have security specified in their associated deployment descriptors (DDs). This is the default Check Roles and Policies setting.

When the value of the Check Roles and Policies setting is: All Web Applications and EJBs, the WebLogic Security Service performs security checks on all URL (Web) and EJB resources, regardless of whether there are any security settings in the deployment descriptors (DDs) for these WebLogic resources. If you change the value of the Check Roles and Policies drop-down menu to All Web Applications and EJBs, you also need to specify what the WebLogic Security Service should do when the URL or EJB resource is redeployed.

If you decide that the WebLogic Security Service should perform security checks on All Web applications and EJBs in the Check Roles and Policies drop-down menu, you also need to tell WebLogic Server which technique you want to use to secure these URL (Web) and EJB resources. You specify this preference using the Future Redeploys attribute.

You should set the value of the Future Redeploys drop-down menu as follows:

- To secure your URL and EJB resources using only the WebLogic Server Administration Console, select the Ignore Roles and Policies From DD (Deployment Descriptors) option.
- To secure your URL and EJB resources using only the deployment descriptors (that is, the `ejb-jar.xml`, `weblogic-ejb-jar.xml`, `web.xml`, and `weblogic.xml` files), select Initialize roles and policies from DD option.

For more information, see [Securing WebLogic Resources](#).

It is important to understand that once information from a `weblogic-ra.xml` deployment descriptor file is loaded into the embedded LDAP server, the original resource adapter remains unchanged. Therefore, if you redeploy the original resource adapter (which will happen if you redeploy it through the WebLogic Server

Administration Console, modify it on disk, or restart WebLogic Server), the data will once again be imported from the `weblogic-ra.xml` deployment descriptor file and credential mapping information may be lost.

To avoid overwriting new credential mapping information with old information in a `weblogic-ra.xml` deployment descriptor file, enable the Ignore Security Data in Deployment Descriptors attribute.

The Web resource is deprecated in WebLogic Server 7.0 SP02. If you wrote a custom Authorization provider that uses the Web resource (instead of the URL resource), enable the Use Deprecated Web Resource attribute. This attribute changes the runtime behavior of the Servlet container to use a Web resource rather than a URL resource when performing authorization.

Tasks

“Changing the Default Security Realm” on page 428-41

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation



Security Realm-->Migration-->Export

[Tasks](#) [Related Topics](#)

Overview

Use this page to export authentication, authorization, credential map, and security role data from a security realm. Once the security data is exported, use the Security Realm-->Migration-->Import tab to import the data into another security realm. Only security data from the WebLogic security providers can be exported and imported. This page exports all the security data from one realm into another. If you want to export specific security data (for example, if you want to export just user and groups or just roles), use the Import and Export tabs for a specific type of security provider.

Tasks

“Importing and Exporting Security Data from Security Realms” on page 428-38

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Security Realm-->Migration-->Import

[Tasks](#) [Related Topics](#)

Overview

Use this page to import authentication, authorization, credential map, and security role data from one security realm to another. Only data from the WebLogic security providers can be exported and imported. This page imports all the security data from one realm into another. If you want to import specific security data (for example, if you want to import just user and groups or just roles), use the Import and Export tabs for a specific type of security provider.

Tasks

“Importing and Exporting Security Data from Security Realms” on page 428-38

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Adjudication Provider

This tab provides a shortcut to the pages used to configure a Adjudication provider for a security realm. Use the [Configure a new Default Adjudicator...](#) link to configure a Adjudication provider.



Auditing Provider

This tab provides a shortcut to the pages used to configure a Auditing provider for a security realm. Use the [Configure a new Default Auditor...](#) link to configure a Auditing provider.



Authentication Provider

This tab provides a shortcut to the pages used to configure an Authentication or Identity Assertion provider. Use one of the following links to configure an Authentication or Identity Assertion provider for the security realm:

- [Configure a new Active Directory Authenticator...](#)
- [Configure a new Realm Adapter Authenticator...](#)
- [Configure a new Novell Authenticator...](#)
- [Configure a new Default Authenticator...](#)
- [Configure a new Open LDAP Authenticator...](#)
- [Configure a new iPlanet Authenticator...](#)
- [Configure a new Default Identity Asserter...](#)

Note: An Identity Assertion provider uses tokens to authenticate users.



Authorization Provider

This tab provides a shortcut to the pages used to configure a Authorization provider for a security realm. Use the [Configure a new Default Authorizer...](#) link to configure a Authorization provider.



Credential Mapping Provider

This tab provides a shortcut to the pages used to configure a Credential Mapping provider. Use the [Configure a new Default Credential Mapper...](#) link to configure a Credential Mapping provider for the security realm.



Keystores

This tab provides a shortcut to the pages used to configure a WebLogic Keystore provider. Use the [Configure a new Default Keystore...](#) link to configure a WebLogic Keystore provider for the security realm.



Role Mapping Provider

This tab provides a shortcut to the pages used to configure a Role Mapping provider. Use the [Configure a new Default Role Mapper...](#) link to configure a Role Mapping provider for the security realm.



Security Realm-->Testing

[Tasks](#) [Related Topics](#)

Overview

Configuring a new security realm is a complicated task. If you configure a security realm incorrectly, you will not be able to set the security realm as the default security realm. Use this page to validate the configuration of a security realm.

Tasks

“Testing a New Security Realm” on page 1-21

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security](#) page in the WebLogic Server documentation



Security Realm-->UserLockout

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

WebLogic Server provides a set of attributes to protect user accounts from intruders. By default, these attributes are set for maximum protection. As a system administrator, you have the option of turning off all the attributes, increasing the number of login attempts before a user account is locked, increasing the time period in which invalid login attempts are made before locking the user account, and changing the amount of time a user account is locked. Use this page to change these attributes. Remember that changing the attributes on this page lessens security and leaves user accounts vulnerable to security attacks.

If a user lockout security event occurs on one node of a cluster, the other nodes in the cluster are notified of the event and the user account is locked on all nodes in the cluster. This feature prevents a hacker from systematically breaking into all the nodes in a cluster.

Note: The User Lockout attributes apply to the security realm and all its security providers. If you are using an Authentication provider that has its own mechanism for protecting user accounts, disable the Lockout Enabled attribute.

If a user account becomes locked and you delete the user account and add another user account with the same name and password, the UserLockout attribute will not be reset.

Tasks

“Protecting User Accounts” on page 428-5

Related Topics

- [Introduction to WebLogic Security](#)
- [Managing WebLogic Security](#)
- [Securing WebLogic Resources](#)
- [Programming WebLogic Security](#)
- [Developing Security Providers for WebLogic Server](#)
- [Securing a Production Environment](#)
- The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)
- [Security FAQ](#)
- The [Security](#) page in the WebLogic Server documentation

Attributes

Table 405-1

Attribute Label	Description	Value Constraints
Lockout Enabled	<p>Requests the locking of a user account after invalid attempts to log in to that account exceed the specified Lockout Threshold. By default, this attribute is enabled.</p> <p><i>MBean:</i> weblogic.management.security.RealmMBean</p> <p><i>Attribute:</i> LockoutEnabled</p>	

Table 405-1

Attribute Label	Description	Value Constraints
Lockout Threshold	<p>Number of failed user password entries that can be tried before that user 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. Invalid login attempts must be made within a span defined by the Lockout Reset Duration attribute. The default is 5.</p> <p><i>MBean:</i> weblogic.management.security.RealmMBean</p> <p><i>Attribute:</i> LockoutThreshold</p>	
Lockout Duration	<p>Number of minutes that a user's account remains inaccessible after being locked in response to several invalid login attempts within the amount of time specified by the Lockout Reset Duration attribute. The default is 30 minutes.</p> <p><i>MBean:</i> weblogic.management.security.RealmMBean</p> <p><i>Attribute:</i> LockoutDuration</p>	<i>Units:</i> minutes

Table 405-1

Attribute Label	Description	Value Constraints
Lockout Reset Duration	<p>Number of minutes within which invalid login attempts must occur in order for the user's account to be locked.</p> <p>An account is locked if the number of invalid login attempts defined in the Lockout Threshold attribute happens within the amount of time defined by this attribute. For example, if the value in Lockout Reset Duration attribute is 5 minutes, the Lockout Threshold is 3, and 3 invalid login attempts are made within a 6 minute interval, then the account is not locked. If 3 invalid login attempts are made within a 5 minute period, however, then the account is locked.</p> <p>The default is 5 minutes.</p> <p><i>MBean:</i> weblogic.management.security.RealmMBean</p> <p><i>Attribute:</i> LockoutResetDuration</p>	<i>Units:</i> minutes
Lockout Cache Size	<p>Specifies the intended cache size of unused and invalid login attempts. The default is 5.</p> <p><i>MBean:</i> weblogic.management.security.RealmMBean</p> <p><i>Attribute:</i> LockoutCacheSize</p>	
Lockout GCThreshold	<p>The maximum number of invalid login records that the server keeps in memory. If the number of invalid login records is equal to or greater than the value of this attribute, the server's garbage collection purges the records that have expired. A record expires when the user associated with the record have been locked out. The default is 400 records</p> <p><i>MBean:</i> weblogic.management.security.RealmMBean</p> <p><i>Attribute:</i> LockoutGCThreshold</p>	

Table 405-1

Attribute Label	Description	Value Constraints
Invalid Login Attempts Total Count	<p>The total number of invalid logins attempted since the server has been started and since lockouts have been enabled.</p> <p><i>MBean:</i> weblogic.management.security.RealmMBean</p> <p><i>Attribute:</i> InvalidLoginAttemptsTotalCount</p>	
User Lockout Total Count	<p>The total number of user lockouts that have occurred since the server has been started.</p> <p><i>MBean:</i> weblogic.management.security.RealmMBean</p> <p><i>Attribute:</i> UserLockoutTotalCount</p>	
Login Attempts While Locked Total Count	<p>The total number of invalid logins attempted since the server has been started and since lockouts have been enabled.</p> <p><i>MBean:</i> weblogic.management.security.RealmMBean</p> <p><i>Attribute:</i> LoginAttemptsWhileLockedTotalCount</p>	
Invalid Login Users High Count	<p>The highest number of users with concurrent unexpired or uncleared invalid login attempts.</p> <p><i>MBean:</i> weblogic.management.security.RealmMBean</p> <p><i>Attribute:</i> InvalidLoginUsersHighCount</p>	
Locked Users Current Count	<p>The number of users that are currently locked out of the server.</p> <p><i>MBean:</i> weblogic.management.security.RealmMBean</p> <p><i>Attribute:</i> LockedUsersCurrentCount</p>	

Table 405-1

Attribute Label	Description	Value Constraints
Unlocked Users Total Count	The total number of times users have been unlocked since the server has been started. <i>MBean:</i> weblogic.management.security.RealmMBean <i>Attribute:</i> UnlockedUsersTotalCount	

Realm Adapter Adjudication Provider-->General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure a Realm Adapter Adjudication provider.

The Realm Adapter Adjudication provider enables the use of both ACLs and security roles and security policies in Compatibility security. The Realm Adapter Adjudication provider is used to make access decisions in the CompatibilityRealm. The Realm Adapter Adjudication provider is provided by default. It cannot be configured.

Tasks

[“Compatibility Security” on page 322-1](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 406-1

Attribute Label	Description	Value Constraints
Name	The name of this Realm Adapter Adjudication provider.	
Description	A short description of this Realm Adapter Adjudication provider. <i>Attribute: Description</i>	
Version	The version number of this Realm Adapter Adjudication provider. <i>Attribute: Version</i>	

Realm Adapter Adjudication Provider-->Details

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure the Require Unanimous Permit attribute for the Realm Adapter Adjudication provider. The Require Unanimous Permit attribute determines how the Realm Adapter Adjudication provider handles a combination of `PERMIT` and `ABSTAIN` votes from the configured Authorization providers.

- If the attribute is enabled, the Realm Adapter Authorization provider and the WebLogic Authorization provider must both vote `PERMIT` in order for the Adjudication provider to vote `true`.
- If the attribute is disabled, `ABSTAIN` votes are counted as `PERMIT` votes. By default, the Require Unanimous Permit attribute is disabled.

Tasks

[“Compatibility Security” on page 322-1](#)

Related Topics

[Introduction to WebLogic Security](#)
[Managing WebLogic Security](#)
[Securing WebLogic Resources](#)
[Programmimg WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 407-2

Attribute Label	Description	Value Constraints
Name	The name of this Realm Adapter Adjudication provider.	
Description	A short description of this Realm Adapter Adjudication provider. <i>Attribute: Description</i>	
Version	The version number of this Realm Adapter Adjudication provider. <i>Attribute: Version</i>	

Realm Adapter Authentication Provider-->Details

[Tasks](#) [Related Topics](#)

Overview

Use this page to configure the Base 64 Decoding Required attribute for the Identity Assertion provider in the Realm Adapter Authentication provider.

If the authentication type in a Web application is set to `CLIENT-CERT`, the Web Application Container in WebLogic Server performs identity assertion on values from request headers and cookies. If the header name or cookie name matches the active token type for the configured Identity Assertion provider, the value is passed to the provider.

The Base64 Decoding Required attribute determines whether the request header value or cookie value must be Base64 Decoded before sending it to the Identity Assertion provider. The setting is enabled by default for purposes of backward compatibility, however, most Identity Assertion providers will disable this attribute.

Tasks

[“Configuring the Realm Adapter Authentication Provider” on page 428-29](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Realm Adapter Authentication Provider-->General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure a Realm Adapter Authentication provider.

The Realm Adapter Authentication provider allows you to use users and groups from 6.x security realms in this release of WebLogic Server. Use the Realm Adapter Authentication provider if you store users and groups in the 6.x Windows NT, UNIX, RDBMS security realms or 6.x custom security realm. (There are no equivalents to the 6.x Windows NT, UNIX, RDBMS security realms in this release of WebLogic Server). A Realm Adapter Authentication provider can be configured instead of or in addition to the WebLogic Authentication provider.

When using Compatibility Security, a Realm Adapter Authentication provider is by default configured for the CompatibilityRealm. However, you can configure a Realm Adapter Authentication provider in any security realm.

The Realm Adapter Authentication provider also allows use of implementations of the `weblogic.security.acl.CertAuthenticator` class with this release of WebLogic Server. The Realm Adapter Authentication provider includes an Identity Assertion provider which provides identity assertion based on X.509 tokens.

Tasks

[“Configuring the Realm Adapter Authentication Provider” on page 428-29](#)

Related Topics

- [Introduction to WebLogic Security](#)
- [Managing WebLogic Security](#)
- [Securing WebLogic Resources](#)
- [Programming WebLogic Security](#)
- [Developing Security Providers for WebLogic Server](#)
- [Securing a Production Environment](#)
- The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)
- [Security FAQ](#)
- The [Security](#) page in the WebLogic Server documentation

Attributes

Table 409-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this Realm Adapter Authentication provider.</p> <p><i>MBean:</i> weblogic.security.providers.realmadapter.RealmAdapterAuthenticatorMBean</p> <p><i>Attribute:</i> Name</p>	

Table 409-1

Attribute Label	Description	Value Constraints
Description	A short description of this Realm Adapter Authentication provider. <i>MBean:</i> weblogic.security.providers.realmadapter.RealmAdapterAuthenticatorMBean <i>Attribute:</i> Description	<i>Default:</i> "Realm Adapter for Authentication and Identity Assertion"
Version	The version of this security provider. <i>MBean:</i> weblogic.security.providers.realmadapter.RealmAdapterAuthenticatorMBean <i>Attribute:</i> Version	<i>Default:</i> "1.0"

Table 409-1

Attribute Label	Description	Value Constraints
Control Flag	<p>Determines how this Realm Adapter Authentication provider fits unto the login sequence.</p> <p>A <code>REQUIRED</code> value specifies this LoginModule must succeed. Even if it fails, authentication proceeds down the list of LoginModules for the configured Authentication providers. This setting is the default.</p> <p>A <code>REQUISITE</code> value specifies this LoginModule must succeed. If other Authentication providers are configured and this LoginModule succeeds, authentication proceeds down the list of LoginModules. Otherwise, control is return to the application.</p> <p>A <code>SUFFICIENT</code> value specifies this LoginModule need not succeed. If it does succeed, return control to the application. If it fails and other Authentication providers are configured, authentication proceeds down the LoginModule list.</p> <p>An <code>OPTIONAL</code> value specifies this LoginModule need not succeed. Whether it succeeds or fails, authentication proceeds down the LoginModule list.</p> <p><i>MBean:</i> <code>weblogic.security.providers.realmadapter.RealmAdapterAuthenticatorMBean</code></p> <p><i>Attribute:</i> <code>ControlFlag</code></p>	<i>Default:</i> "REQUIRED"

Table 409-1

Attribute Label	Description	Value Constraints
Supported Types	<p>The list of token types supported by the Identity Assertion provider. To see a list of default token types, refer the Javadoc for <code>weblogic.security.spi.IdentityAssertion</code></p> <p><i>MBean:</i> <code>weblogic.security.providers.realmadapter.RealmAdapterAuthenticatorMBean</code></p> <p><i>Attribute:</i> <code>SupportedTypes</code></p>	<p><i>Default:</i> <code>new String[] { weblogic.security.spi.IdentityAssertion.X509_TYPE, weblogic.security.spi.IdentityAssertion.AU_TYPE }</code></p>
Active Types	<p>Specifies what type of token is currently being used by the Identity Assertion provider.</p> <p><i>MBean:</i> <code>weblogic.security.providers.realmadapter.RealmAdapterAuthenticatorMBean</code></p> <p><i>Attribute:</i> <code>ActiveTypes</code></p>	



Realm Adapter Authorization Provider-->General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The Realm Adapter Authorization provider allows you to use access control lists (ACLs) defined in a 6.x security configuration with WebLogic Server deployments using Compatibility security. The Realm Adapter Authorization provider is provided by default in the CompatibilityRealm. It cannot be configured.

Tasks

[“Compatibility Security” on page 322-1](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security](#) page in the WebLogic Server documentation

Attributes

Table 410-1

Attribute Label	Description	Value Constraints
Name	The name of this Realm Adapter Authorization provider. <i>MBean:</i> weblogic.security.providers.authorization.DefaultAuthorizerMBean <i>Attribute:</i> Name	
Description	A short description of the Realm Adapter Authorization provider. <i>MBean:</i> weblogic.security.providers.authorization.DefaultAuthorizerMBean <i>Attribute:</i> Description	
Version	The version number of this security provider. <i>MBean:</i> weblogic.security.providers.authorization.DefaultAuthorizerMBean <i>Attribute:</i> Version	

Realm Adapter Authorization Provider-->Details

[Tasks](#) [Related Topics](#)

Overview

Use this page to set the Policy Deployment Enabled attribute for the Realm Adapter Authorization provider.

The Policy Deployment Enabled attribute specifies whether or not this Authorization provider stores policy information (as opposed to retrieving policy information from a deployment descriptor) for the security realm.

Tasks

[“Compatibility Security” on page 322-1](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Security Realms

This page lists the names of the configured security realms including the default security realm. The default security realm has the `Default Realm` attribute in the table set to `true`. To delete a security realm, click on the trash can icon in the Realms table. Use the [Configure a new Realm...](#) link to create a new security realm. To use the new security realm, it must be set as the default security realm.

- For more information, see [“Changing the Default Security Realm”](#) on page 428-41.



Global Roles

Security roles are abstract, logical collections of users similar to a group. The difference between groups and security roles is that a group is a static identity a server administrator assigns, while membership in a security role is dynamically calculated based on data such as username, group membership, or the time of day. Security roles are granted to individual users or to groups. For more efficient management, BEA recommends granting security roles to groups rather than to individual users. Once you create a security role, you define an association between the security role and a WebLogic resource. This association (called a security policy) specifies who has what access to the WebLogic resource.

Global roles, the types of security roles you work with here, are security roles that apply to all WebLogic resources in a security realm. You can also create scoped roles, or security roles that apply to a specific instance of a WebLogic resource (such as a method of an EJB or a branch of a JNDI tree), by right-clicking the WebLogic resource.

This Global Roles page displays key information about each global security role that has been configured in this security realm.

All the global roles defined in the security realm are listed in the Roles table. The Provider column in the Roles table indicates the Role Mapping provider in which the role is defined.

To delete a global role, click on the trash can icon that is located in the same row as the global role you want to delete.

To define a new global role in the security realm, click the [Configure a new Role...](#) link.

- For more information, see [Securing WebLogic Resources](#).



Security Role-->Conditions

Use this page to grant a user or group a role and apply a time constraint to a role. You can also use this page to revoke a role from a user or group.

Use the Role Condition box to select a condition that must be met for users/groups to be considered "in" this security role.

Click Add to provide the specifics for the Role Statement and if necessary, reorder the role statements using the Move Up/Move Down buttons.

You can also edit or remove role statements by clicking the individual Role Statement and then the Edit... or Remove buttons. Be sure to click Apply when you are finished.

- For more information, see [Securing WebLogic Resources](#).



Security Role-->General

A security role is an abstract, logical collection of users that is similar to a group. The difference between groups and security roles is that a group is a static identity a server administrator assigns, while membership in a security role is dynamically calculated based on data such as username, group membership, or the time of day. Security roles are granted to individual users or to groups, and multiple security roles can be used to create security policies for a WebLogic resource.

A global role is a security role that applies to all WebLogic resources within a security realm.

Use this page to define a global role in this security realm.

- For more information, see [Securing WebLogic Resources](#).



Security Role-->Details

This page is not used in this release of WebLogic Server.



Security Role-->General

A security role is an abstract, logical collection of users that is similar to a group. The difference between groups and security roles is that a group is a static identity a server administrator assigns, while membership in a security role is dynamically calculated based on data such as username, group membership, or the time of day. Security roles are granted to individual users or to groups, and multiple security roles can be used to create security policies for a WebLogic resource.

A global role is a security role that applies to all WebLogic resources within a security realm.

Use this page to define a global role in this security realm.

- For more information, see [Securing WebLogic Resources](#).



Role Mapping Provider

This page displays the name of the default Role Mapping provider for the default security realm (for example, `myrealm`). By default, the WebLogic Role Mapping provider is configured as the default Role Mapping provider. Use the [Configure a new Default Role Mapper...](#) link to configure a different Role Mapping provider as the default.

- For more information, see [“Configuring the WebLogic Role Mapping Provider” on page 428-32](#) or [“Configuring a Custom Security Provider” on page 428-35](#).



Security Realm-->Providers

This tab provides shortcuts to the pages used for configuring security providers for the new security realm.



Users

This page lists all the users defined in the default security realm (for example, `myrealm`). The `Provider` column indicates the Authentication provider in which the user is defined. In order to use the WebLogic Server Administration Console to define users in an Authentication provider, the provider must implement in the `UserEditor` SSPI.

To define a new user in the security realm, click the `Configure a new User...` link on this page.

If you have a large number of users, use the `Filter By` field to retrieve and list only the users that match your search criteria. The `Filter By` field uses the asterisk (*) as the wildcard character.

To delete an existing user, click the trash can icon that is located in the same row as the user you want to delete.

The word `Details` in the `User` table indicates the user account is locked. To unlock the user account, click `Details`.

For more information, see:

- [“Defining Users” on page 428-3](#)
- [“Unlocking a User Account” on page 428-6](#)



User-->Details

This page is not used in this release of WebLogic Server.



User-->General

[Tasks](#) [Related Topics](#)

Overview

Use this page to add users to an Authentication provider. The WebLogic Server Administration Console detects when an Authentication provider implements the UserEditor MBean and automatically loads user information into the provider. The Provider column indicates the Authentication provider in which the user is defined.

Users are entities that can be authenticated in a security realm (such as `myrealm`). A user can be a person or a software entity, such as a Java client. Each user is given a unique identity within the security realm. The minimum password length for users is 8 characters. As a system administrator, you must guarantee that no two users in the same system have the same name. For more efficient management, BEA recommends adding users to groups.

Note: Users and groups must have unique names.

If you have a large number of users, use the Filter By field to retrieve and list only the users that match your search criteria. The Filter By field uses the asterisk (*) as the wildcard character.

All the users defined for a security realm are listed in the User table on the [Users](#) page. If a user account becomes locked, a `Details` link appears in the User table. The `Details` page describes the security violation for the user. For more information, see [“Unlocking a User Account” on page 428-6](#).

Tasks

[“Defining Users” on page 428-3](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security](#) page in the WebLogic Server documentation

User-->Groups

Use this page to add users to the groups available in a security realm (for example, myrealm). A group is an abstract, logical collection of users that has static membership. Groups can be used to define security policies for WebLogic Resources. By default, WebLogic Server has the following groups:

- **Administrators**—View and modify all resource attributes and perform start and stop operations. By default, the user from WebLogic Server is booted is included in this group.
- **Operators**—View all resource attributes and perform server lifecycle operations. By default, this group is empty.
- **Deployers**—View all resource attributes and deploy applications such as EJBs. By default, this group is empty.
- **Monitors**—View all resource attributes, modify resource attributes, and perform operations that are not restricted by a role. By default, this group is empty.

You do not have to use the default groups provided by WebLogic Server. BEA recommends creating groups that more closely reflect your own business structure and practices.

Note: Group and user names must be unique.

For more information, see [“Defining Users” on page 428-3](#).



War Policies and Roles

Use this page to assign scoped roles and security policies to a Web application WAR file.

- For more information, see [Securing Weblogic Resources](#).



Security-->URL Resource-->General

Use this page to assign scoped roles and security policies to a URL resource. This type of WebLogic resource can be a WAR (Web Application aRchive) file or individual components of a Web application (such as servlets and JSPs).

For more information, see [Securing Weblogic Resources](#).



Credential Maps

[Tasks](#) [Related Topics](#)

Overview

A credential map is an association between the credentials used to login to an Enterprise Information System (EIS) and the credentials used to authenticate to a WebLogic resource.

When one or more credential maps are configured for this WebLogic resource, this Credential Maps page displays key information about each of them. To create a new credential map, click the [Configure a new Credential Map...](#) link.

Tasks

[“Creating Credential Maps” on page 428-42](#)

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Security-->Credential Map

[Tasks](#) [Related Topics](#)

Overview

Resource adapters defined by the J2EE Connector Architecture can acquire the credentials necessary to authenticate users defined in an Enterprise Information System (EIS) when they request access to a protected WebLogic resource. The container in WebLogic Server that hosts resource adapters can retrieve the appropriate set of credentials for the WebLogic resource using a credential map. A credential map creates an association between a user in WebLogic Server security realm and an identity (a username and password combination) used to authenticate that user in an EIS such as an Oracle database, a SQL server, or a SAP application.

This page allows you to specify the credential mappings needed for EIS users (referred to as remote users) to access WebLogic Server users to access a

Tasks

“Creating Credential Maps” on page 428-42

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programmimg WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

1 Security

[“Attributes and Console Screen Reference for Security” on page 429-1]

This topic describes configuring and managing security in this release of WebLogic Server. For more information, see [Managing WebLogic Security](#).

For information about configuring and managing security for WebLogic Server deployments using Compatibility security, see “[Compatibility Security](#)” on page 322-1 and [Using Compatibility Security](#) in *Managing WebLogic Security*.

Tasks

The Default Security Configuration in WebLogic Server

To simplify the configuration and management of security in WebLogic Server, a default security realm (myrealm) is provided. The default security realm has WebLogic Authentication, Identity Assertion, Authorization, Adjudication, Role Mapping, and Credential Mapping providers configured. When using the default security configuration, you only need to define groups, users, and security roles for the security realm and create security policies for the WebLogic resources in the domain. You also need to verify that the configuration of the embedded LDAP server configuration is appropriate for your use. Optionally, you can configure an Auditing provider for the default realm.

If the default security configuration does not meet your requirements, you can create a new security realm with any combination of WebLogic and custom security providers and then set the new security realm as the default security realm. For more information, see “[Configuring a New Security Realm](#)” on page 428-18.

Defining Groups

Note: This section applies to Authentication providers that implement the GroupReader Security Service Provider Interface (SSPI) (for example, the WebLogic Authentication provider). If the Authentication provider you are using does not implement this SSPI, you cannot manage users through the WebLogic Server Administration Console.

User and group names must be unique. BEA recommends using initial capitalization and plural names for groups; for example, Administrators.

To define a group:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, myrealm).
3. Click Groups.

The Groups table appears. This table displays the names of all groups defined in the Authentication provider configured in the security realm.

4. Click the Configure a New Group... link.
5. On the Groups-->General page, enter the name of the group.
6. Enter a short description of the group (for example, Product Managers for Code Examples).
7. Click Apply to save your changes.
8. Click the Membership tab to add existing groups to the new group.
 - All available groups appear in the Possible Groups table.
 - All the groups currently defined for a group appear in the Current Groups table.

To add a group to another group, highlight the desired group name and click the right arrow to move the group name to the Current Groups table.

9. Click Apply to save your changes.

Deleting Groups

To delete a group:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, myrealm).
3. Click Groups.

The Groups table appears. This table displays the names of all groups defined in the Authentication provider configured in the security realm.

4. To delete a group, click the trash can icon in the corresponding row of the Groups table.

Defining Users

Note: This section applies to Authentication providers that implement the UserEditor SSPI (for example, the WebLogic Authentication provider). If the Authentication provider you are using does not implement the UserEditor SSPI, you cannot manage users through the WebLogic Server Administration Console.

User and group names must be unique. Do not define a group and a user with the same name. Be sure that there are no spaces or < > characters in the user name. User names are case sensitive.

Do not use the username/password combination `weblogic/weblogic` in production.

To define a user:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, myrealm).
3. Click Users.

The Users table appears. This table displays the names of all users defined in the Authentication provider.

4. Click the Configure a New User... link.
5. On the User-->General page, enter the name of the user.
6. Enter a password for the user.

Note: The minimum password length for a user defined in the WebLogic Authentication provider is 8 characters. However, password rules (for example, length and type of characters) vary by Authentication provider.

7. Re-enter the password for the user in the Confirm Password field.
8. Click Apply to save your changes.

Note: For more efficient management, BEA recommends adding users to groups.

9. Click the Groups tab.
10. In the Possible Groups list box, click the name of a group to highlight it.
11. Click the right arrow to move the group to the Current Groups list box.
12. If desired, repeat steps 6 and 7 to add the user to multiple groups.
13. Click Apply to save your changes.

Deleting Users

To delete a user:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, myrealm).
3. Click Users.

The Users table appears. This table displays the names of all users defined in the Authentication provider.

4. To delete a user, click the trash can icon in the corresponding row of the Users table.

Changing the Password of a User

Note: The minimum password length for a user defined in the WebLogic Authentication provider is 8 characters. However, password rules (for example, length and type of characters) vary by Authentication provider.

To change the password of a user:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, myrealm).
3. Click Users.
4. Select a user.
5. Click the Change... link in the Password attribute.
6. Enter a password for the user.
7. Click Apply.

Protecting User Accounts

Weblogic Server provides a set of attributes to protect user accounts from intruders. By default, these attributes are set for maximum protection. As a system administrator, you have the option of turning off all the attributes, increasing the number of login attempts before a user account is locked, increasing the time period in which invalid login attempts are made before locking the user account, and changing the amount of time a user account is locked. Remember that changing the attributes lessens security and leaves user accounts vulnerable to security attacks.

To set the User Lockout attributes:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, myrealm).
3. Select the User Lockout tab.
4. Configure the attributes on this page by entering values at the appropriate prompts and selecting the required checkboxes.

5. To save your changes, click Apply.
6. Reboot WebLogic Server.

Unlocking a User Account

To unlock a user account:

1. Expand the Servers-->Monitoring tab.
2. Click the Security tab.
3. Enter the user name for a user of this server who has been locked out in the Unlock User attribute.
4. Click Apply.

If the unlock was successful, a confirmation message appears at the top of the Monitoring-->Security page.

Defining Global Roles

A security role that applies to all WebLogic resources deployed within a security realm (and thus the entire WebLogic Server domain) is called a global role.

This topic highlights the process for creating global roles. However, creating global roles is a multi-step process with many options. To fully understand this process, read *Securing WebLogic Resources*.

Note: BEA recommends using initial capitalization, singular names for global roles; for example, SecurityEng.

To define a global role:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, myrealm).
3. Click Roles.

The Select Roles page appears. This page displays all the global roles currently defined in the WebLogic Role Mapping provider's database.

4. Click the Configure a New Role... link.

The Create Role page appears.

5. On General page, enter the name of the global role in the Name field.

Notes: Be sure that there are no spaces or < > characters in the security role name. Security role names are case sensitive. The BEA convention is that all security role names are singular.

6. Click the Apply button to save your changes.
7. Click the Conditions tab.
8. In the Role Condition list box, click one of the conditions.

Note: BEA recommends that you create expressions using the `Caller is a Member of the Group` condition. When a group is used to create a security role, the security role can be granted to all members of the group (that is, multiple users).

9. Click the Add button. A customized window appears.
10. If you selected the `Hours of Access are Between` condition, use the Time Constraint window to select start and end times, and then click the OK button.

If you selected one of the other conditions, follow these steps:

- a. Use the Users or Groups window to enter the name of a user or group, and then click the Add button.

Note: You can repeat this step multiple times to add more than one user or group.

- b. If necessary, use the buttons located to the right of the list box to modify the expressions.

The Move Up and Move Down buttons change the ordering of the highlighted user or group name. The Change button switches the highlighted `and` and `or` statements between expressions. The Remove button deletes the highlighted user or group name.

- c. Click OK to add the expression to the role statement.

11. If desired, repeat steps 8-10 to add expressions based on different role conditions.
12. If necessary, use the buttons located to the right of the Role Statement list box to modify the expressions:
 - The Move Up and Move Down buttons change the ordering of the highlighted expression.
 - The Change button switches the highlighted `and` and `or` statements between expressions.
 - The Edit... button reopens the customized window for the highlighted expression and allows you to modify the expression.
 - The Remove button deletes the highlighted expression.
13. When all the expressions in the Role Statement list box are correct, click the Apply button.

Note: Clicking the Reset button will delete all expressions shown in the Role Statement list box.

Deleting Global Roles

To delete a global role:

1. Expand the Security --> Realms nodes.
2. Click the name of the realm you are configuring (for example, myrealm).
3. Click Roles.

The Select Roles page appears. This page displays all the global roles currently defined in the WebLogic Role Mapping provider's database.

4. To delete a role, click the trash can icon in the corresponding row of the Select Roles page.

A confirmation window appears.

5. Click Yes to delete the global role.

Defining Scoped Roles

A security role that applies to a specific instance of a WebLogic resource deployed in a security realm (such as a method on an EJB or a branch of a JNDI tree) is called a scoped role. The procedure for creating scoped roles differs slightly, depending on the type of WebLogic resource and level at which you want to scope the security role. Use the appropriate instructions provided in the following sections:

- “URL (Web) Resources” on page 428-9
- “Enterprise JavaBean (EJB) Resources” on page 428-12
- “JNDI Resources” on page 428-14
- “Other Types of WebLogic Resources” on page 428-16

For a complete description of defining scoped roles for WebLogic resources, see *Securing WebLogic Resources*.

URL (Web) Resources

To create a scoped role for a URL (Web) resource, follow these steps:

1. Using the navigation tree at the left side of the WebLogic Server Administration Console, click the + sign next to Deployments.

The Deployments node expands to show the types of WebLogic resources that can be deployed.

2. Click the right mouse button at the level of the URL (Web) resource at which you want to create the scoped role.

A menu of options appears.

- To create a scoped role for *all* Web applications (WARs), click the right mouse button on Web Applications.
 - To create a scoped role for a *particular* WAR or a component in a WAR (for example, a specific servlet or JSP), click the + sign next to Web Applications, then click the right mouse button on the name of a Web application (WAR).
3. If you are creating the scoped role for all Web applications (WARs), select the Define Role... option.

If you are creating the scoped role for a particular WAR, or a component within a WAR, follow these steps:

- a. Select the Define Role... option.

The General page appears.

- b. Enter a URL pattern in the text field.

A URL pattern is a path to a specific component within a Web application. Or, you can use /* to associate the scoped role with all components (servlets, JSPs, and so on) within the Web application.

- c. Click the Define Role... button to proceed. The Select Roles page appears. If any are available, this page displays the scoped roles that are currently defined for this WebLogic resource in the WebLogic Role Mapping provider's database.

4. Click the Configure a New Role... link.

The Create Role page appears

5. On General page, enter the name of the scoped role in the Name field.

Notes: Be sure that there are no spaces or <> characters in the security role name. Role names are case sensitive. The BEA convention is that all security role names are singular.

Warning: If you create a scoped role with the same name as a global role, the scoped role takes precedence over the global role.

6. Click the Apply button to save your changes.

7. Click the Conditions tab.

The Role Editor page appears.

8. In the Role Condition list box, click one of the conditions.

Note: BEA recommends that you create expressions using the `Caller is a Member of the Group` condition. When a group is used to create a security role, the security role can be granted to all members of the group (that is, multiple users).

9. Click the Add button.

10. If you selected the `Hours of Access are Between` condition, use the `Time Constraint` window to select start and end times, and then click the `OK` button.

If you selected one of the other conditions, follow these steps:

- a. Use the `Users or Groups` window to enter the name of a user or group, and then click the `Add` button.

Note: You can repeat this step multiple times to add more than one user or group.

- b. If necessary, use the buttons located to the right of the list box to modify the expressions.

The `Move Up` and `Move Down` buttons change the ordering of the highlighted user or group name. The `Change` button switches the highlighted `and` and `or` statements between expressions. The `Remove` button deletes the highlighted user or group name.

- c. Click `OK` to add the expression to the role statement.

11. If desired, repeat steps 8 - 10 to add expressions based on different role conditions.

12. If necessary, use the buttons located to the right of the `Role Statement` list box to modify the expressions:

- The `Move Up` and `Move Down` buttons change the ordering of the highlighted expression.
- The `Change` button switches the highlighted `and` and `or` statements between expressions.
- The `Edit...` button reopens the customized window for the highlighted expression and allows you to modify the expression.
- The `Remove` button deletes the highlighted expression.

13. When all the expressions in the `Role Statement` list box are correct, click the `Apply` button.

Note: Clicking the `Reset` button will delete all expressions shown in the `Role Statement` list box.

Enterprise JavaBean (EJB) Resources

To create a scoped role for an EJB resource, follow these steps:

1. Using the navigation tree at the left side of the WebLogic Server Administration Console, click the + sign next to Deployments.

The Deployments node expands to show the types of WebLogic resources that can be deployed.

2. Click the right mouse button at the level of the EJB resource for which you want to create the scoped role. .

To create a scoped role for *all* EJB JARs, click the right mouse button on EJB. To create a scoped role for a *particular* EJB JAR, or for an EJB within a JAR, click the + sign next to EJB, then click the right mouse button on the name of an EJB JAR.

3. If you are creating the scoped role for all EJB JARs or for a particular EJB JAR, select the Define Role... option.

The Select Roles page appears. If any are available, this page displays the scoped roles that are currently defined for this WebLogic resource in the WebLogic Role Mapping provider's database.

If you are creating the scoped for a particular EJB within an EJB JAR, follow these steps:

- a. Select the Define Policies and Roles for Individual Beans... option. A list of EJBs appears.
 - b. Click the [Define Roles] link that is located in the same row as the particular EJB for which you want to create the scoped role. The Select Roles page appears. If any are available, this page displays the scoped roles that are currently defined for this WebLogic resource in the WebLogic Role Mapping provider's database.
4. Click the Configure a New Role... link.

The Create Role page appears

5. On General page, enter the name of the scoped role in the Name field.

Notes: Be sure that there are no spaces or <> characters in the security role name. Security role names are case sensitive. The BEA convention is that all security role names are singular.

Warning: If you create a scoped role with the same name as a global role, the scoped role takes precedence over the global role.

6. Click the Apply button to save your changes.
7. Click the Conditions tab.

The Role Editor page appears.

8. In the Role Condition list box, click one of the conditions.

Note: BEA recommends that you create expressions using the `Caller is a Member of the Group` condition. When a group is used to create a security role, the security role can be granted to all members of the group (that is, multiple users).

9. Click the Add button.

10. If you selected the `Hours of Access are Between` condition, use the Time Constraint window to select start and end times, and then click the OK button.

If you selected one of the other conditions, follow these steps:

- a. Use the Users or Groups window to enter the name of a user or group, and then click the Add button.

Note: You can repeat this step multiple times to add more than one user or group.

- b. If necessary, use the buttons located to the right of the list box to modify the expressions.

The Move Up and Move Down buttons change the ordering of the highlighted user or group name. The Change button switches the highlighted `and` and `or` statements between expressions. The Remove button deletes the highlighted user or group name.

- c. Click OK to add the expression to the role statement.

11. If desired, repeat steps 8 - 10 to add expressions based on different role conditions.

12. If necessary, use the buttons located to the right of the Role Statement list box to modify the expressions:

- The Move Up and Move Down buttons change the ordering of the highlighted expression.

- The Change button switches the highlighted `and` and `or` statements between expressions.
 - The Edit... button reopens the customized window for the highlighted expression and allows you to modify the expression.
 - The Remove button deletes the highlighted expression.
13. When all the expressions in the Role Statement list box are correct, click the Apply button.
- Note:** Clicking the Reset button will delete all expressions shown in the Role Statement list box.

JNDI Resources

To create a scoped role for a JNDI resource, follow these steps:

1. Using the navigation tree at the left side of the WebLogic Server Administration Console, click the + sign next to Servers.

The Servers node expands to show the servers available in the current WebLogic Server domain.
 2. Click the *right* mouse button on the name of the server that contains the JNDI resource for which you want to create the scoped role. (For example, `myserver`.)
 3. Select the View JNDI Tree option.
 4. Click the *right* mouse button at the level of the JNDI tree at which you want to create the scoped role.
 5. Select the Define Role... option.
 6. Click the Configure a New Role... link.

The Create Role page appears. If any are available, this page displays the scoped roles that are currently defined for this WebLogic resource in the WebLogic Role Mapping provider's database.
 7. On General page, enter the name of the scoped role in the Name field.
- Notes:** Be sure that there are no spaces or `<>` characters in the security role name. Security role names are case sensitive. The BEA convention is that all security role names are singular.

Warning: If you create a scoped role with the same name as a global role, the scoped role takes precedence over the global role.

8. Click the Apply button to save your changes.
9. Click the Conditions tab. The Role Editor page appears.
10. In the Role Condition list box, click one of the conditions.

Note: BEA recommends that you create expressions using the `Caller is a Member of the Group` condition. When a group is used to create a security role, the security role can be granted to all members of the group (that is, multiple users).

11. Click the Add button.
12. If you selected the `Hours of Access are Between` condition, use the Time Constraint window to select start and end times, and then click the OK button.

If you selected one of the other conditions, follow these steps:

- a. Use the Users or Groups window to enter the name of a user or group, and then click the Add button.

Note: You can repeat this step multiple times to add more than one user or group.

- b. If necessary, use the buttons located to the right of the list box to modify the expressions.

The Move Up and Move Down buttons change the ordering of the highlighted user or group name. The Change button switches the highlighted `and` and `or` statements between expressions. The Remove button deletes the highlighted user or group name.

- c. Click OK to add the expression to the role statement.

13. If desired, repeat steps 10 - 12 to add expressions based on different role conditions.

14. If necessary, use the buttons located to the right of the Role Statement list box to modify the expressions:

- The Move Up and Move Down buttons change the ordering of the highlighted expression.

- The Change button switches the highlighted `and` and `or` statements between expressions.
 - The Edit... button reopens the customized window for the highlighted expression and allows you to modify the expression.
 - The Remove button deletes the highlighted expression.
15. When all the expressions in the Role Statement list box are correct, click the Apply button.
- Note:** Clicking the Reset button will delete all expressions shown in the Role Statement list box.

Other Types of WebLogic Resources

With the exception of Web Services resources, you can create scoped roles for the other types of WebLogic resources using the WebLogic Server Administration Console. However, not all WebLogic resource types are listed under the Deployments node in the WebLogic Server Administration Console's navigation tree, and not all of the WebLogic resource types allow scoped roles to be created at the same levels in the resource hierarchy. JDBC connection pools, for example, are shown *under* the Services→JDBC node, and scoped roles for JMS resources may only be created *at* the Services→JMS node. Therefore, you will need to adapt the instructions provided in the previous sections to create scoped roles for other WebLogic resource types, as the process for accomplishing this task differs only in small ways.

Deleting Scoped Roles

To delete a scoped role, follow these steps:

1. Navigate to the Select Roles page for your WebLogic resource.

This page displays all the scoped roles currently defined in the WebLogic Role Mapping provider's database.
2. Click the trash can icon that is located in the same row as the scoped role you want to delete.
3. Click the Yes button.

Click the Continue link.

Protecting WebLogic Resources

Security policies are used to protect WebLogic resources. A security policy is created when you define an association between a WebLogic resource and a user, group, or security role. You can also optionally associate a time constraint with a security policy. A WebLogic resource has no protection until you assign it a security policy.

Creating security policies is a multi-step process with many options. To fully understand this process, read [Securing WebLogic Resources](#).

Configuring the Embedded LDAP Server

The embedded LDAP server contains user, group, group membership, security role, security policy, and credential map information. By default, each WebLogic Server domain has an embedded LDAP server configured with the default values set for each attribute. The WebLogic Authentication, Authorization, Credential Mapping, and Role Mapping providers use the embedded LDAP server as their database. If you use any of these providers in a new security realm, you may want to change the default values for the embedded LDAP server to optimize its use in your environment.

To configure the embedded LDAP server:

1. Expand the Domain node (for example, Examples).
2. Click the View Domain-Wide Security Settings link on the Domain-->General page.
3. Select the Security Configuration-->Embedded LDAP tab.
4. Set attributes on the Embedded LDAP Server page.
5. Click Apply to save your changes.
6. Reboot WebLogic Server.

Note: The WebLogic Security providers stored their data in the embedded LDAP server. When you delete a WebLogic Security provider, the security data in the embedded LDAP server is not automatically deleted. The security data

remains in the embedded LDAP server in case you want to use the provider again. Use an external LDAP browser to delete the security data from the embedded LDAP server.

Configuring Backups for the Embedded LDAP Server

To configure the backups of the embedded LDAP server:

1. Expand the Domain node (for example, Examples).
2. Click the View Domain-Wide Security Settings link on the Domain-->General page.
3. Click the Security Configuration-->the Embedded LDAP tab.
4. Set the Backup Hour, Backup Minute, and Backup Copies attributes on the Embedded LDAP Server page.
5. Click Apply to save your changes.
6. Reboot WebLogic Server.

Configuring a New Security Realm

To configure a new security realm:

1. Expand the Security node.
2. Expand the Realms node.

All the security realms available for the WebLogic domain are listed in the Realms table.
3. Click the Configure a new Realm... link.
4. Enter the name of the new security realm in the Name attribute on the General page.
5. Set the Check Roles and Security Policies attribute. The following options are available:

- Web Applications and EJBs Protected in DD—This option specifies that the WebLogic Security Service only performs security checks on URL and EJB resources that have security specified in their associated deployment descriptors (DDs). This option is the default Check Roles and Policies setting.
 - All Web Applications and EJBs—This option specifies that the WebLogic Security Service performs security checks on all URL (Web) and EJB resources, regardless of whether there are any security settings in the deployment descriptors (DDs) for these WebLogic resources. If you change the setting of the Check Roles and Policies drop-down menu to All Web Applications and EJBs, specify the Future Redeploys attribute as described in Step 6.
6. Use the Future Redeploys attribute to tell WebLogic Server how URL and EJB resources are to be secured. The following options are provided:
 - To secure URL and EJB resources using only the WebLogic Server Administration Console, select the Ignore Roles and Policies From DD (Deployment Descriptors) option.
 - To secure URL and EJB resources using only the deployment descriptors (that is, the `ejb-jar.xml`, `weblogic-ejb-jar.xml`, `web.xml`, and `weblogic.xml` files), select Initialize roles and policies from DD option.
 7. You have the option of loading credential maps from `weblogic-ra.xml` deployment descriptor files into the embedded LDAP server and then using the WebLogic Server Administration Console to create new credential maps or modify existing credential maps.

Once information from a `weblogic-ra.xml` deployment descriptor file is loaded into the embedded LDAP server, the original resource adapter remains unchanged. Therefore, if you redeploy the original resource adapter (which will happen if you redeploy it through the WebLogic Server Administration Console, modify it on disk, or restart WebLogic Server), the data will once again be imported from the `weblogic-ra.xml` deployment descriptor file and credential mapping information may be lost.

To avoid overwriting new credential mapping information with old information in a `weblogic-ra.xml` deployment descriptor file, enable the Ignore Security Data in Deployment Descriptors attribute.

Note: To use load credential maps into the embedded LDAP server, the Credential Mapping provider in the security realm must have the Credential Mapping Deployment Enabled attribute checked. For more information, see “Configuring the WebLogic Credential Mapping Provider” on page 428-31.

8. The Web resource was deprecated in a previous release of WebLogic Server. If you wrote a custom Authorization provider that uses the Web resource (instead of the URL resource), enable the Use Deprecated Web Resource attribute. This attribute changes the runtime behavior of the Servlet container to use a Web resource rather than a URL resource when performing authorization.
9. Click Create.
10. Configure the required security providers for the security realm. In order for a security realm to be valid, you must configure an Authentication provider, an Authorization provider, an Adjudication provider, a Credential Mapping provider, and a Role Mapping provider. Otherwise, you will not be able to set the new security realm as the default security realm.
11. Optionally, define an Identity Assertion and Auditing provider.
12. Define groups and users for the security realm. For more information, see “Defining Groups” on page 428-2 and “Defining Users” on page 428-3.
13. Grant users and groups in the security realm roles. For more information, see “Defining Global Roles” on page 428-6
14. Protect WebLogic resources in the security realm with security policies. For more information, see [Securing WebLogic Resources](#).
15. Reboot WebLogic Server. If you do not reboot WebLogic Server, you cannot set the realm to the default security realm.
16. Set the new realm as the default security realm for the WebLogic domain. For more information, see “Changing the Default Security Realm” on page 428-41.

Testing a New Security Realm

Configuring a new security realm is a complicated task. If you configure a security realm incorrectly, you will not be able to set the security realm as the default security realm. WebLogic Server can validate the configuration of a security realm to ensure it is correct.

To validate the configuration of a new security realm:

1. Configure the security realm as described in “Configuring a New Security Realm” on page 428-18.
2. Expand the Security-->Realms nodes.
The Realms table shows all security realms configured for the WebLogic Server domain.
3. Click the realm you want to validate.
4. Click the Testing tab.
5. Click the Validate this realm... link.

Any problems with the configuration of the security realm are displayed on the Testing page.

Configuring an Authentication Provider: Main Steps

WebLogic Server offers the following types of Authentication providers:

- *The WebLogic Authentication provider* allows you to manage users and groups in one place, the embedded LDAP server. For more information, see “Configuring the WebLogic Authentication Provider” on page 428-24.
- *LDAP Authentication providers* access external LDAP stores. WebLogic Server provides LDAP Authentication providers which access Open LDAP, Netscape iPlanet, Microsoft Active Directory and Novell NDS stores. For more information, see “Configuring an LDAP Authentication Provider” on page 428-25.

Note: You are not limited to these LDAP Authentication providers. To use an LDAP server other than the supported LDAP servers, choose the LDAP server type that has the closest defaults to the LDAP server you want to use and modify the attribute values accordingly.

- *The Realm Adapter Authentication provider* accesses user and group information stored in 6.x security realms. For more information, see “Configuring the Realm Adapter Authentication Provider” on page 428-29.

In addition, you can use a Custom Authentication provider which offers different types of authentication technologies. For more information, see “Configuring a Custom Security Provider” on page 428-35.

Note: The WebLogic Server Administration Console refers to the WebLogic Authentication provider as the Default Authenticator.

Each security realm must have one at least one Authentication provider configured. The WebLogic Security Framework is designed to support multiple Authentication providers (and thus multiple LoginModules) for multipart authentication. Therefore, you can use multiple Authentication providers as well as multiple types of Authentication providers in a security realm. The Control Flag attribute determines how the LoginModule for each Authentication provider is used in the authentication process. For more information, see “Setting the JAAS Control Flag” on page 428-23.

To configure an Authentication provider:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, TestRealm).
3. Expand the Providers-->Authentication Providers nodes.
4. Choose an Authentication provider by selecting the appropriate link.
 - Configure a new Active Directory Authenticator...
 - Configure a new Realm Adapter Authenticator...
 - Configure a new Novell Authenticator...
 - Configure a new iPlanet Authenticator...
 - Configure a new Default Authenticator...
 - Configure a new OpenLDAP Authenticator...

5. Go to the appropriate sections to configure an Authentication provider.
 - “Configuring an LDAP Authentication Provider” on page 428-25
 - “Configuring the Realm Adapter Authentication Provider” on page 428-29
 - “Configuring the WebLogic Authentication Provider” on page 428-24
6. Repeat these steps to configure additional Authentication providers.

If you are configuring multiple Authentication providers, refer to “Setting the JAAS Control Flag” on page 428-23.
7. After you finish configuring Authentication providers, reboot WebLogic Server.

Setting the JAAS Control Flag

If a security realm has multiple Authentication providers configured, the Control Flag attribute on the Authenticator-->General page determines the ordered execution of the Authentication providers. The values for the Control Flag attribute are as follows:

- **REQUIRED**—The Authentication provider is always called, and the user must always pass its authentication test.
- **REQUISITE**—If the user passes the authentication test of this Authentication provider, other providers are executed but can fail (except for Authentication providers with the JAAS Control Flag set to **REQUIRED**).
- **SUFFICIENT**—If the user passes the authentication test of the Authentication provider, no other Authentication providers are executed (except for Authentication providers with the JAAS Control Flag set to **REQUIRED**) because the user was sufficiently authenticated.
- **OPTIONAL**—The user is allowed to pass or fail the authentication test of this Authentication provider. However, if all Authentication providers configured in a security realm have the JAAS Control Flag set to **OPTIONAL**, the user must pass the authentication test of one of the configured providers.

The order in which the Authentication providers are configured is the order in which in the LoginModules for the Authentication providers are called. The ordering of the Authentication providers can be changed at any time. For more information, see “Changing the Order of Authentication Providers” on page 428-30.

Notes: If you define multiple Authentication providers, in order to boot WebLogic Server, the user from which the server is booted must be defined as a user in all the Authentication providers than have the Control Flag attribute set to REQUISITE or REQUIRED.

The WebLogic Server Administration Console actually sets the JAAS Control Flag to OPTIONAL when creating a security provider. MBeans for the security providers actually default to REQUIRED.

Configuring the WebLogic Authentication Provider

Note: The WebLogic Server Administration Console refers to the WebLogic Authentication provider as the Default Authenticator.

The WebLogic Authentication provider is case insensitive. Ensure user names are unique.

The WebLogic Authentication provider allows you to edit, list, and manage users and group membership. User and group membership information for the WebLogic Authentication provider is stored in the embedded LDAP server.

To configure the WebLogic Authentication provider:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, TestRealm).
3. Expand the Providers-->Authentication Providers nodes.
4. Choose the Configure a new Default Authenticator... link.
5. Define values for the attributes on the General page.
6. Click Apply to save your changes.
7. Define values on the Details page.
8. Optionally, configure additional Authentication providers.
9. Reboot WebLogic Server.

Configuring an LDAP Authentication Provider

To configure an LDAP Authentication provider:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, TestRealm).
3. Expand the Providers-->Authentication Providers nodes.
4. Choose an LDAP Authentication provider from the following available links:
 - Configure a new Active Directory Authenticator...
 - Configure a new Novell Authenticator...
 - Configure a new OpenLDAP Authenticator...
 - Configure a new iPlanet Authenticator...
5. If you using multiple Authentication providers, define a value for the Control Flag attribute on the General page. For more information, “Setting the JAAS Control Flag” on page 428-23.
6. Click Apply to create a new LDAP Authentication provider.
7. Proceed to “Setting LDAP Server and Caching Information” on page 428-25.

Setting LDAP Server and Caching Information

To set LDAP server and caching information:

1. Click the LDAP tab under the Configuration tab for the LDAP Authentication provider you want to use.

For example, click the iPlanet LDAP tab under the iPlanet Configuration tab.
2. Enable communication between WebLogic Server and the LDAP server by defining values for the attributes shown on the LDAP page.
3. To save your changes, click Apply.
4. Click the Details tab to configure additional attributes that control the behavior of the LDAP server. The following attributes are available:

- **Follow Referrals**—Specifies that a search for a user or group within the Active Directory Authentication provider will follow referrals to other LDAP servers or branches within the LDAP directory. By default, this attribute is enabled.
- **Bind Anonymously On Referrals**—By default, an LDAP Authentication provider uses the same DN and password used to connect to the LDAP server when following referrals during a search. If you want to connect as an anonymous user, enable this attribute. Contact your LDAP system administrator for more information.
- **Results Time Limit**—The maximum number of milliseconds for the LDAP server to wait for results before timing out. If this attribute is set to 0, there is not maximum time limit. The default is 0.
- **Connect Timeout**—The maximum time in seconds to wait for the connection to the LDAP server to be established. If this attribute is set to 0, there is not a maximum time limit. The default is 0.
- **Parallel Connect Delay**—The delay in seconds when making concurrent attempts to attempt to multiple LDAP servers. If this attribute is set to 0, connection attempts are serialized. An attempt is made to connect to the first server in the list. The next entry in the list is tried only if the attempt to connect to the current host fails. If this attribute is not set and an LDAP server is unavailable, an application may be blocked for a long time. If this attribute is greater than 0, another connection is started after the specified time.

5. To save your changes, click **Apply**.

6. Proceed to “Locating Users in the LDAP Directory” on page 428-26.

For a more secure deployment, BEA recommends using the SSL protocol to protect communications between the LDAP server and WebLogic Server.

Locating Users in the LDAP Directory

To specify how users are located in the LDAP directory:

1. Click the **Users** tab under the **Configuration** tab for the LDAP server you chose.

For example, click the **Users** tab under the **iPlanet Configuration** tab.

2. Define information about how users are stored and located in the LDAP directory by defining values for the attributes shown on the Users page.
3. To save your changes, click Apply.
4. Proceed to “Locating Groups in the LDAP Directory” on page 428-27.

Locating Groups in the LDAP Directory

To specify how groups are stored and located in the LDAP directory:

1. Click the Groups tab under the Configuration tab.
For example, click the Groups tab under the iPlanet Configuration tab.
2. Define information about how groups are stored and located in the LDAP directory by defining values for the attributes shown on the Groups page.
3. To save your changes, click Apply.
4. Proceed to “Locating Members of a Group in the LDAP Directory” on page 428-27.

Locating Members of a Group in the LDAP Directory

Note: The iPlanet Authentication provider supports dynamic groups. To use dynamic groups, set the Dynamic Group Object Class, Dynamic Group Name Attribute, and Dynamic Member URL Attribute attributes on the Members page.

To specify how groups members are stored and located in the LDAP directory:

1. Click on the Membership tab under the Configuration tab.
For example, click the Membership tab under the iPlanet Configuration tab.
2. Define information about how group members are stored and located in the LDAP directory by defining values for the attributes shown on the Membership page.
3. To save your changes, click Apply.

4. Optionally, configure additional Authentication and/or Identity Assertion providers.
5. Reboot WebLogic Server.

Configuring Failover for LDAP Authentication Providers

To configure failover of the LDAP servers configured for an LDAP Authentication provider, perform the following steps:

1. Click the LDAP tab under the Configuration tab for the LDAP Authentication provider for which you want to configure failover.

For example, click the iPlanet LDAP tab under the iPlanet Configuration tab.

2. Click the LDAP tab.
3. Specify more than one LDAP server name in the Host attribute on the LDAP tab. The attribute must contain a space-delimited list of host names. Each host name may include a trailing colon and port number. For example:

```
directory.knowledge.com:1050 people.catalog.com 199.254.1.2
```

4. Click Apply.
5. Click the Details tab.
6. Set the Parallel Connect Delay attribute.

The Parallel Connect Delay attribute specifies the number of seconds to delay when making concurrent attempts to connect to multiple servers. An attempt is made to connect to the first server in the list. The next entry in the list is tried only if the attempt to connect to the current host fails. This setting might cause your application to block for unacceptably long time if a host is down. If the attribute is set to a value greater than 0, another connection setup thread is started after the specified number of delay seconds has passed. If the attribute is set to 0, connection attempts are serialized.

7. Set the Connection Timeout attribute.

The Connection Timeout attribute specifies the maximum number of seconds to wait for the connection to the LDAP server to be established. If the attribute is set to 0, there is no maximum time limit and WebLogic Server will wait until the

TCP/IP layer times out to return a connection failure. This attribute may be set to a value over 60 seconds depending upon the configuration of TCP/IP.

8. Click Apply.
9. Reboot WebLogic Server.

Configuring the Realm Adapter Authentication Provider

To configure the Realm Adapter Authentication provider:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, TestRealm).
3. Expand the Providers-->Authentication Providers nodes.

The Authenticators table displays the name of the default Authentication and Identity Assertion providers.
4. Choose the Configure a new Realm Adapter Authenticator... link.
5. Set the Control Flag for the Realm Adapter Authentication provider. For more information, see “Setting the JAAS Control Flag” on page 428-23.
6. Set the Active Type for the Identity Asserter in the Realm Adapter Authentication provider.
 - a. In the Available list box, click X.509 to highlight it.
 - b. Click the right arrow to move X.509 to the Chosen list box.
7. Click Apply to save your changes.
8. Optionally, configure additional Authentication and/or Identity Assertion providers.
9. Reboot WebLogic Server.

Changing the Order of Authentication Providers

The way you configure multiple Authentication providers can affect the overall outcome of the authentication process, which is especially important for multipart authentication. Authentication providers are called in the order in which they are configured. The Authentication Providers table lists the authentication providers in the order they were configured. Click the Re-order the Configured Authentication Providers... link to change the order of the providers. Be aware that the way each Authentication provider's Control Flag attribute is set effects the outcome of the authentication process. For more information, see "Setting the JAAS Control Flag" on page 428-23.

To change the ordering of Authentication providers:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, TestRealm).
3. Expand the Providers-->Authentication Providers nodes.
4. Choose the Re-order the Configured Authentication Providers... link.
5. Select an Authentication provider from the list of configured Authentication providers.
6. Use the arrow buttons to move it up or down in the list.
7. Click Apply to save your changes.
8. Reboot WebLogic Server.

Configuring the WebLogic Authorization Provider

Note: The WebLogic Server Administration Console refers to the WebLogic Authorization provider as the Default Authorizer.

To configure the WebLogic Authorization provider:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, TestRealm).

3. Expand the Providers node.
4. Click Authorizers.
5. Click the Configure a new Default Authorizer... link.
6. Define values for the attribute on the General page.
7. Click Apply to save your changes.
8. Define values for the attribute on the Details tab.
9. Reboot WebLogic Server.

Configuring the WebLogic Credential Mapping Provider

To configure the WebLogic Credential Mapping provider:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, TestRealm).
3. Expand the Providers node.
4. Click Credential Mappers.
5. Click the Configure a new Default Credential Mapper... link.
6. On the General page, set the Credential Mapping Deployment Enabled attribute.

The Credential Mapping Deployment Enabled attribute specifies whether or not this Credential Mapping provider imports credential maps from a `weblogic-ra.xml` deployment descriptor file. In order to support the Credential Mapping Deployment Enabled attribute, a Credential Mapping provider must implement the `DeployableCredentialProvider` SSPI. By default, the WebLogic Credential Mapping provider has this attribute enabled. The credential mapping information is stored in the embedded LDAP server.

7. Click Apply to save your changes.
8. Reboot WebLogic Server.

Configuring the WebLogic Role Mapping Provider

To configure an Role Mapping provider:

1. Expand the Security node.
2. Expand the Realms node.
3. Click the name of the realm you are configuring (for example, TestRealm).
4. Click the Providers node.
5. Click Role Mappers.

The Role Mappers page appears. This page displays the name of the default Role Mapping provider for the realm that is being configured.

6. Click the Configure a new Default Role Mapper... link.

The General page appears.

7. Define values for the attributes on the General page.
8. Click Apply to save your changes.
9. Reboot WebLogic Server.

Configuring a WebLogic Identity Assertion Provider

Note: The WebLogic Server Administration Console refers to the WebLogic Identity Assertion provider as the Default Identity Asserter.

To define attributes for the WebLogic Identity Assertion provider:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, TestRealm).
3. Expand the Providers-->Authentication Providers nodes.

The Authenticators table displays the name of the default Authentication and Identity Assertion providers.

4. Choose the Configure a new Default Identity Asserter... link from the Authenticators tab.

The General tab appears.

5. Configure a user name mapper. For more information, see [“Configuring a User Name Mapper” on page 428-36](#) and [“Configuring a Custom User Name Mapper” on page 428-38](#).
6. In the Trusted Client Principals attribute define the list of client principals that can use CSIV2 identity assertion. You can use an asterisk (*) to specify all client principals.
7. Define the active token type for the WebLogic Identity Assertion provider. The list of token types supported by the Identity Assertion is displayed in the Available list box. To define the active token type for the Identity Assertion provider:
 - a. In the Available list box, click the desired token type to highlight it.
 - b. Click the right arrow to move token type to the Chosen list box.
8. Click Apply to save your changes.
9. Click the Details tab.
10. Verify the setting of the Base64 Decoding Required attribute.

If the authentication type in a Web application is set to `CLIENT-CERT`, the Web Application Container in WebLogic Server performs identity assertion on values from request headers and cookies. If the header name or cookie name matches the active token type for the configured Identity Assertion provider, the value is passed to the provider.

The Base64 Decoding Required attribute determines whether the request header value or cookie value must be Base64 Decoded before sending it to the Identity Assertion provider. The setting is enabled by default for purposes of backward compatibility, however, most Identity Assertion providers will disable this attribute.

11. Click Apply.
12. Optionally, configure additional Authentication and/or Identity Assertion providers.
13. Reboot WebLogic Server.

Configuring the WebLogic Adjudication Provider

Note: The WebLogic Server Administration Console refers to the WebLogic Adjudication provider as the Default Adjudicator.

To configure the WebLogic Adjudication provider:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, TestRealm.)
3. Expand the Providers node.
4. Click Adjudicators.
5. Click the Configure a new Default Adjudicator... link.
6. Optionally, on the Detail page, set the Require Unanimous Permit attribute.
7. Click Apply to save your changes.
8. Reboot WebLogic Server.

Configuring a WebLogic Auditing Provider

Warning: Using an Auditing provider affects the performance of WebLogic Server even if only a few events are logged.

If you are creating a new security realm, configuring an Auditing provider is an optional step. The WebLogic Server Administration Console refers to the WebLogic Auditing provider as the Default Auditor.

To configure the WebLogic Auditing provider:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, TestRealm).
3. Expand the Providers node.
4. Click Auditors.
5. Click the Configure a new Default Auditor... link.

The General page appears.

6. Click the details
7. Choose the auditing severity level appropriate for your WebLogic Server deployment by setting the Severity attribute.
8. Click Create to save your changes.
9. Reboot WebLogic Server.

Configuring a Custom Security Provider

To configure a Custom security provider:

1. Write a Custom security provider. For more information, see [Developing Security Providers for WebLogic Server](#).
2. Put the MBean JAR file for the provider in the `WL_HOME\lib\mbeantypes` directory.
3. Start the WebLogic Server Administration Console.
4. Expand the Security-->Realms nodes.
5. Click on the name of the realm you are configuring (for example, TestRealm.)
6. Expand the Providers node.
7. Expand the node for the type of provider you are configuring. For example, expand the Authentication Providers node to configure a Custom Authentication provider.

The table page for the provider appears.

8. Click the Configure a new Custom *Security_Provider_Type...* link
where *Security_Provider_Type* is the name of your custom security provider. This name is read from the `DisplayName` attribute in the *MBeanType* tag of the MBean Definition File (MDF).

9. The General page appears.

The Name attribute displays the name of your Custom Security provider.

10. If desired, adjust the values for the attributes for the Custom Security provider.
11. Click Apply to save your changes.
12. Reboot WebLogic Server.

Deleting a Security Provider

To delete a security provider:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm in which the provider you want to delete is configured (for example, TestRealm).
3. Expand the Providers node.
4. Click the type of provider you want to delete (for example, TestRealm-->Authorizers).
5. The table page for the provider appears (for example, the Authorizers table). The table page for the provider displays the names of all the available providers.
6. To delete a provider, click the trash can icon in the corresponding row of the provider table.
7. Reboot WebLogic Server.

Note: Deleting and modifying configured security providers by using the WebLogic Server Administration Console may require manual clean up of the databases.

Configuring a User Name Mapper

When using 2-way SSL, WebLogic Server verifies the digital certificate of the Web browser or Java client when establishing an SSL connection. However, the digital certificate does not identify the Web browser or Java client as a user in the WebLogic Server security realm. If the Web browser or Java client requests a WebLogic Server resource protected by a security policy, WebLogic Server requires the Web browser or

Java client to have an identity. The WebLogic Identity Assertion provider allows you to enable a user name mapper that maps the digital certificate of a Web browser or Java client to a user in a WebLogic Server security realm.

The user name mapper is an implementation the `weblogic.security.providers.authentication.UserNameMapper` interface. By default, WebLogic Server provides a default implementation of the `weblogic.security.providers.authentication.UserNameMapper` interface. You can also write your own implementation

The WebLogic Identity Assertion provider calls the user name mapper for the following types of identity assertion token types:

- X.509 digital certificates passed via the SSL handshake
- X.509 digital certificates passed via CSIV2
- X.501 distinguished names passed via CSIV2

The default user name mapper uses the attributes from the subject DN of the digital certificate or the distinguished name to map to the appropriate user in the WebLogic Server security realm. For example, the user name mapper can be configured to map a user from the Email attribute of the subject DN (`smith@bea.com`) to a user in the WebLogic Server security realm (`smith`).

To use the default user name mapper:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, TestRealm).
3. Expand the Providers-->Authentication Providers nodes.
4. Choose the Default Identity Assertion provider.
5. Click the Details tab.
6. Check the Use the Default User Name Mapper attribute to enable the user name mapper.
7. Specify the following attributes:
 - Default User Name Mapper Attribute Type—The attribute of the subject distinguished name (DN) in a digital certificate used to create a username. Valid values are: C, CN, E, L, O, and OU.

- Default User Name Mapper Attribute Delimiter—The attribute that ends the username. The user name mapper uses everything to the left of the attribute to create a username.
8. Click Apply.
 9. Reboot WebLogic Server.

Configuring a Custom User Name Mapper

To install a custom user name mapper:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, TestRealm).
3. Expand the Providers-->Authentication Providers nodes.
4. Choose the Default Identity Assertion provider.
5. Click the General tab.
6. Enter the name of the implementation of the `weblogic.security.providers.authentication.UserNameMapper` interface in the User Name Mapper Class Name attribute.
7. Click Apply.
8. Reboot WebLogic Server.

Importing and Exporting Security Data from Security Realms

When creating new security realms, security data (authentication, authorization, credential map, and role data) from one security realm can be exported into a file and then imported into another security realm. This feature allows you to develop and test new security realms without recreating all the security data (for example, when moving a development security realm to production). Only information from the WebLogic security providers can be exported and imported. Two options are available:

- Export all security data from a security realm.
- Export specific data (for example, user and groups or roles) from a specific provider. For information, “Importing and Exporting Security Data from Security Providers” on page 428-40.

Note: You can only export and import security data between security realms in the same WebLogic Server release.

To export and import security data:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, TestRealm).
3. Click the Migration-->Export tab.
4. Specify the directory and filename in which to export the security data in the Export Directory on Server attribute.

Note: You can specify a directory and file location on another server.

5. Click Export.
6. Expand the Realms node.
7. Click the name of the security realm in which the security data is to be imported.
8. Click the Migration-->Import tab.
9. Specify the directory location and file name of the file that contains the exported security data in the Import Directory on Server attribute.
10. Click Import.

To verify the security data was imported correctly:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm into which the security data was imported.
3. Click Users.
4. Users from the security realm from which you exported the security data should appear in the Users table.

Importing and Exporting Security Data from Security Providers

Provider-specific security data can also be exported and imported between providers in different security realms. Each provider displays the supported formats (DefaultAtn, DefaultAtz, DefaultCreds, or DefaultRoles). The constraints define the data types (users, groups, roles, and credmaps). The constraints are only displayed for the WebLogic Authentication provider because you have the option of exporting or importing users and groups, just users, just groups, specific users, or specific groups.

To export and import security data from a security provider:

1. Expand the Security-->Realms nodes.
2. Click the name of the realm you are configuring (for example, TestRealm).
3. Click the type of provider from which you want to export security data (for example, Authentication Providers).
4. Click the security provider from which you want to export security data.
5. Click the Migration-->Export tab.
6. Specify the directory and filename in which to export the security data in the Export Directory attribute.
7. Optionally, define a specific set of security data to be exported in the Export Constraints box.
8. Click Export.
9. Expand the Realms node.
10. Click the name of the security realm in which the security data is to be imported.
11. Expand the Providers node.
12. Click the security provider in which the security data is to be imported.
13. Click the Migration-->Import tab.
14. Specify the directory location and file name of the file that contains the exported security data in the Import Directory on Server attribute or use the Browse button to locate the exported file on your computer.

15. Click Import.

Changing the Default Security Realm

By default, WebLogic Server sets the myrealm as the default security realm.

1. Configure a new security realm. For more information, see “Configuring a New Security Realm” on page 428-18.
2. Reboot WebLogic Server.
3. Expand the Domain node (for example, Examples).
4. Click the View Domain-Wide Security Settings link on the General page.
5. Select the Security Configuration-->General tab.

The pull-down menu on the Default Realm attribute displays the security realms configured in the WebLogic domain.

Note: If you create a new security realm but do not configure the required security providers, the realm will not be available from the pull-down menu.

6. Select the security realm you want to set as the default security realm.
7. Click Apply.
8. Reboot WebLogic Server. If you not reboot WebLogic Server, the new realm is not set as the default security realm.

To verify you set the default security realm correctly:

1. Expand the Security node.
2. Expand the Realm node.

The [Realms](#) table appears. All the realms available in the domain are listed. The default security realm has the Default Realm attribute set to `true`.

Deleting A Security Realm

1. Expand the Security node.
2. Expand the Realm node.

The Realms table appears. All the realms available the domain are listed in a table.
3. To delete a security realm, click the trash can icon in the corresponding row of the Realms table.
4. A Delete confirmation window appears.
5. Click Yes in response to the following prompt:

Are you sure you want to permanently delete *OldRealm* from the domain configuration?

A confirmation message appears when the security realm is deleted.

Creating Credential Maps

Resource adapters defined by the J2EE Connector Architecture can acquire the credentials necessary to authenticate users defined in an Enterprise Information System (EIS) when they request access to a protected WebLogic resource. The container in WebLogic Server that hosts resource adapters can retrieve the appropriate set of credentials for the WebLogic resource using a credential map. A credential map creates an association between a user in WebLogic Server security realm and an identity (a username and password combination) used to authenticate that user in an EIS such as an Oracle database, a SQL server, or a SAP application.

Credential maps can be created through the WebLogic Server Administration Console. If you are using the WebLogic Credential Mapping provider, the credential maps are stored in the embedded LDAP server.

To create a credential map:

1. Verify the Ignore Security Data in Deployment Descriptors attribute is enabled on the default (active) security realm. Otherwise, you risk overwriting credential maps with old information in `weblogic-ra.xml` deployment descriptor files.

2. Define a user or group for the EIS user. For more information, see [Users and Groups](#) in *Securing WebLogic Resources*.
3. Deploy a resource adapter. For more information, see [Programming WebLogic J2EE Connectors](#).
4. In the left pane of the WebLogic Server Administration Console, expand Deployments, then Connector Modules.
5. Right-click the name of the Connector for which you want to create a credential map, and choose Define Credential Mappings... to display the Credential Mappings page.

If available, a table of currently defined credential maps appears in the right pane.
6. Click the Configure a New Credential Mapping... link.

If multiple WebLogic Credential Mapping providers are configured in the security realm, select which WebLogic Credential Mapping provider's database should store information for the new credential map.
7. Enter the WebLogic Server user or group name you defined for the EIS user in step 2 in the WLS User field.
8. Click Apply to save your changes.

Configuring Keystores and SSL

By default, WebLogic Server is configured with two keystores:

- `DemoIdentity.jks`—Contains a demonstration private key for WebLogic Server. This keystore establishes an identity for WebLogic Server.
- `DemoTrust.jks`—Contains a list of certificate authorities trusted by WebLogic Server. This keystore establishes trust for WebLogic Server.

These keystores are located in the `BEA_HOME\weblogic710\server\lib` directory. For testing and development purposes, the keystore configuration is complete. Use the steps in this section to configure identity and trust keystores for production use.

Before you perform the steps in this section, you need to:

1. Obtain private keys and digital certificates from a reputable certificate authority such as Verisign, Inc. or Entrust.net.
2. Create identity and trust keystores.
3. Load the private keys and trusted CAs into the keystores.

For a complete description of these steps, see [Managing WebLogic Security](#).

To set attributes for the identity and trust keystores:

1. Expand the Servers node.
2. Click the name of the server for which you want to configure keystores (for example, exampleserver).
3. Click the Configuration-->Keystores and SSL tab.

The information about the demonstration keystores is displayed in the Keystore Configuration.

4. Click the Change... link in the Keystore Configuration to configure new keystores.
5. Choose the type of keystore configuration being used. The following options are available:
 - Demo Identity and Demo Trust—The demonstration identity and trust keystores located in the `BEA_HOME\server\lib` directory and configured by default.
 - Custom Identity and Java Standard Trust—A keystore you create and the trusted CAs defined in the `cacerts` file in the `JAVA_HOME\jre\lib\security\cacerts` directory.
 - Custom Identity and Custom Trust—Identity and trust keystores you create.
 - Custom Identity and Command-Line Trust—An identity keystore you create and command-line arguments that specify the location of the trust keystore.
6. Click Continue.
7. Define attributes for the Identity keystore.
 - Custom Identity Keystore File—The fully qualified path to the identity keystore.

- Custom Identity Keystore Type—The type of the keystore. Generally, this attribute is `jks`.
- Custom Identity Keystore Passphrase—The password defined when creating the keystore. This attribute is optional or required depending on the type of keystore. All keystores require the passphrase in order to write to the keystore. However, some keystores do not require the passphrase to read from the keystore. WebLogic Server only reads from the keystore so whether or not you define this property depends on the requirements of the keystore.

Note: The passphrase for the Demo Identity keystore is `DemoIdentityPassPhrase`.

8. Define properties for the trust keystore.

If you choose Java Standard Trust, specify the password defined when creating the keystore. Confirm the password.

If you choose Custom Trust, define the following attributes:

- Custom Trust Keystore File—The fully qualified path to the trust keystore.
- Custom Trust Keystore Type—The type of the keystore. Generally, this attribute is `jks`.
- Custom Trust Keystore Passphrase—The password defined when creating the keystore. This attribute is optional or required depending on the type of keystore. All keystores require the passphrase in order to write to the keystore. However, some keystores do not require the passphrase to read from the keystore. WebLogic Server only reads from the keystore so whether or not you define this property depends on the requirements of the keystore.

9. Click Continue.

10. If necessary, update the definitions for the SSL attributes. The attributes are:

- Alias—The alias you used when loading the private key for WebLogic Server into the identity keystore.
- Passphrase—The password used to retrieve the private key for WebLogic Server from the identity keystore.
- Confirm—Re-enter the password.

11. Click Continue.

12. Click Finish.

13. Reboot WebLogic Server.

Configuring Two-Way SSL

By default, WebLogic Server is configured to use one-way SSL (the server passes its identity to the client). For a more secure SSL connection, use two-way SSL. In a two-way SSL connection, the client verifies the identity and trust of the server and then passes its identity and trust to the server. The server then validates the identity and trust of the client before completing the SSL connection. The server determines whether or not two-way SSL is used.

To enable two-way SSL:

1. Expand the Servers node.
2. Click the name of the server for which you want to configure keystores (for example, `exampleserver`).
3. Click the Configuration-->Keystores and SSL tab.
4. Click the Show link under Advanced Options.
5. Go to the Server attributes section of the window.
6. Set the Two Way Client Cert Behavior attribute. The following options are available:
 - Client Certs Not Requested—The default (meaning one-way SSL).
 - Client Certs Requested But Not Enforced—Requires a client to present a certificate. If a certificate is not presented, the SSL connection continues.
 - Client Certs Requested And Enforced—Requires a client to present a certificate. If a certificate is not presented, the SSL connection is terminated.
7. Click Apply.
8. Reboot WebLogic Server.

Enabling Trust Between WebLogic Domains

A trust relationship is established when principals in a Subject from one WebLogic Server domain (referred to as a domain) are accepted as principals in the local domain. If you want two domains to interoperate, perform the following procedure in both domains.

To establish a trust relationship between WebLogic Server domains:

1. Expand the Domains node (for example, Examples).
2. Click the View Domain-Wide Security Settings link on the Domain-->General page.
3. Select the Security Configuration-->Advanced tab.
4. Uncheck the Enable Generated Credential attribute.
5. Enter a password for the domain in the Credential text field. Choose the password carefully. BEA Systems recommends using a combination of upper and lower case letters and numbers.
6. Confirm the password by entering it in the Confirm Credential text field.
7. Click Apply.
8. Reboot WebLogic Server.

If you want a WebLogic Server 6.x domain to interoperate with a WebLogic Server 7.0 domain, change the Credential attribute in the WebLogic Server 7.0 domain to the password of the `system` user in the WebLogic Server 6.x domain.

Configuring Connection Filtering

To configure a connection filter:

1. Expand the Domains node.
2. Click the View Domain-Wide Security Settings link on the Domain-->General tab.
3. Select the Security Configuration-->Filter tab.

4. Click the Connection Logger Enabled attribute to enable the logging of accepted messages.
5. Enter the class that implements the network connection filter in the Connection Filter attribute. This class must also be specified in the CLASSPATH for WebLogic Server.
6. Enter the syntax for the connection filter rules. For more information about connection filter rules, see [Using Network Connection Filters](#) in *Programming WebLogic Security*.
7. Click Apply.
8. Reboot WebLogic Server.

Attributes and Console Screen Reference for Security

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

- “Domain-->Security-->General” on page 360-1
- “Domain-->Security-->Embedded LDAP” on page 362-1
- “Domain-->Security-->Filter” on page 361-1
- “Domain-->Security-->Advanced” on page 363-1
- “Servers-->Configuration-->Keystores and SSL” on page 436-1
- “Credential Maps” on page 426-1
- “Security-->Credential Map” on page 427-1
- “Security-->URL Resource-->General” on page 425-1
- “Active Directory Authentication Provider-->Active Directory” on page 324-1
- “Active Directory Authentication Provider-->General” on page 325-1
- “Active Directory Authentication Provider-->Groups” on page 326-1
- “Active Directory Authentication Provider-->Membership” on page 327-1
- “Active Directory Authentication Provider-->Users” on page 328-1
- “Active Directory Authentication Provider-->Details” on page 329-1
- “Adjudication Provider” on page 330-1
- “Auditing Provider” on page 331-1
- “Authentication Providers” on page 332-1
- “Authorization Provider” on page 333-1
- “Credential Mapping Provider” on page 335-1

“Keystore Provider” on page 379-1

“Role Mapping Provider” on page 418-1

“WebLogic Adjudication Provider-->General” on page 336-1

“WebLogic Adjudication Provider-->Details” on page 337-1

“WebLogic Auditing Provider-->General” on page 338-1

“WebLogic Auditing Provider-->Details” on page 339-1

“WebLogic Authentication Provider-->General” on page 341-1

“WebLogic Authentication Provider-->Details” on page 340-1

“WebLogic Authentication Provider-->Export” on page 342-1

“WebLogic Authentication Provider-->Import” on page 343-1

“Weblogic Authorization Provider-->General” on page 345-1

“WebLogic Authorization Provider-->Details” on page 344-1

“WebLogic Authorization Provider-->Export” on page 346-1

“WebLogic Authorization Provider-->Import” on page 347-1

“WebLogic Credential Mapping-->General” on page 349-1

“WebLogic Credential Mapping Provider-->Details” on page 348-1

“WebLogic Credential Mapping Provider-->Export” on page 350-1

“WebLogic Credential Mapping Provider-->Import” on page 351-1

“Weblogic Identity Assertion Provider-->General” on page 352-1

“WebLogic Identity Assertion Provider-->Details” on page 353-1

“WebLogic Keystore Provider-->General” on page 355-1

“Weblogic Keystore Provider-->Details” on page 354-1

“WebLogic Role Mapping Provider-->General” on page 357-1

“WebLogic Role Mapping Provider-->Details” on page 356-1

“WebLogic Role Mapping-->Export” on page 358-1

“WebLogic Role Mapping Provider-->Import” on page 359-1

“EJB Policies and Roles” on page 368-1

“Groups” on page 369-1

“Groups-->General” on page 370-1

“Group-->Details” on page 371-1

“Group-->Membership” on page 372-1

“iPlanet Authentication Provider-->General” on page 374-1

“iPlanet Authentication Provider-->Details” on page 373-1

“iPlanet Authentication Provider-->Groups” on page 375-1

“iPlanet Authentication Provider-->iPlanet LDAP” on page 376-1

“iPlanet Authentication Provider-->Membership” on page 377-1

“iPlanet Authentication Provider-->Users” on page 378-1

“Novell Authentication Provider-->General” on page 380-1

“Novell Authentication Provider-->Details” on page 381-1

“Novell Authentication Provider-->Groups” on page 382-1

“Novell Authentication Provider-->Membership” on page 383-1

“Novell Authentication Provider-->Novell LDAP” on page 384-1

“Novell Authentication Provider-->Users” on page 385-1

“Open LDAP Authentication Provider-->General” on page 386-1

“Open LDAP Authentication Provider-->Details” on page 387-1

“Open LDAP Authentication Provider-->Groups” on page 388-1

“Open LDAP Authentication Provider-->Membership” on page 389-1

“Open LDAP Authentication Provider-->Open LDAP” on page 390-1

“Open LDAP Authentication Provider-->Users” on page 391-1

“Define Policy” on page 392-1

“Security Realms” on page 412-1

“Security Realm-->User Management” on page 393-1

“Security Realm-->General” on page 394-1

“Security Realm-->Migration-->Export” on page 395-1

“Security Realm-->Migration-->Import” on page 396-1

“Security Realm-->UserLockout” on page 405-1

“Security Realm-->Providers” on page 419-1

“Security Realm-->Testing” on page 404-1

“Adjudication Provider” on page 397-1

“Auditing Provider” on page 398-1

“Authentication Provider” on page 399-1

“Authorization Provider” on page 400-1

“Credential Mapping Provider” on page 401-1

“Keystores” on page 402-1

“Role Mapping Provider” on page 403-1

“Realm Adapter Authentication Provider-->General” on page 409-1

“Security Realms” on page 412-1

“Global Roles” on page 413-1

“Security Role-->General” on page 415-1

“Security Role-->General” on page 417-1

“Security Role-->Conditions” on page 414-1

“Security Role-->Details” on page 416-1

“User-->General” on page 422-1

“User-->Details” on page 421-1

“User-->Groups” on page 423-1

“Users” on page 420-1

“Change Password” on page 334-1

“Unlock User Accounts” on page 320-1

“War Policies and Roles” on page 424-1



Server --> Notes

Attributes

Overview

Use this page to describe the configuration or function of this server.

Enter freeform text in the Notes field and click Apply to apply your changes. The note is saved as part of the server's configuration and is persisted in the domain's `config.xml` file.

Attributes

Table 430-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code> <i>Attribute:</i> <code>Notes</code>	<i>Dynamic:</i> yes



Configure SSL-->Determine Compatibility Level

[Tasks](#) [Related Topics](#)

Overview

Use this page to configure the location of identity and trust for WebLogic Server for the purposes of SSL configuration.

For purposes of backward compatibility, WebLogic Server allows you to store private keys and trusted certificates authorities in files or in the WebLogic Keystore provider. If you use either of these mechanisms for identity and trust, choose the Files or Keystore providers option from this page.

Click Continue to specify where your identity and trust is located.

Note: When using the WebLogic Keystore provider digital certificates are stored in files.

Tasks

“Configuring Keystores and SSL” on page 1-43

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Server --> Configuration --> Cluster

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

A WebLogic Server cluster is a group of servers that work together to provide a more scalable, more reliable application platform than a single server.

Use this page to specify a server's primary and secondary replication group, cluster weight, and the interface address used to handle multicast traffic.

Tasks

“Specifying a Server's Cluster Replication Group” on page 41-5

“Specifying a Server's Cluster Weight” on page 41-6

“Specifying the Address of a Server's NIC Card for Cluster Communication” on page 41-6

Related Topics

“Setting Up WebLogic Clusters” in *Using WebLogic Clusters*.

Attributes

Table 432-1

Attribute Label	Description	Value Constraints
Replication Group	<p>Defines preferred clustered instances considered for hosting replicas of the primary HTTP session states created on the server.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> ReplicationGroup</p>	
Preferred Secondary Group	<p>Defines secondary clustered instances considered for hosting replicas of the primary HTTP session states created on the server.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> PreferredSecondaryGroup</p>	
Cluster Weight	<p>Defines a value used specify the proportion of the load the server will bear relative to other servers in a cluster.</p> <p>If all servers have the default weight (100) or the same weight, each bears an equal proportion of the load. If one server has weight 50 and all other servers have weight 100, the 50-weight server will bear half as much load as any other server.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> ClusterWeight</p>	<p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 100</p> <p><i>Default:</i> 100</p> <p><i>Configurable:</i> yes</p>

Table 432-1

Attribute Label	Description	Value Constraints
Interface Address	The interface address used to specify the NIC that handles cluster multicast traffic. <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> InterfaceAddress	<i>Configurable:</i> yes



Server --> Configuration --> Deployment

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use the Server-->Configuration-->Deployments tab to:

- Specify whether deployment should copy the applications's files to the managed server's staging area (`StagingMode`)
- Set the location to which files should be staged (`StagingDirectoryName`)
- Specify the target directory path on the Admin Server for uploaded files (`upload directory`)

Tasks

[“Setting the Server Staging Mode” on page 62-5](#)

[“Deploying New Applications and Modules” on page 62-6](#)

Related Topics

[“Starting and Stopping Servers” on page 497-1](#)

Attributes

Table 433-1

Attribute Label	Description	Value Constraints
Staging Mode	<p>During application preparation, the application's files are copied from the source on the admin server to the managed server's staging area. If this attribute is nostage or external_stage, the copy will not occur. This is useful when the staging area is a shared directory, already containing the application files, or if this is a single server domain. The administrator must ensure that the managed server's staging directory is set appropriately. Deployment errors will result if the application is not available during the preparation or activation of the application. This attribute can be overridden with the ApplicationMBean StagingMode attribute.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> StagingMode</p>	<p><i>Default:</i> null</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ stage■ nostage■ external_stage
Staging Directory Name	<p>Identifies the directory path on the managed server where all staged (prepared) applications are placed. This attribute is not dynamic. If an absolute directory name is not specified, the path is relative to rootdirectory/ Once configured, this attribute may not be changed. Remove all applications from the server prior to changing this attribute. Changes to this attribute require a server restart to take effect. The default staging directory is "stage", relative to the server root..</p>	

Table 433-1

Attribute Label	Description	Value Constraints
Upload Directory Name	Identifies the directory path on the AdminServer where all uploaded applications are placed. If an absolute directory name is not specified, the path is relative to rootdirectory/. The default staging directory is "stage", relative to the server root.. On the ManagedServer this returns null, and is not configurable <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> UploadDirectoryName	<i>Dynamic:</i> yes



Server --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The ~~Server—Configuration—General~~ tab defines general configuration attributes for an instance of WebLogic Server.

Tasks

“Configuring a Machine” on page 268-1

“Configuring a Cluster” on page 41-1

“Configuring the Listen Address” on page 495-11

“Configuring the Listen Ports” on page 495-13

“Starting Managed Servers from the Administration Console” on page 497-5

“Shutting Down Instances of WebLogic Server” on page 497-28

“Adding and Removing Servers in an Existing Domain” on page 495-2

“Deleting a Managed Server” on page 495-5

"Creating Execute Queues" in *WebLogic Server Performance and Tuning*

Related Topics

[Using WebLogic Server Clusters](#)

[Administration Port Configuration and Startup](#)

Attributes

Table 434-1

Attribute Label	Description	Value Constraints
Name	<p>An alphanumeric name for this server instance. This field will not accept spaces.</p> <p>The name must be unique for all configuration objects in the domain. Within a domain, each server, machine, cluster, JDBC connection pool, virtual host, and any other resource type must be named uniquely and must not use the same name as the domain.</p> <p>The server name is not used as part of the URL for applications that are deployed on the server. It is for your identification purposes only. The server name displays in the Administration Console, and if you use WebLogic Server command-line utilities or APIs, you use this name to identify the server.</p> <p>After you have created a server, you cannot change its name. Instead, clone the server and provide a new name for the clone. For more information, refer to “Cloning a Server” on page 495-4.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> Name</p>	

Table 434-1

Attribute Label	Description	Value Constraints
Machine	<p>The WebLogic Server host computer (machine) on which this server is meant to run.</p> <p>If you want to use a Node Manager to start this server, you must assign the server to a machine and you must configure the machine for the Node Manager. Node Manager uses this information to communicate with the server on its host machine.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>Machine</code></p>	<i>Configurable:</i> yes
Cluster	<p>The cluster to which this server belongs. If set, the server will listen for cluster multicast events.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>Cluster</code></p>	<i>Configurable:</i> yes
Listen Address	<p>The address from which this server listens for requests.</p> <p>Servers can be reached through the following URL:</p> <p><i>protocol://listen-address:listen-port</i></p> <p>By default, a server's listen address attribute is undefined, which enables you to reach the server through an IP address of the computer that hosts the server, a DNS name that resolves to the host, or the <code>localhost</code> string. For more information, refer to "Configuring the Listen Address" on page 495-11.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>ListenAddress</code></p>	

Table 434-1

Attribute Label	Description	Value Constraints
Listen Port Enabled	<p>Determines whether the server can be reached through the default plain-text (non-SSL) listen port.</p> <p>If you disable this listen port, you must enable the default SSL listen port.</p> <p>You can define additional listen ports for this server by configuring network channels.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> ListenPortEnabled</p>	<p><i>Default:</i> true</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false
Listen Port	<p>The default plain-text (non-SSL) listen port for this server.</p> <p>For more information, refer to “Configuring the Listen Ports” on page 495-13.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> ListenPort</p>	<p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 65534</p> <p><i>Default:</i> 7001</p> <p><i>Configurable:</i> yes</p>
SSL Listen Port Enabled	<p>Determines whether the server can be reached through the default SSL listen port.</p> <p>If you disable this listen port, you must enable the default plain-text (non-SSL) listen port.</p> <p>You can define additional listen ports for this server by configuring network channels.</p> <p><i>MBean:</i> weblogic.management.configuration.SSLMBean</p> <p><i>Attribute:</i> Enabled</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false

Table 434-1

Attribute Label	Description	Value Constraints
SSL Listen Port	<p>The default SSL listen port for this server.</p> <p>For more information, refer to “Configuring the Listen Ports” on page 495-13.</p> <p><i>MBean:</i> weblogic.management.configuration.SSLMBean</p> <p><i>Attribute:</i> ListenPort</p>	<p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 65534</p> <p><i>Default:</i> 7002</p> <p><i>Configurable:</i> yes</p>
Client Cert Proxy Enabled	<p>When set to true for a server instance, this attribute specifies that certs from clients of web applications hosted on the server instance are provided in the special WL-Proxy-Client-Cert header sent by a proxy plug-in or HttpClusterServlet.</p> <p>(ClientCertProxyEnabled can be defined at the cluster level, at the server level, and at the web application level, in web.xml.)</p> <p>This setting is useful if user authentication is performed on the proxy server—setting clientCertProxy to true causes the plug-in to pass on the certs to the cluster in the WL-Proxy-Client-Cert header.</p> <p>The header can be used by any client with direct access to WebLogic Server. WebLogic Server takes the certificate information from that header, trusting that it came from a secure source (the plug-in) and uses that information to authenticate the user.</p> <p>If you set clientCertProxy to true, use a connection filter to ensure that WebLogic Server accepts connections only from the machine on which the plug-in is running. See "Using Network Connection Filters" in <i>Programming WebLogic Security</i>.</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ true ■ false

Table 434-1

Attribute Label	Description	Value Constraints
Java Compiler	The Java Compiler for all applications that need to compile Java code. <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> JavaCompiler	<i>Default:</i> "javac" <i>Dynamic:</i> yes

Advanced Attributes

Table 434-2

Attribute Label	Description	Value Constraints
WebLogic Plug-In Enabled	<p>Set this attribute to true if the server instance will receive requests from a proxy plug-in or <code>HttpClusterServlet</code>.</p> <p>When <code>WeblogicPluginEnabled</code> is true, a call to <code>getRemoteAddr</code> will return the address of the browser client from the proprietary <code>WL-Proxy-Client-IP</code> header, instead of the web server.</p> <p>If the server instance is a member of a cluster that will receive proxied requests, set <code>WeblogicPluginEnabled</code> for the cluster, on the Cluster</p> <p>-->Configuration-->General tab.</p> <p><code>WeblogicPluginEnabled</code> can be configured in <code>ClusterMBean</code> or <code>ServerMBean</code>. If specified in both <code>ClusterMBean</code> and <code>ServerMBean</code>, the value in <code>ClusterMBean</code> value takes precedence.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>WeblogicPluginEnabled</code></p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false

Table 434-2

Attribute Label	Description	Value Constraints
Startup Mode	<p>The state in which this server should be started.</p> <p>In the <code>RUNNING</code> state, a server offers its services to clients and can operate as a full member of a cluster. In the <code>STANDBY</code> state, a server can accept administration commands and participate in cluster communication, but is not accessible for requests that come from external clients. (If you select <code>STANDBY</code>, you must also enable the domain-wide administration port under this domain's Configuration > General tab.)</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>StartupMode</code></p>	<p><i>Valid values:</i></p> <ul style="list-style-type: none">■ <code>RUNNING</code>■ <code>STANDBY</code>
Prepend to classpath	<p>The options to prepend to the Java compiler classpath for when we need to compile Java code.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>JavaCompilerPreClassPath</code></p>	<p><i>Default:</i> null</p>
Append to classpath	<p>The options to append to the Java compiler classpath for when we need to compile Java code.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>JavaCompilerPostClassPath</code></p>	<p><i>Default:</i> null</p>

Table 434-2

Attribute Label	Description	Value Constraints
Extra RMI Compiler Options	<p>The extra options passed to <code>rmic</code> during server-side generation.</p> <p>If extra <code>rmic</code> options are specified at the EJB component level—on <code>EJBComponentMBean</code>—that value overrides this server-level value. See Configuring Compiler Options.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>ExtraRmicOptions</code></p>	<i>Default:</i> null
Extra EJB Compiler Options	<p>The extra options passed to <code>ejbc</code> during server-side generation.</p> <p>Note: If extra <code>ejbc</code> options are specified at the EJB component level—on <code>EJBComponentMBean</code>—that value overrides this server-level value. See Configuring Compiler Options.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>ExtraEjbcOptions</code></p>	<i>Default:</i> null

Table 434-2

Attribute Label	Description	Value Constraints
External Listen Address	<p>The external DNS name for the current server, which will be sent with HTTP session cookies and also with the dynamic server lists to HTTP proxies.</p> <p>This attribute is required for configurations in which a firewall is performing Network Address Translation, unless clients are accessing WebLogic Server using t3 and the default channel. For instance, define the external DNS name for configurations in which a firewall is performing Network Address Translation, and clients are accessing WebLogic Server using HTTP via a proxy plug-in.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> ExternalDNSName</p>	<p><i>Default:</i> null</p> <p><i>Configurable:</i> yes</p> <p><i>Required:</i> yes</p>
Administration Port Enabled	<p>Indicates whether the administrative channel is enabled for the domain.</p> <p>You enable the administrative channel on the Domain > Configuration > General tab.</p> <p>For more information, refer to “Administration Port Configuration and Startup.”</p>	<p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false

Table 434-2

Attribute Label	Description	Value Constraints
Local Administration Port Override (0: no override)	<p>Overrides the domain-wide administration port and specifies a different listen port on which this server listens for administrative requests. Valid only if the administrative channel is enabled for the domain.</p> <p>By default, if you enable the domain's administrative channel, all servers in the domain use the same listen port to listen for administrative requests.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>AdministrationPort</code></p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 65534</p> <p><i>Default:</i> 0</p>



Server --> Configuration --> Health Monitoring

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

WebLogic Server provides a self-health monitoring capability to improve the reliability and availability of servers in a domain. Selected subsystems within each WebLogic Server instance monitor their health status based on criteria specific to the subsystem.

Use this page to configure the frequency of this server's automated health checks, and the frequency with which the Node Manager application (optional) checks the servers's health state. You can also use this page to specify whether Node Manager automatically stops and restarts the server if the server reaches the "failed" health state.

Tasks

["Configuring a Machine" on page 268-1](#)

["Configure Self-Health Monitoring, Shutdown, and Restart for Managed Servers" on page 495-22](#)

Related Topics

["Overview of Node Manager"](#)

Attributes

Table 435-1

Attribute Label	Description	Value Constraints
Auto Restart	<p>When Auto Restart is enabled, Node Manager will try to restart the Managed Server if it crashes or goes down unexpectedly, for instance, as the result of a machine reboot. Deselect the check box if you do not want Node Manager to automatically restart the Managed Server after a crash.</p> <p>Enables/Disables automatic restart of a crashed server by the Node Manager.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> AutoRestart</p>	<p><i>Default:</i> true</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false <p><i>Configurable:</i> yes</p> <p><i>Dynamic:</i> yes</p>
Auto Kill If Failed	<p>Check this box to enable Node Manager to automatically kill the Managed Server when its health state is failed. By default, this option is disabled.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> AutoKillIfFailed</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false <p><i>Configurable:</i> yes</p> <p><i>Dynamic:</i> yes</p>
Restart Interval	<p>Enter the period of time (in seconds) during which Node Manager should attempt to restart the Managed Server. This attribute is used in conjunction with the Max Restarts within Interval attribute to limit attempts to restart the Managed Server.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> RestartIntervalSeconds</p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> 300</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 3600</p> <p><i>Configurable:</i> yes</p> <p><i>Dynamic:</i> yes</p>

Table 435-1

Attribute Label	Description	Value Constraints
Max Restarts within Interval	Enter the maximum number of times Node Manager can restart the Managed Server within the interval specified by Restart Interval. <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> RestartMax	<i>Minimum:</i> 0 <i>Maximum:</i> 2147483647 <i>Default:</i> 2 <i>Configurable:</i> yes <i>Dynamic:</i> yes
Health Check Interval	Specify the interval (in seconds) between Node Manager health state queries to the Managed Server. <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> HealthCheckIntervalSeconds	<i>Units:</i> seconds <i>Minimum:</i> 1 <i>Maximum:</i> 2147483647 <i>Default:</i> 180 <i>Configurable:</i> yes <i>Dynamic:</i> yes
Health Check Timeout	Enter the number of seconds that Node Manager waits for a response to a health state query. If the timeout is reached, Node Manager assumes the Managed Server has failed. <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> HealthCheckTimeoutSeconds	<i>Units:</i> seconds <i>Minimum:</i> 1 <i>Maximum:</i> 2147483647 <i>Default:</i> 60 <i>Configurable:</i> yes <i>Dynamic:</i> yes
Restart Delay Seconds	Enter the number of seconds that Node Manager should wait before trying to restart a Managed Server. After killing a server process, the system might need several seconds to release the TCP port(s) the server was using. If Node Manager attempts to restart the Managed Server while its ports are still active, the startup attempt fails. <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> RestartDelaySeconds	<i>Units:</i> seconds <i>Minimum:</i> 0 <i>Maximum:</i> 2147483647 <i>Default:</i> 0 <i>Configurable:</i> yes <i>Dynamic:</i> yes



Servers-->Configuration-->Keystores and SSL

[Tasks](#) [Related Topics](#)

Overview

Use this page to configure attributes for the identity (private keys) and trust (trusted certificate authorities) keystores for WebLogic Server and attribute for SSL.

WebLogic Server is configured with a default identity keystore (`DemoIdentity.jks`) and a default trust keystore (`DemoTrust.jks`). In addition, WebLogic Server trusts the certificate authorities in the `cacerts` file the JDK. This keystore configuration is appropriate for testing and development purposes. However, these keystores should not be used in a production environment. Click **Change** in the Keystore Configuration pane to modify the keystore configuration for WebLogic Server.

After you configure identity and trust keystores for WebLogic Server, you need to configure attributes for SSL. These attributes include information about the identity and trust location for a particular server instances. Click **Change** in the SSL pane to specify this information. If you are not using keystores for the purpose of SSL (for example, if the keystores are used for digital signing purposes), you do not need to define these attributes.

For purposes of backward compatibility, WebLogic Server allows you to store private keys and trusted certificates authorities in files or in the WebLogic Keystore provider. If you use either of these mechanisms for identity and trust, choose the **Files or Keystore providers** option.

Note: When using the WebLogic Keystore provider digital certificates are stored in files.

Click **Show** in the Advanced Options pane to set additional attributes in the following cases: you are using the Node Manager; a WebLogic Server instance is acting as a client; the administration port is enabled and you have Managed Servers; application code is using SSL; or an external LDAP server is running over SSL.

Tasks

“Configuring Keystores and SSL” on page 428-43

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Server --> Configuration --> Remote Start

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to configure the startup arguments that Node Manager will use when you starting this Managed Server from the Administration Console.

To start the Managed Server using Node Manager, a Node Manager process must run on the same machine as the Managed Server.

Tasks

[“Configure Startup Arguments for Managed Servers” on page 497-7](#)

[“Configuring a Machine” on page 268-1](#)

[“Starting Managed Servers from the Administration Console” on page 497-5](#)

[“Starting All Managed Servers in a Domain” on page 497-11](#)

[“Shutting Down a Server” on page 497-28](#)

[“Shutting Down All Managed Servers in a Domain” on page 497-29](#)

Related Topics

["Overview of Node Manager"](#)

["Configuring, Starting, and Stopping Node Manager"](#)

[“Starting and Stopping Servers” on page 497-1](#)

Attributes

Table 437-1

Attribute Label	Description	Value Constraints
Java Home	<p>The Java home directory (on the Node Manager machine) to use when starting this server.</p> <p>Specify the parent directory of the JDK's bin directory. For example, c:\bea\jdk141.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerStartMBean</p> <p><i>Attribute:</i> JavaHome</p>	
BEA Home	<p>The BEA home to be used to start this server.</p> <p>Specify the directory on the Node Manager machine under which all BEA products and licenses were installed. For example, c:\bea.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerStartMBean</p> <p><i>Attribute:</i> BeaHome</p>	
Root Directory	<p>The directory this server uses as its root directory. For more information, refer to “A Server's Root Directory.”</p> <p>If you do not specify a Root Directory value, the default Node Manager working directory is used (generally WL_HOME\common\nodemanager).</p> <p><i>MBean:</i> weblogic.management.configuration.ServerStartMBean</p> <p><i>Attribute:</i> RootDirectory</p>	

Table 437-1

Attribute Label	Description	Value Constraints
Class Path	<p>The full class path required to start this Managed Server. Note that all paths refer to paths on the Node Manager machine.</p> <p>At a minimum you will need to specify the following values for the class path option:</p> <pre>WL_HOME/server/lib/weblogic_sp.jar;WL_HOME/server/lib/weblogic.jar</pre> <p>where <i>WL_HOME</i> is the directory in which you installed WebLogic Server on the Node Manager machine.</p> <p>The shell environment determines which character you use to separate path elements. On Windows, you typically use a semicolon (;). In a BASH shell, you typically use a colon (:).</p> <p>For more information about class path requirements, refer to "Setting the Classpath."server.</p> <pre>MBean: weblogic.management.configuration.ServerStartMBean</pre> <p><i>Attribute:</i> ClassPath</p>	

Table 437-1

Attribute Label	Description	Value Constraints
Arguments	<p>The startup arguments to pass to the JVM when starting this server.</p> <p>These are the first arguments appended immediately after java portion of the startup command. For example, you can set Java heap memory or specify any WebLogic Server argument described in “weblogic.Server Command-Line Reference.”</p> <p>Separate arguments with a space.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerStartMBean</p> <p><i>Attribute:</i> Arguments</p>	
Security Policy File	<p>The security policy file to use when starting this server.</p> <p>Enter the full path to the WebLogic security policy file on the Node Manager machine. For example, c:\bea\weblogic810\server\lib\weblogic.policy.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerStartMBean</p> <p><i>Attribute:</i> SecurityPolicyFile</p>	

Table 437-1

Attribute Label	Description	Value Constraints
Username	<p>The name of an existing user who has privileges to start a server.</p> <p>For information on user privileges, refer to "Security Roles."</p> <p>The Administration Console and Domain Configuration Wizard insert a user name in this field when you create a Managed Server. The Administration Console inserts the user name that you supplied when you logged in to the console. The Domain Configuration Wizard inserts the user name that you defined when you created the domain.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerStartMBean</p> <p><i>Attribute:</i> Username</p>	<p><i>Default:</i> ""</p>
Password and Confirm Password	<p>The password of the username used to boot the server.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerStartMBean</p> <p><i>Attribute:</i> Password</p>	<p><i>Default:</i> ""</p> <p><i>Encrypted:</i> yes</p>



Server --> Configuration --> Tuning

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab defines configuration attributes for tuning WebLogic Server performance and functionality.

Tasks

[Allocating Threads to Act as Socket Readers](#)

[Setting Thread Count](#)

[Detecting "Stuck" Threads](#)

[Tuning Execute Queues for Overflow Conditions](#)

[Using Execute Queues to Control Thread Usage](#)

[Tuning Connection Backlog Buffering](#)

Related Topics

[BEA WebLogic Server Performance and Tuning guide](#)

Attributes

Table 438-1

Attribute Label	Description	Value Constraints
Enable Native IO	Whether or not native I/O is enabled for the server. <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> NativeIOEnabled	<i>Default:</i> true <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false <i>Configurable:</i> yes
Socket Readers	The percentage (1-99) of execute threads from the default queue hat may be used as socket readers. <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> ThreadPoolPercentSocketReaders	<i>Minimum:</i> 1 <i>Maximum:</i> 99 <i>Default:</i> 33 <i>Configurable:</i> yes <i>Dynamic:</i> yes
Maximum Open Sockets	The maximum number of open sockets allowed in server at a given point of time. When max threshold is reached, server stops accepting any more new requests until no of sockets drops below threshold. <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> MaxOpenSockCount	<i>Minimum:</i> -1 <i>Maximum:</i> 2147483647 <i>Default:</i> -1 <i>Configurable:</i> yes <i>Dynamic:</i> yes
Accept Backlog	Allowed backlog of new TCP connection requests for both the plaintext and SSL port. Setting the backlog to 0 may prevent accepting any incoming connection on some of the OS. <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> AcceptBacklog	<i>Minimum:</i> 0 <i>Default:</i> 50 <i>Dynamic:</i> yes

Table 438-1

Attribute Label	Description	Value Constraints
Login Timeout	<p>The login timeout for the server's plain-text (non-SSL) port, in milliseconds. This is the maximum amount of time allowed for a new connection to establish. A value of 0 indicates there is no maximum. The value must be equal to or greater than 0.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> LoginTimeoutMillis</p>	<p><i>Units:</i> milliseconds</p> <p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 100000</p> <p><i>Default:</i> 5000</p> <p><i>Dynamic:</i> yes</p>
SSL Login Timeout	<p>The login timeout for this server's default SSL listen port. This is the maximum amount of time allowed for a new connection to establish. A value of 0 indicates there is no maximum.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> SSLoginTimeoutMillis</p>	<p><i>Units:</i> milliseconds</p> <p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 100000</p> <p><i>Default:</i> 5000</p> <p><i>Dynamic:</i> yes</p>
Reverse DNS Allowed	<p>Returns whether or not the kernel is allowed to perform reverse DNS lookups.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> ReverseDNSAllowed</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false <p><i>Configurable:</i> yes</p> <p><i>Dynamic:</i> yes</p>

Advanced Attributes

Table 438-2

Attribute Label	Description	Value Constraints
Low Memory GCThreshold	<p>The threshold level (0-99 percent) at which this server logs low memory conditions and changes the server health state to "Warning." By default, the server logs a low memory warning in the log file and changes the server health state to Warning after the average free memory reaches 5 percent of the initial free memory measured at the server's boot time.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> LowMemoryGCThreshold</p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 99</p> <p><i>Default:</i> 5</p> <p><i>Configurable:</i> yes</p>
Low Memory Granularity Level	<p>The granularity level used in reporting LowMemory information.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> LowMemoryGranularityLevel</p>	<p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 100</p> <p><i>Default:</i> 5</p> <p><i>Configurable:</i> yes</p>
Low Memory Sample Size	<p>The total sample size used for LowMemoryTimeInterval. By default '10' samples are taken at each LowMemoryTimeInterval.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> LowMemorySampleSize</p>	<p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 10</p> <p><i>Configurable:</i> yes</p>

Table 438-2

Attribute Label	Description	Value Constraints
Low Memory Time Interval	<p>The amount of time (in seconds) that defines the interval over which this server determines average free memory values. By default, the server obtains an average free memory value every 3600 seconds.</p> <p>This interval is not used when the server runs under the JRockit VM. Instead, memory samples are collected immediately after the JRockit VM performs scheduled garbage collection. Sampling memory after a garbage collection provides a more accurate average of free memory.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>LowMemoryTimeInterval</code></p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> 300</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 3600</p> <p><i>Configurable:</i> yes</p>
Managed Server Independence Enabled	<p>Indicates whether Managed Server Independence is enabled for this server. With Managed Server Independence enabled, you can start a Managed Server even if the Administration Server is unavailable. In such a case, the Managed Server retrieves its configuration by reading a configuration file and other files directly.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>ManagedServerIndependenceEnabled</code></p>	<p><i>Default:</i> true</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false

Table 438-2

Attribute Label	Description	Value Constraints
MSI File Replication Enabled	<p>Indicates whether the replication of configuration files is enabled for a Managed Server. With file replication enabled, the Administration Server copies its configuration file and <code>SerializedSystemIni.dat</code> into the Managed Server's root directory every 5 minutes. This option does not replicate a boot identity file.</p> <p>Regardless of the name of the configuration file that you used to start the Administration Server, the replicated file is always named <code>msi-config.xml</code>. For example, if you specified <code>-Dweblogic.ConfigFile=MyConfig.xml</code> when you started the Administration Server, if you have enabled file replication, the Administration Server copies <code>MyConfig.xml</code> and names the copy <code>msi-config.xml</code>.</p> <p>Depending on your backup schemes and the frequency with which you update your domain's configuration, this option might not be worth the performance cost of copying potentially large files across a network.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>MSIFileReplicationEnabled</code></p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false

Server --> Control --> Start-Stop

[Tasks](#) [Related Topics](#) [Operations](#) [Attributes](#) [Status](#)

Overview

The Server—Control—Start/Stop tab changes the state of a server.

Some of the operations require the Node Manager and the domain-wide administration port. In addition, some operations cannot be used for an Administration Server.

Tasks

“Starting Managed Servers from the Administration Console” on page 497-5

“Shutting Down a Server” on page 497-28

“Starting a Managed Server in the STANDBY State” on page 497-12

“Resuming a Server” on page 497-24

“Starting All Managed Servers in a Domain” on page 497-11

“Shutting Down All Managed Servers in a Domain” on page 497-29

[“Enabling the Domain-Wide Administration Port” on page 74-1](#)

Related Topics

[Overview of Node Manager](#)

[Overview of the Server Lifecycle](#)

Operations

The following table describes the applicability and requirements of each operation:

Operation	Description	Requirements/Restrictions
Start this server	Starts a Managed Server. By default, a server instance starts in the <code>RUNNING</code> state, but the Startup Mode setting can change the default behavior. The Startup Mode setting is located on the Servers—Configuration—General tab, under Advanced Options.	Requires the Node Manager. Only available for Managed Servers.
Resume this server	Moves a server from the <code>STANDBY</code> state to <code>RUNNING</code> .	Requires the administration port to be enabled.
Graceful shutdown of this server	Gracefully stops a server. New requests are rejected but in-work requests are completed before the server stops.	
Force shutdown of this server	Immediately stops a server. In-work requests are dropped, no new requests are accepted, and the server immediately stops.	

Attributes

Table 439-1

Attribute Label	Description	Value Constraints
Ignore Sessions During Shutdown	<p>Indicates whether a graceful shutdown operation drops all HTTP sessions immediately. If this attribute is set to <code>false</code>, a graceful shutdown operation waits for HTTP sessions to complete or timeout.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>IgnoreSessionsDuringShutdown</code></p>	<p><i>Default:</i> <code>false</code></p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ <code>true</code>■ <code>false</code> <p><i>Configurable:</i> <code>yes</code></p> <p><i>Dynamic:</i> <code>yes</code></p>
Graceful Shutdown Timeout	<p>Number of seconds a graceful shutdown operation waits before forcing a shutdown. A graceful shutdown gives WebLogic Server subsystems time to complete certain application processing currently in progress. If subsystems are unable to complete processing within the number of seconds that this attribute specifies, then the server will force shutdown automatically.</p> <p>See “Graceful Shutdown.”</p> <p>A value of <code>0</code> means that the server will wait indefinitely for graceful shutdown to complete.</p> <p>This attribute applies only to graceful shutdown operations, while the <code>ServerLifecycleTimeoutVal</code> attribute applies only to force shutdowns.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>GracefulShutdownTimeout</code></p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> <code>0</code></p> <p><i>Default:</i> <code>0</code></p> <p><i>Configurable:</i> <code>yes</code></p> <p><i>Dynamic:</i> <code>yes</code></p>

Status

Table 439-2

Table Column	Description
Name	The name of the current server instance.
State	The current lifecycle state of the server. For a description of lifecycle states, refer to Understanding Server State .
Transition Activity	Shows the status of the most recent transition in lifecycle state. For details on the transition, click the link in the Status tab.

Deprecated SSL Identity and Trust Configuration

[Tasks](#) [Related Topics](#)

Overview

When configuring SSL, you need to specify a file name for the private key, server certificate, and trusted CA. If you are using a keystore accessed by the WebLogic Keystore provider, you need to specify the alias used to load the private key into the keystore and the password used to retrieve the private key from the keystore. This page allows you to specify this information.

Tasks

“Configuring Keystores and SSL” on page 1-43

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Connection

The Connection page displays information about the remote agents that are currently communicating with the WebLogic Server. For example, if an Administration Server is communicating with a Managed Server, this page indicates the channel on which the servers are communicating, along with the ID of the Java Virtual Machine.



Execute Thread

This page displays statistics and information for all active threads the server instance's available execute queues. For more information, see:

- ["Tuning the Default Execute Threads"](#) in *WebLogic Server Performance and Tuning*.
- ["Using Execute Queues to Control Thread Usage"](#) in *WebLogic Server Performance and Tuning*.



Server --> Deployments --> Startup/Shutdown

Tasks

Overview

The Server—Deployments—Startup/Shutdown tab lists all of the configurations of startup or shutdown classes that have been configured for this server. Startup and shutdown classes are Java programs that you create and configure for use with WebLogic Server. A server instances invokes them during its startup or shutdown process.

You can use this tab to assign or unassign startup and shutdown classes for this server.

Note: If you want to assign a startup or shutdown class to multiple servers or to a cluster, you can use the Domain—Startup & Shutdown—Targets tab.

Tasks

“Configure a Startup or Shutdown Class” on page 525-2

“Assign a Startup or Shutdown Configuration to Servers or Clusters” on page 525-5

“Clone a Configuration of a Startup or Shutdown Class” on page 525-4



SSL Identity Configuration

[Tasks](#) [Related Topics](#)

Overview

When configuring SSL, you need to specify the alias used to load the private key into the keystore and the password used to retrieve the private key from the keystore. This page allows you to specify this information.

Tasks

“Configuring Keystores and SSL” on page 1-43

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation



Server --> Logging --> Domain

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The Server—Logging—Domain determines which messages a server sends to the domain log (in addition to its own log). The domain log collects messages from all servers within the domain.

By default, a server sends all messages of severity level `ERROR` or higher to the domain log. You can use this tab to assign a log filter, which changes the set of messages that a server sends. You can also use this tab to disable the server from sending any messages to the domain log.

A server does not send HTTP requests, JDBC messages, JTA transaction messages, or messages of the `DEBUG` severity to the domain log file regardless of whether you use a domain log filter.

Tasks

“Specifying the Messages That a Server Forwards to the Domain Log” on page 81-1

“Viewing the Domain Log” on page 253-12

Related Topics

“Overview of WebLogic Server Log Messages and Log Files” on page 253-2

Attributes

Table 445-1

Attribute Label	Description	Value Constraints
Log to Domain log file	Determines whether this server sends messages to the domain log (in addition to keeping its own log). <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> EnabledForDomainLog	<i>Default:</i> true <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false <i>Dynamic:</i> yes
Use log filter	Determines which messages this server sends to the domain log. If you specify none, the server sends all messages of severity <code>ERROR</code> and higher. This list contains all Domain Log Filters that have been defined for the domain. A server can user only one Domain Log Filter. This property is relevant only if <code>Log To Domain File</code> is enabled. <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> DomainLogFilter	<i>Dynamic:</i> yes

Server --> Logging --> HTTP

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The **Server—Logging—HTTP** tab configures HTTP logging for the server. If you enable HTTP logging, the server saves HTTP requests in a separate log file; it does not store HTTP requests in the server log file or the domain log file.

You can also specify the name of the log file that stores HTTP requests and the longevity and format of the data in the file.

Note: If you set up HTTP logging for a virtual host, all HTTP requests for the virtual host are saved in a separate log file.

Tasks

“Enabling and Configuring an HTTP Log” on page 253-28

Related Topics

For more information about log files, refer to [Setting Up HTTP Access Logs](#). This topic includes information about using the extended format in HTTP logs.

For information about VirtualHosts, refer to "Configuring Virtual Hosting" under [Configuring WebLogic Server Web Components](#).

Attributes

Table 446-1

Attribute Label	Description	Value Constraints
Enable HTTP Logging	Enables logging of HTTP requests. <i>MBean:</i> weblogic.management.configuration.WebServerMBean <i>Attribute:</i> LoggingEnabled	<i>Default:</i> true <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false <i>Configurable:</i> yes
HTTP Log File Name	The HTTP request log file The name of the file that stores HTTP requests. If the pathname is not absolute, the path is assumed to be relative to the root directory of the machine on which the server is running. This value is relevant only if HTTP logging is enabled. To include a time or date stamp in the file name when the log file is rotated, add <code>java.text.SimpleDateFormat</code> variables. Surround each variable with percentage (%) characters. For example, <code>access_%yyyy%_%MM%_%dd%_%hh%_%mm%.log</code> If you do not include a time and date stamp, the rotated log files are numbered in order of creation <code>filenamennnnn</code> , where <i>filename</i> is the name configured for the log file. <i>MBean:</i> weblogic.management.configuration.WebServerMBean <i>Attribute:</i> LogFileName	<i>Default:</i> access.log <i>Configurable:</i> yes

Table 446-1

Attribute Label	Description	Value Constraints
Format	<p>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.</p> <p><i>MBean:</i> weblogic.management.configuration.WebServerMBean</p> <p><i>Attribute:</i> LogFileFormat</p>	<p><i>Default:</i> "common"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "common"■ "extended" <p><i>Configurable:</i> yes</p> <p><i>Dynamic:</i> yes</p>
Log Buffer Size	<p>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 <code>LogFileFlushSecs</code> property to determine the frequency with which the server checks the size of the buffer.</p> <p><i>MBean:</i> weblogic.management.configuration.WebServerMBean</p> <p><i>Attribute:</i> LogFileBufferKBytes</p>	<p><i>Units:</i> kilobytes</p> <p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 1024</p> <p><i>Default:</i> 8</p> <p><i>Configurable:</i> yes</p>

Table 446-1

Attribute Label	Description	Value Constraints
Rotation Type	<p>Criteria for moving old HTTP requests to a separate log file:</p> <ul style="list-style-type: none"> ■ <code>size</code>. When the log file reaches the size that you specify in <code>MaxLogFileSizeKBytes</code>, the server renames the file as <code>LogFileName.n</code>. ■ <code>date</code>. At each time interval that you specify in <code>LogRotationPeriodMin</code>, the server renames the file as <code>LogFileName.n</code>. <p>After the server renames a file, subsequent messages accumulate in a new file with the name that you specified in <code>LogFileName</code>.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.WebServerMBean</code></p> <p><i>Attribute:</i> <code>LogRotationType</code></p>	<p><i>Default:</i> "size"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ "size" ■ "date" <p><i>Configurable:</i> yes</p>
Maximum Log File Size	<p>The file size (1 - 65535 kilobytes) that triggers the server to move log messages to a separate file. After the log file reaches the specified size, the next time the server checks the file size, it will rename the current log file as <code>FileName.n</code> and create a new one to store subsequent messages.</p> <p>0 causes the file to grow indefinitely.</p> <p>This property is relevant only if you choose to rotate files by <code>size</code>.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.WebServerMBean</code></p> <p><i>Attribute:</i> <code>MaxLogFileSizeKBytes</code></p>	<p><i>Units:</i> kilobytes</p> <p><i>Minimum:</i> 0</p> <p><i>Default:</i> 5000</p> <p><i>Configurable:</i> yes</p>

Table 446-1

Attribute Label	Description	Value Constraints
Rotation Period	<p>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.</p> <p><i>MBean:</i> weblogic.management.configuration.WebServerMBean</p> <p><i>Attribute:</i> LogRotationPeriodMins</p>	<p><i>Units:</i> minutes</p> <p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 1440</p> <p><i>Configurable:</i> yes</p>
Rotation Time	<p>Determines the start time for a time-based rotation sequence. At the time that this value specifies, the server renames the log file. Thereafter, the server renames the log file at an interval that you specify in LogRotationPeriodMins.</p> <p>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.</p> <p>If the time that you specify has already past, then the server starts its file rotation immediately.</p> <p><i>MBean:</i> weblogic.management.configuration.WebServerMBean</p> <p><i>Attribute:</i> LogRotationTimeBegin</p>	<p><i>Configurable:</i> yes</p>
Limit Number Of Retained Log Files	<p>Specifies whether the number of files that this WebLogic Server creates to store old messages should be limited. After the server reaches this limit, it overwrites the oldest file.</p> <p><i>MBean:</i> weblogic.management.configuration.LogMBean</p> <p><i>Attribute:</i> NumberOfFilesLimited</p>	<p><i>Default:</i> false</p>

Table 446-1

Attribute Label	Description	Value Constraints
Log Files to Retain	<p>The maximum number of log files that this WebLogic Server creates when it rotates the log. (This field is relevant only if you check the Limit Number Of Retained Log Files box.)</p> <p><i>MBean:</i> weblogic.management.configuration.LogMBean</p> <p><i>Attribute:</i> FileCount</p>	<p><i>Default:</i> 7</p>
Flush Every	<p>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 LogFileBufferKBytes property, the server writes the data in the buffer to the HTTP request log file.</p> <p><i>MBean:</i> weblogic.management.configuration.WebServerMBean</p> <p><i>Attribute:</i> LogFileFlushSecs</p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 360</p> <p><i>Default:</i> 60</p> <p><i>Configurable:</i> yes</p>

Server --> Logging --> JDBC

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

On the **Server—Logging—JDBC** tab, you can enable JDBC logging and specify a log file name for the JDBC log.

Tasks

[“Enabling JDBC Logging” on page 253-27](#)

Related Topics

[“JDBC” on page 111-1](#)

[Introduction to WebLogic JDBC](#) in *Programming WebLogic JDBC*

Attributes

Table 447-1

Attribute Label	Description	Value Constraints
Enable JDBC Logging	Determines whether this server maintains a JDBC log file. <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> JDBCLoggingEnabled	<i>Valid values:</i> <ul style="list-style-type: none">■ true■ false
JDBC Log File Name	The name of the JDBC log file. If the pathname is not absolute, the path is assumed to be relative to the server's root directory. If the log has no path element and is atomic, ie. jdbc.log to avoid name space conflicts the file will be placed relative to the root directory in <code>./SERVER_NAME/</code> <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> JDBCLogFileName	<i>Default:</i> jdbc.log

Server --> Logging --> JTA

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The **Server—Logging—JTA** tab determines the location of the transaction log files and specifies a policy for writing transaction log records to disk.

BEA recommends that you locate transaction log files on a highly available file system, for example, on a RAID device. To take advantage of the migration capability of the Transaction Recovery Service for servers in a cluster, you must store the transaction log in a location that is available to a server and its backup servers, preferably on a dual-ported SCSI disk or on a Storage Area Network (SAN).

Warning: On Windows, the Direct-Write transaction log file write policy may leave transaction data in the on-disk cache without immediately writing it to disk. This is not transactionally safe because a power failure can cause loss of on-disk cache data. To prevent cache data loss when using the Direct-Write transaction log file write policy on Windows, disable all write caching for the disk (enabled by default) or use a battery backup for the storage system. See [“Disabling the On-Disk Cache For a Disk Drive on Windows 2000” on page 235-3](#) for instructions

Tasks

[“Setting the Transaction Log File Location \(Prefix\)” on page 237-11](#)

[“Setting the Transaction Log File Write Policy” on page 237-12](#)

Related Topics

(May require an Internet connection.)

[“Transaction Log Files” on page 237-10](#)

[Introducing Transactions](#) in *Programming WebLogic JTA*

[Configuring and Managing Transactions](#) in *Programming WebLogic JTA*

Attributes

Table 448-1

Attribute Label	Description	Value Constraints
Transaction Log File Prefix	<p>The path prefix for the server's JTA transaction log files. If the pathname is not absolute, the path is assumed to be relative to the server's root directory.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> TransactionLogFilePrefix</p>	<i>Default:</i> "/"

Table 448-1

Attribute Label	Description	Value Constraints
Transaction Log File Write Policy	<p>The policy used for writing transaction log records to disk:</p> <ul style="list-style-type: none">■ Cache-Flush—Flushes operating system and on-disk caches after each entry to the transaction log. Transactions cannot commit until the commit record is written to stable storage.■ Direct-Write—Forces the operating system to write transaction log entries directly to disk with each write. To be transactionally safe, this option may require additional OS parameter settings. <p>Direct-Write performs as well or better than Cache-Flush, depending on operating system and OS parameter settings, and is available on Windows, HP-UX, and Solaris.</p>	<i>Default:</i> Cache-Flush



Socket

This page displays IP address and protocol information for this server's opened sockets.



Inflight JTA Transactions

The Inflight JTA Transactions table shows information about transactions currently being processed on this server. Click on a Transaction Id to see more details about the transaction and manually resolve the transaction. You can click column headings in the table to sort the information in the table. You can also click Customize this view to select the columns to display in the table.

- For information about manually resolving a transaction, see [“Manually Resolving Current \(Inflight\) Transactions” on page 237-6](#).
- For information about configuring WebLogic JTA in the Administration Console, see [“Configuring JTA” on page 237-2](#).
- To learn more about WebLogic JTA, see [Programming WebLogic JTA](#).

Attributes

Transaction Id—The transaction identifier assigned by the Transaction Manager. You can click a Transaction Id to see details about the transaction and manually resolve the transaction.

Name—The transaction name as specified in the application that created the transaction.

Status—The overall status of the transaction. For valid values, see Table 237-1.

Seconds Active—The number of seconds since the transaction was created.

Servers—A list of all servers that participate in the transaction.

Resources—A list of all resources that participate in the transaction. The list also includes the status for each resource.



Server --> Services --> Bridge

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

On this tab, you can configure the size of the WebLogic Messaging Bridge execute thread pool. 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 a messaging bridge to use the WebLogic Server default thread pool.

You can also monitor the status of all the messaging bridges in the domain by clicking the Monitor all Messaging Bridge Runtimes link.

Tasks

[“Monitoring All Messaging Bridges” on page 289-23](#)

[“Configuring a Messaging Bridge Instance” on page 289-11](#)

Related Topics

[“Tuning WebLogic Server”](#) and [“Tuning WebLogic Server Applications”](#) in the *Performance and Tuning Guide*.

Attributes

Table 451-1

Attribute Label	Description	Value Constraints
Messaging Bridge Thread Pool Size	The size of the messaging bridge execute thread pool.	<i>Minimum:</i> 0 <i>Maximum:</i> 65534 <i>Default:</i> 5 <i>Configurable:</i> yes

Server --> Services --> File T3

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to assign selected FileT3 file systems to a server.

Tasks

“Assign Servers for a FileT3 File System” on page 109-2

“Create a File System” on page 109-1

Related Topics

[Using WebLogic File Services](#)

Attributes

Table 452-1

Attribute Label	Description	Value Constraints
Targets	The targets in the current domain on which this item can be deployed.	<i>Dynamic:</i> yes



Server --> Services --> JDBC

[Tasks](#) [Related Topics](#)

Overview

In the **Server—Services—JDBC** tab, you can view all JDBC objects deployed on the selected server. Click an object name to go to the property pages for that object.

Tasks

[“Monitoring Connections in a JDBC Connection Pool” on page 127-36](#)

[“Creating and Configuring a JDBC Connection Pool” on page 127-4](#)

[“Creating and Configuring a JDBC Data Source” on page 141-5](#)

[“Creating and Configuring a JDBC MultiPool” on page 149-2](#)

Related Topics

[“Creating and Deploying JDBC Components—Connection Pools, MultiPools, and Data Sources” on page 111-1](#)

[“Connection Pool and Data Source Configuration Guidelines” on page 127-15](#)

[Introduction to WebLogic JDBC](#) in *Programming WebLogic JDBC*



Server --> Services --> JMS

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

On this tab, you can define whether to use the default JMS connection factories and configure the size of the JMS execute thread pool.

Default JMS Connection Factories

WebLogic Server defines two default connection factories, which can be looked up using the following JNDI names:

- `weblogic.jms.ConnectionFactory`
- `weblogic.jms.XAConnectionFactory` (the XA Connection Factory Enabled attribute is switched on for enabling JTA user-transactions)

An XA factory is required for JMS applications to use JTA user-transactions, but is not required for transacted sessions. All other preconfigured attributes for the default connection factories are set to the same default values as a user-defined connection factory. If the preconfigured settings of the default factories are appropriate for your application, you do not need to configure any additional factories for your application.

Note: When using the default connection factories, you have no control over targeting the WebLogic Server instances where the connection factory may be deployed. However, you can disable the default connection factories on a per-server basis. To deploy a connection factory on independent servers, on specific servers within a cluster, or on an entire cluster, you need to configure a connection factory and specify the appropriate server targets.

JMS Thread Pool Size

On the server, incoming JMS related requests execute in the JMS execute queue/thread pool. Additional work that cannot be completed in the request thread is forwarded to the "default" execute queue.

Tasks

[“Configuring a JMS Connection Factory”](#) on page 232-10

[“JMS Connection Factory Tasks”](#) on page 232-8

Related Topics

[“WebLogic JMS Fundamentals”](#) in *Programming WebLogic JMS*

[“Using Transactions With WebLogic JMS”](#) in *Programming WebLogic JMS*

[“Developing a WebLogic JMS Application”](#) in *Programming WebLogic JMS*

[“Tuning WebLogic Server”](#) and [“Tuning WebLogic Server Applications”](#) in the *Performance and Tuning Guide*.

Attributes

Table 454-1

Attribute Label	Description	Value Constraints
Enable Default JMS Connection Factories	<p>Enables the default JMS connection factories:</p> <ul style="list-style-type: none">■ <code>weblogic.jms.ConnectionFactory</code>■ <code>weblogic.jms.XAConnectionFactory</code> <p>The XA Connection Factory Enabled setting is turned on by default on the default transaction connection factory, <code>weblogic.jms.XAConnectionFactory</code>. An XA factory is required for JMS applications to use JTA user-transactions, but is not required for transacted sessions.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>JMSDefaultConnectionFactoriesEnabled</code></p>	<p><i>Default:</i> true</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false <p><i>Dynamic:</i> yes</p>
JMS Thread Pool Size	<p>The size of the JMS execute thread pool.</p> <p>Note: Incoming RMI calls execute in the JMS execute queue/thread pool, if one exists; otherwise, they execute in the default execute queue.</p> <p>Additional executes (work that cannot be completed in the initial RMI thread) are executed in the default execute queue.</p> <p>The difference in setting up a JMS-specific thread pool is that JMS will not be starved by other execute threads and vice versa.</p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 65534</p> <p><i>Default:</i> 15</p> <p><i>Configurable:</i> yes</p>



Server --> Services --> Jolt

[Tasks](#) [Related Topics](#)

Overview

Jolt is a Java-based client API that manages requests to BEA Tuxedo services via a Jolt Service Listener (JSL) running on the Tuxedo server. The Jolt API is embedded within the WebLogic API, and is accessible from a servlet or any other BEA WebLogic application. The Jolt Java client class library can be used in HTTP servlets running in WebLogic Server and provides an interface between HTML browser clients and BEA Tuxedo services.

Jolt connection pools allow you to connect WebLogic Server clients to BEA Tuxedo domains. The server creates the Jolt connection pools at startup and assigns connections to WebLogic Server clients as needed.

When one or more Jolt connection pools are configured, this Jolt page displays key information about each of them.

Tasks

[“Create a Jolt Startup & Shutdown Class” on page 249-2](#)

[“Create a Jolt Connection Pool” on page 249-3](#)

[“Monitor Active Instances of a Jolt Connection Pool” on page 249-6](#)

Related Topics

Configuring Jolt for WebLogic Server at
<http://e-docs.bea.com/tuxedo/tux80/atmi/configu3.htm>

BEA Jolt at <http://e-docs.bea.com/tuxedo/tux80/interm/jolt.htm>



Server --> Services --> Mail

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

WebLogic Server includes the JavaMail API version 1.1.3 reference implementation from Sun Microsystems. Using the JavaMail API, you can add email capabilities to your WebLogic Server applications. JavaMail provides access from Java applications to IMAP- and SMTP-capable mail servers on your network or the Internet. It does not provide mail server functionality; so you must have access to a mail server to use JavaMail.

Tasks

“Configuring a New Mail Session” on page 274-1

“Cloning a Mail Session” on page 274-2

“Deleting a Mail Session” on page 274-3

Related Topics

“Mail” on page 274-1

See “Using JavaMail with WebLogic Server” in *Developing WebLogic Server Applications*.

Attributes

Table 456-1

Attribute Label	Description	Value Constraints
Targets	The targets in the current domain on which this item can be deployed. <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> Targets	<i>Dynamic:</i> yes

Server --> Services --> WTC

[Tasks](#) [Related Topics](#)

Overview

Use this tab to assign selected WTCServers to a server.

Tasks

“Assign a WTC Service to a Server” on page 619-2

Related Topics

[Configuring WebLogic Tuxedo Connector at {DOCROOT}/wtc_admin/WTC_Admin_Install.html](#)



Server --> Services --> XML

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to associate an existing XML Registry with a WebLogic Server instance.

Also use this tab to configure the external entity cache, such as the cache memory size, the cache disk size, and how long WebLogic Server should wait between refreshing the cache.

Tasks

“Configuring the External Entity Cache” on page 640-7

“Monitoring the External Entity Cache” on page 640-8

Related Topics

[External Entity Configuration Tasks at
{DOCR00T}/xml/xml_admin.html#admin009](#)

Attributes

Table 458-1

Attribute Label	Description	Value Constraints
XMLRegistry	The xMLRegistry attribute of the ServerMBean object <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> XMLRegistry	<i>Configurable:</i> yes
Cache Memory Size	The memory size in KBytes of the cache. <i>MBean:</i> weblogic.management.configuration.XMLEntityCacheMBean <i>Attribute:</i> CacheMemorySize	<i>Minimum:</i> 0 <i>Default:</i> 500 <i>Dynamic:</i> yes
Cache Disk Size	The disk size in MBytes of the cache. <i>MBean:</i> weblogic.management.configuration.XMLEntityCacheMBean <i>Attribute:</i> CacheDiskSize	<i>Minimum:</i> 0 <i>Default:</i> 5 <i>Dynamic:</i> yes
Cache Timeout Interval	The default timeout interval in seconds for the cache. <i>MBean:</i> weblogic.management.configuration.XMLEntityCacheMBean <i>Attribute:</i> CacheTimeoutInterval	<i>Minimum:</i> 0 <i>Default:</i> 120 <i>Dynamic:</i> yes

Server

The Servers page summarizes the status of each WebLogic Server instance that has been configured for the domain. To change the columns that the summary table displays, click the [Customize this view](#) link.

To modify the configuration of an existing server, in the summary table click the name of the server.

To create a new instance of WebLogic Server, either click the [Configure a new server](#) link or click the Clone icon in the summary table. Server names must be unique within a domain. For example, if you create a server instance named `ManagedServer1` in a domain named `DomainA`, you cannot create another server instance named `ManagedServer1` in `DomainA`.

To delete a server, click the Delete icon in the summary table.

For more information, see the following tasks:

- “Adding and Removing Servers in an Existing Domain” on page 495-2
- “Cloning a Server” on page 495-4
- “Deleting a Managed Server” on page 495-5
- “Starting Managed Servers from the Administration Console” on page 497-5

For information on changing the information that this tab displays, refer to “Customizing Table Views” on page 6-15.



ServletSessionRuntime

This table allows you to view and manage the Servlets you have deployed on this instance of WebLogic Server. It allows you to choose the attributes by which you want to sort the Servlets and to clone or delete selected Servlets.

To select attributes to be shown in the table Click the *Customize this View* link.

- For more information, see
- “Configuring and Deploying a New Enterprise Application or Web Service” on page 26-2



Server --> Protocols --> jCOM

You must be in the Admin security role to see the Security, CompatibilitySecurity, and Services —> jCOM nodes in the navigation tree. [Related Topics](#) [Attributes](#)

Overview

Use this tab to enable and disable jCOM and configure various aspects of its behavior, including:

- The NT Authentication Host address
- Whether it runs in native or non-native mode
- Whether the server logs memory usage
- Whether prefetch enumeration is enabled
- Whether jCOM is initialized in native mode

Note: You must be in the Admin security role to see the Security, CompatibilitySecurity, and Services —> jCOM nodes in the navigation tree.

Tasks

“Enabling and Configuring jCOM” on page 495-10

Related Topics

[Programming WebLogic jCOM at {DOCROOT}/jcom/index.html.](#)

Attributes

Table 461-1

Attribute Label	Description	Value Constraints
Enable COM	Whether or not COM support is enabled on the plaintext port. (COM is not supported on the SSL port.) <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> COMEnabled	<i>Default:</i> false <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false
NT Authentication Host	The address of the primary domain controller to be used for authenticating clients. If this property is not set, COM clients will not be authenticated. <i>MBean:</i> weblogic.management.configuration.COMMBean <i>Attribute:</i> NTAUTHHost	
Enable Native Mode	Use native DLLs to allow Java objects to interact with COM Objects. Only supported on Windows. <i>MBean:</i> weblogic.management.configuration.COMMBean <i>Attribute:</i> NativeModeEnabled	<i>Default:</i> false <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false
Enable Memory Logging	Enables logging of memory usage <i>MBean:</i> weblogic.management.configuration.COMMBean <i>Attribute:</i> MemoryLoggingEnabled	<i>Default:</i> false <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false

Table 461-1

Attribute Label	Description	Value Constraints
Prefetch Enumeration	<p>Some COM methods return a COM VariantEnumeration type. The java2com tool automatically converts the returned type into a java.lang.Enumeration. This is not a perfect match since COM enumerations have no equivalent to the hasMoreElements() call. The client must continue to call nextElement until a NoSuchElementException occurs. Setting this property will cause jCOM to prefetch the next element in behind the scenes and return the correct value when hasMoreElements is called.</p> <p><i>MBean:</i> weblogic.management.configuration.COMMBean</p> <p><i>Attribute:</i> PrefetchEnums</p>	<p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false
Apartment Threaded	<p>Controls the flag that is used to initialize COM in native mode. By default, jCOM initializes COM using the COINIT_MULTITHREADED_FLAG. If the server logs a Class Not Registered Message when using native mode, try setting this property.</p> <p>(COINIT_APARTMENTTHREADED)</p> <p><i>MBean:</i> weblogic.management.configuration.COMMBean</p> <p><i>Attribute:</i> ApartmentThreaded</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false



Server --> Services --> Virtual Hosts

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Virtual hosting allows you to define host names to which servers or clusters respond. When you use virtual hosting you use DNS to specify one or more host names that map to the IP address of a WebLogic Server or cluster and you specify which Web Applications are served by each virtual host.

Tasks

“Assigning a VirtualHost” on page 540-5

“Associating a Virtual Host with a Server” on page 540-5

“Removing an Associated Virtual Host” on page 540-6

Related Topics

For additional information about VirtualHosts, please see the *Administration Guide*, [Overview of WebLogic Server HTTP Services](#).

Attributes

Table 462-1

Attribute Label	Description	Value Constraints
Targets	The targets in the current domain on which this item can be deployed. <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> Targets	<i>Dynamic:</i> yes

Server --> Services --> Web Services

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to configure the reliable SOAP messaging attributes for the current WebLogic Server in its role as both a *sender* and a *receiver* of a reliable SOAP message to and from a Web service running on a different WebLogic Server.

Tasks

“Configuring Web Service Reliable SOAP Messaging” on page 574-8

Related Topics

[Using Reliable SOAP Messaging at {DOCROOT}/webserv/reliable.html](#)

Attributes

Table 463-1

Attribute Label	Description	Value Constraints
Store	<p>The persistent JMS store used by WebLogic Server to persist the reliable SOAP messages that it either sends or receives.</p> <p><i>MBean:</i> weblogic.management.configuration.WSReliableDeliveryPolicyMBean</p> <p><i>Attribute:</i> Store</p>	
Default Retry Count	<p>The default maximum number of times that the sender should attempt to redeliver a message that the receiver WebLogic Web service has not yet acknowledged.</p> <p><i>MBean:</i> weblogic.management.configuration.WSReliableDeliveryPolicyMBean</p> <p><i>Attribute:</i> DefaultRetryCount</p>	<i>Default:</i> 10
Default Retry Interval	<p>The default minimum number of seconds that the sender should wait between retries if the receiver does not send an acknowledgement of receiving the message, or if the sender detects a communications error while attempting to send a message.</p> <p><i>MBean:</i> weblogic.management.configuration.WSReliableDeliveryPolicyMBean</p> <p><i>Attribute:</i> DefaultRetryInterval</p>	<i>Default:</i> 6000

Table 463-1

Attribute Label	Description	Value Constraints
Default Time To Live	<p>The default minimum number of seconds that the receiver of the reliably sent message should persist the message in its storage.</p> <p>If the DefaultTimeToLive number of message have passed since the message was first sent, the sender should not resent a message with the same message id.</p> <p>If a sender cannot send a message successfully before the DefaultTimeToLive has passed, the sender should report a delivery failure. The receiver, after recovering from a crash, will not dispatch saved messages that have expired.</p> <p><i>MBean:</i> weblogic.management.configuration.WSReliableDeliveryPolicyMBean</p> <p><i>Attribute:</i> DefaultTimeToLive</p>	<i>Default:</i> 60000



Server --> Protocols --> IIOP

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The Servers --> Protocols --> IIOP tab allows you to configure attributes for Internet Interop-Orb-Protocol (IIOP) connections.

Tasks

“Enabling and Configuring the IIOP Protocol” on page 495-9

Related Topics

[Programming WebLogic RMI over IIOP](#)

Attributes

Table 464-1

Attribute Label	Description	Value Constraints
Enable IIOP	Whether or not IIOP support is enabled for both the SSL and non-SSL ports. <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> IIOPEnabled	<i>Default:</i> true <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false

Advanced Attributes

Table 464-2

Attribute Label	Description	Value Constraints
Default GIOP Version	<p>Specify the default minor GIOP version for IIOP messages. This attribute is useful for client orbs with broken GIOP 1.2 implementations.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.IIOPMBean</code></p> <p><i>Attribute:</i> <code>DefaultMinorVersion</code></p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2</p> <p><i>Default:</i> 2</p> <p><i>Configurable:</i> yes</p>
Transaction Mechanism	<p>Specify whether IIOP should use the WebLogic-specific JTA to propagate transactions or the OMG-specified OTS. It is not possible to use both since it affects the wayr transactions are negotiated.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.IIOPMBean</code></p> <p><i>Attribute:</i> <code>TxMechanism</code></p>	<p><i>Default:</i> "OTS"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "OTS"■ "JTA" <p><i>Configurable:</i> yes</p>
Default Char Codeset	<p>Specify what codeset should be used for the default native character codeset. This is required to interoperate with some Orbs that do not do codeset negotiation, or do it incorrectly. Setting this to anything other than US-ASCII will cause failure against JDK 1.3.1 clients.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.IIOPMBean</code></p> <p><i>Attribute:</i> <code>DefaultCharCodeset</code></p>	<p><i>Default:</i> "US-ASCII"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "US-ASCII"■ "UTF-8"■ "ISO-8859-1" <p><i>Configurable:</i> yes</p>

Table 464-2

Attribute Label	Description	Value Constraints
Default Wide Char Codeset	<p>Specify what codeset should be used for the default native wide character codeset. This is required to interoperate with some Orbs that do not do codeset negotiation, or do it incorrectly. Setting this to anything other than UCS-2 will cause failure against JDK 1.3.1 clients.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.IIOPMBean</code></p> <p><i>Attribute:</i> <code>DefaultWideCharCodeset</code></p>	<p><i>Default:</i> "UCS-2"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "UCS-2"■ "UTF-16"■ "UTF-8"■ "UTF-16BE"■ "UTF-16LE"
Default IIOP Username	<p>The default IIOP user.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>DefaultIIOPUser</code></p>	<p><i>Default:</i> null</p>
Default IIOP Password	<p>The password for the default IIOP user.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>DefaultIIOPPassword</code></p>	<p><i>Default:</i> null</p> <p><i>Encrypted:</i> yes</p>



Server --> Protocols --> HTTP

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The Servers --> Protocols --> HTTP tab allows you to configure attributes for HTTP servlets.

Tasks

“Configuring the HTTP Protocol” on page 495-7

Related Topics

[Programming HTTP Servlets](#)

Attributes

Table 465-1

Attribute Label	Description	Value Constraints
Default Server Name	The HTTP defaultServerName <i>MBean:</i> weblogic.management. configuration. WebServerMBean <i>Attribute:</i> DefaultServerName	<i>Default:</i> null <i>Configurable:</i> yes

Table 465-1

Attribute Label	Description	Value Constraints
Post Timeout	Timeout (in seconds) for reading HTTP POST data in a servlet request. <i>MBean:</i> weblogic.management.configuration.WebServerMBean <i>Attribute:</i> PostTimeoutSecs	<i>Minimum:</i> 0 <i>Maximum:</i> 120 <i>Default:</i> 30
Max Post Size	Max Post Size (in bytes) for reading HTTP POST data in a servlet request. MaxPostSize < 0 means unlimited <i>MBean:</i> weblogic.management.configuration.WebServerMBean <i>Attribute:</i> MaxPostSize	<i>Units:</i> bytes <i>Default:</i> -1
Enable Keepalives	Returns whether or not HTTP keep-alive is enabled <i>MBean:</i> weblogic.management.configuration.WebServerMBean <i>Attribute:</i> KeepAliveEnabled	<i>Default:</i> true <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false
Duration	Number of seconds to maintain HTTP keep-alive before timing out the request. <i>MBean:</i> weblogic.management.configuration.WebServerMBean <i>Attribute:</i> KeepAliveSecs	<i>Units:</i> seconds <i>Minimum:</i> 5 <i>Maximum:</i> 120 <i>Default:</i> 30
HTTPS Duration	Number of seconds to maintain HTTPS keep-alive before timing out the request. <i>MBean:</i> weblogic.management.configuration.WebServerMBean <i>Attribute:</i> HttpsKeepAliveSecs	<i>Units:</i> seconds <i>Minimum:</i> 30 <i>Maximum:</i> 360 <i>Default:</i> 60

Advanced Attributes

Table 465-2

Attribute Label	Description	Value Constraints
Frontend Host	<p>The HTTP frontendHost is set when the Host information coming from the URL may be inaccurate due to the presence of a firewall or proxy. If this parameter is set, the HOST header is ignored and this value is always used.</p> <p><i>MBean:</i> weblogic.management.configuration.WebServerMBean <i>Attribute:</i> FrontendHost</p>	<p><i>Default:</i> null <i>Configurable:</i> yes</p>
Frontend HTTP Port	<p>The frontendHTTPPort is set when the Port information coming from the URL may be inaccurate due to the presence of a firewall or proxy. If this parameter is set, the HOST header is ignored and this value is always used.</p> <p><i>MBean:</i> weblogic.management.configuration.WebServerMBean <i>Attribute:</i> FrontendHTTPPort</p>	<p><i>Default:</i> 0 <i>Configurable:</i> yes</p>
Frontend HTTPS Port	<p>The frontendHTTPSPort is set when the Port information coming from the URL may be inaccurate due to the presence of a firewall or proxy. If this parameter is set, the HOST header is ignored and this value is always used.</p> <p><i>MBean:</i> weblogic.management.configuration.WebServerMBean <i>Attribute:</i> FrontendHTTPSPort</p>	<p><i>Default:</i> 0 <i>Configurable:</i> yes</p>

Table 465-2

Attribute Label	Description	Value Constraints
WAP Enabled	Enables WAP <i>MBean:</i> weblogic.management.configuration.WebServerMBean <i>Attribute:</i> WAPEnabled	<i>Default:</i> false <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false
Send Server Header	Returns whether or not to send a response Server header <i>MBean:</i> weblogic.management.configuration.WebServerMBean <i>Attribute:</i> SendServerHeaderEnabled	<i>Default:</i> true <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false
Accept Context Path In Get Real Path	Beginning with this release inclusion of the contextPath in the virtualPath to the context.getRealPath() will not be allowed as it breaks the case when the subdirectories have the same name as contextPath. In order to support applications which might have been developed according to the old behaviour we are providing a compatibility switch. This switch will be deprecated in future releases. <i>MBean:</i> weblogic.management.configuration.WebServerMBean <i>Attribute:</i> AcceptContextPathInGetRealPath	<i>Default:</i> false <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false

Table 465-2

Attribute Label	Description	Value Constraints
HTTP Max Message Size	<p>Specify the maximum HTTP message size allowable in a message header. This attribute attempts to prevent a denial of service attack whereby a caller attempts to force the server to allocate more memory than is available thereby keeping the server from responding quickly to other requests. This setting only applies to connections that are initiated using one of the default ports (ServerMBean <code>setListenPort</code> and <code>setAdministrationPort</code> or SSLMBean <code>setListenPort</code>). Connections on additional ports are tuned via the <code>NetworkChannelMBean</code>.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>MaxHTTPMessageSize</code></p>	<p><i>Units:</i> bytes</p> <p><i>Minimum:</i> 4096</p> <p><i>Maximum:</i> 2000000000</p> <p><i>Default:</i> -1</p> <p><i>Configurable:</i> yes</p> <p><i>Dynamic:</i> yes</p>
HTTP Message Timeout	<p>Specify the maximum number of seconds spent waiting for a complete HTTP message to be received. This attribute helps guard against denial of service attacks in which a caller indicates that they will be sending a message of a certain size which they never finish sending. This setting only applies to connections that are initiated using one of the default ports (ServerMBean <code>setListenPort</code> and <code>setAdministrationPort</code> or SSLMBean <code>setListenPort</code>). Connections on additional ports are tuned via the <code>NetworkChannelMBean</code>.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>CompleteHTTPMessageTimeout</code></p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 480</p> <p><i>Default:</i> -1</p> <p><i>Configurable:</i> yes</p> <p><i>Dynamic:</i> yes</p>



Servers --> Protocols --> Channels

[Tasks](#) [Related Topics](#)

Overview

Use this tab to configure a network channel for the server instance. A network channel allows you to segregate network traffic by Network Interface Card (NIC) and to listen for requests on multiple ports.

Tasks

“Configuring a Custom Network Channel for a Non-Clustered Server” on page 495-15

Related Topics

["Understanding Network Channels"](#)

["Configuring Network Channels"](#)



Server --> Monitoring --> Security

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Displays the current settings for the attributes used to protect user accounts.

Tasks

“Protecting User Accounts” on page 428-5

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Attributes

Table 467-1

Attribute Label	Description	Value Constraints
User Lockout Total Count	The cumulative number of user lockouts done on this server <i>MBean:</i> weblogic.management.runtime.ServerSecurityRuntimeMBean <i>Attribute:</i> UserLockoutTotalCount	
Total Invalid Logins	The cumulative number of invalid logins attempted on this server <i>MBean:</i> weblogic.management.runtime.ServerSecurityRuntimeMBean <i>Attribute:</i> InvalidLoginAttemptsTotalCount	
Total Login Attempts while Locked	The cumulative number of invalid logins attempted on this server attempted while the user was locked <i>MBean:</i> weblogic.management.runtime.ServerSecurityRuntimeMBean <i>Attribute:</i> LoginAttemptsWhileLockedTotalCount	
Total Users Unlocked	The number times we have unlocked a user on this server <i>MBean:</i> weblogic.management.runtime.ServerSecurityRuntimeMBean <i>Attribute:</i> UnlockedUsersTotalCount	

Table 467-1

Attribute Label	Description	Value Constraints
Invalid Logins High	<p>The highwater number of users with outstanding invalid login attempts for this server</p> <p><i>MBean:</i> weblogic.management.runtime.ServerSecurityRuntimeMBean</p> <p><i>Attribute:</i> InvalidLoginUsersHighCount</p>	
Locked Users	<p>The number of currently locked users on this server</p> <p><i>MBean:</i> weblogic.management.runtime.ServerSecurityRuntimeMBean</p> <p><i>Attribute:</i> LockedUsersCurrentCount</p>	



Server --> Monitoring --> Performance

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This page displays performance metrics related to pending and processed requests for the current server.

Tasks

“Monitoring a Server Instance from the Administration Console” on page 495-21

Related Topics

["Monitoring Servers"](#)

Attributes

Attribute Label	Description	Value Constraints
Idle Threads	<p>The number of idle threads assigned to the queue.</p> <p><i>MBean:</i> weblogic.management.runtime.ExecuteQueueRuntimeMBean</p> <p><i>Attribute:</i> ExecuteThreadCurrentIdleCount</p>	<p><i>Default:</i> 0</p> <p><i>Readable:</i> yes</p>

Attribute Label	Description	Value Constraints
PendingRequestOldestTime	<p>The time that the longest waiting request was placed in the queue.</p> <p><i>MBean:</i> weblogic.management.runtime.ExecuteQueueRuntimeMBean</p> <p><i>Attribute:</i> PendingRequestOldestTime</p>	<p><i>Default:</i> 0</p> <p><i>Readable:</i> yes</p>
Throughput	<p>The number of requests that have been processed by the queue.</p> <p><i>MBean:</i> weblogic.management.runtime.ExecuteQueueRuntimeMBean</p> <p><i>Attribute:</i> ServicedRequestTotalCount</p>	<p><i>Default:</i> 0</p> <p><i>Readable:</i> yes</p>
Queue Length	<p>The number of waiting requests in the queue.</p> <p><i>MBean:</i> weblogic.management.runtime.ExecuteQueueRuntimeMBean</p> <p><i>Attribute:</i> PendingRequestTotalCount</p>	<p><i>Default:</i> 0</p> <p><i>Readable:</i> yes</p>
Memory Usage	<p>The current amount of memory (in bytes) that is available in the JVM heap.</p>	<p><i>Default:</i> 0</p> <p><i>Readable:</i> yes</p>

Server --> Monitoring --> JTA

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

On the Server—Monitoring—JTA tab, you can monitor transactions for the selected server in the domain. Transaction statistics are displayed for a specific server, not the entire domain, even though transaction *settings* apply to the entire domain. For more information, see “Monitoring Transactions” on page 237-5.

Tasks

“Viewing Transaction Statistics for a Server” on page 237-5

“Viewing Transaction Statistics for Named Transactions” on page 237-5

“Viewing Transaction Statistics for Server Resources” on page 237-6

“Viewing Current (Inflight) Transactions for a Server” on page 237-6

“Manually Resolving Current (Inflight) Transactions” on page 237-6

Related Topics

[Introducing Transactions](#)

Attributes

Table 469-1

Attribute Label	Description	Value Constraints
Total Transactions	<p>The total number of transactions processed. This total includes all committed, rolled back and heuristic transaction completions.</p> <p><i>MBean:</i> weblogic.management.runtime.JTARuntimeMBean</p> <p><i>Attribute:</i> TransactionTotalCount</p>	
Total Committed	<p>The number of committed transactions.</p> <p><i>MBean:</i> weblogic.management.runtime.JTARuntimeMBean</p> <p><i>Attribute:</i> TransactionCommittedTotalCount</p>	
Total Rolled Back	<p>The number of transactions that were rolled back.</p> <p><i>MBean:</i> weblogic.management.runtime.JTARuntimeMBean</p> <p><i>Attribute:</i> TransactionRolledBackTotalCount</p>	
Timeout Rollbacks	<p>The number of transactions that were rolled back due to a timeout expiration.</p> <p><i>MBean:</i> weblogic.management.runtime.JTARuntimeMBean</p> <p><i>Attribute:</i> TransactionRolledBackTimeoutTotalCount</p>	

Table 469-1

Attribute Label	Description	Value Constraints
Resource Rollbacks	<p>The number of transactions that were rolled back due to a resource error.</p> <p><i>MBean:</i> weblogic.management.runtime.JTARuntimeMBean</p> <p><i>Attribute:</i> TransactionRolledBackResourceTotalCount</p>	
Application Rollbacks	<p>The number of transactions that were rolled back due to an application error.</p> <p><i>MBean:</i> weblogic.management.runtime.JTARuntimeMBean</p> <p><i>Attribute:</i> TransactionRolledBackAppTotalCount</p>	
System Rollbacks	<p>The number of transactions that were rolled back due to an internal system error.</p> <p><i>MBean:</i> weblogic.management.runtime.JTARuntimeMBean</p> <p><i>Attribute:</i> TransactionRolledBackSystemTotalCount</p>	
Total Heuristics	<p>The number of transactions that completed with a heuristic status.</p> <p><i>MBean:</i> weblogic.management.runtime.JTARuntimeMBean</p> <p><i>Attribute:</i> TransactionHeuristicsTotalCount</p>	
Total Transactions Abandoned	<p>The number of transaction that were abandoned.</p> <p><i>MBean:</i> weblogic.management.runtime.JTARuntimeMBean</p> <p><i>Attribute:</i> TransactionAbandonedTotalCount</p>	

Table 469-1

Attribute Label	Description	Value Constraints
Average Commit Time	<p>The average amount of time (in milliseconds) it takes this server to commit a transaction.</p> <p><i>MBean:</i> weblogic.management.runtime.JTARuntimeMBean</p> <p><i>Attribute:</i> AverageCommitTime</p>	<i>Units:</i> milliseconds
Active Transactions	<p>The total number of active transactions on the server.</p> <p><i>MBean:</i> weblogic.management.runtime.JTARuntimeMBean</p> <p><i>Attribute:</i> ActiveTransactionsTotalCount</p>	

Server --> Monitoring --> JMS

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab provides statistics on JMS servers and connections on your server. This page also provides links to the tables of active JMS connections, active JMS servers, and active pooled JMS connections, which monitor such attributes as total current sessions.

Totals for JMS server statistics are displayed on the page. You can also click the monitoring text links to monitor all active JMS servers, connections, and pooled JMS connections, which are session pools used by EJBs and servlets that use a *resource-reference* element in their EJB deployment descriptor to define their JMS connection factories.

Tasks

“JMS Connection Runtime” on page 160-1

“Active JMS Servers” on page 197-1

[“JMS Pooled Connections” on page 185-1](#)

Related Topics

[“Monitoring a Welogic Server Domain”](#) in the *Administration Guide*

Attributes

Table 470-1

Attribute Label	Description	Value Constraints
Current Connections	The current number of connections to this WebLogic Server. <i>MBean:</i> weblogic.management.runtime.JMSRuntimeMBean <i>Attribute:</i> ConnectionsCurrentCount	
Connections High	The peak number of connections to this WebLogic Server since the last reset. <i>MBean:</i> weblogic.management.runtime.JMSRuntimeMBean <i>Attribute:</i> ConnectionsHighCount	
Total Connections	The total number of connections made to this WebLogic Server since the last reset. <i>MBean:</i> weblogic.management.runtime.JMSRuntimeMBean <i>Attribute:</i> ConnectionsTotalCount	
Current JMS Servers	The current number of JMS servers that are deployed on this WebLogic Server instance. <i>MBean:</i> weblogic.management.runtime.JMSRuntimeMBean <i>Attribute:</i> JMSServersCurrentCount	
Servers High	The peak number of JMS servers that were deployed on this WebLogic Server instance since the server was started. <i>MBean:</i> weblogic.management.runtime.JMSRuntimeMBean <i>Attribute:</i> JMSServersHighCount	

Table 470-1

Attribute Label	Description	Value Constraints
Servers Total	<p>The number of JMS servers that were deployed on this WebLogic Server instance since the server was started.</p> <p><i>MBean:</i> weblogic.management.runtime.JMSRuntimeMBean</p> <p><i>Attribute:</i> JMSServersTotalCount</p>	



Server --> Monitoring --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This page displays the current state of the server, and provides links for monitoring active execute queues, open connections, and open sockets.

Tasks

“Monitoring a Server” on page 495-20

Related Topics

["Understanding Server State"](#)

["Monitoring Servers"](#)

Attributes

Table 471-1

Attribute Label	Description	Value Constraints
State	Returns current state of the server. <i>MBean:</i> weblogic.management.runtime.ServerRuntimeMBean <i>Attribute:</i> State	

Table 471-1

Attribute Label	Description	Value Constraints
Activation Time	<i>MBean:</i> weblogic.management.runtime.ServerRuntimeMBean <i>Attribute:</i> ActivationTime	

Advanced Attributes

Table 471-2

Attribute Label	Description	Value Constraints
Weblogic Version	The version of the server <i>MBean:</i> weblogic.management.runtime.ServerRuntimeMBean <i>Attribute:</i> WeblogicVersion	
JDK Vendor	The vendor of the JVM.	
JDK Version	The Java version of the JVM.	
Operating System	The operating system on which the JVM is running.	
OS Version	The operating-system version on which the JVM is running.	

Server --> Monitoring --> JRockit

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab is available only when you run a server with the JRockit Virtual Machine (VM). It displays runtime data about the JRockit VM that is running the current WebLogic Server instance. It also displays information about the memory and processors on the computer that is hosting the VM.

To view additional data about the VM, such as how long it spends in a specific method, use the JRockit Management Console. You can also use the JRockit Management Console to send email notifications when specific VM events occur.

You cannot change the VM's operating parameters while the VM is active. Instead, use the startup options that are described in the JRockit documentation.

Tasks

For information on specifying Java options, refer to “Specifying Java Options for a WebLogic Server Instance” on page 497-25.

Related Topics

[JRockit Java Virtual Machine User Guide](#)

Attributes

Table 472-1

Attribute Label	Description	Value Constraints
Used Heap	<p>Indicates the amount (in bytes) of Java heap memory that is currently being used by the Virtual Machine.</p> <p><i>MBean:</i> weblogic.management.runtime.JRockitRuntimeMBean</p> <p><i>Attribute:</i> UsedHeap</p>	
Used Physical Memory	<p>Indicates the amount (in bytes) of physical memory that is currently being used on the host computer.</p> <p>This value reports the memory that is being used by all processes on the computer, not just by the Virtual Machine.</p> <p><i>MBean:</i> weblogic.management.runtime.JRockitRuntimeMBean</p> <p><i>Attribute:</i> UsedPhysicalMemory</p>	

Table 472-1

Attribute Label	Description	Value Constraints
Total Nursery Size	<p>Indicates the amount (in bytes) of memory that is currently allocated to the nursery.</p> <p>The nursery is the area of the Java heap that the VM allocates to most objects. Instead of garbage collecting the entire heap, generational garbage collectors focus on the nursery. Because most objects die young, most of the time it is sufficient to garbage collect only the nursery and not the entire heap.</p> <p>If you are not using a generational garbage collector, the nursery size is 0.</p> <p><i>MBean:</i> weblogic.management.runtime.JRockitRuntimeMBean</p> <p><i>Attribute:</i> TotalNurserySize</p>	
Max Heap Size	<p>Indicates the maximum amount of memory (in bytes) that the Virtual Machine can allocate for its Java heap. This number is fixed at startup time of the VM, typically by the <code>-Mx</code> option.</p>	

Table 472-1

Attribute Label	Description	Value Constraints
Gc Algorithm	<p>Indicates the type of garbage collector (GC) that the Virtual Machine is using. JRockit provides the following types of GCs:</p> <ul style="list-style-type: none">■ Generational Copying, which is suitable for testing applications on a desktop machine with a small (less than 128 MB) heap.■ Single Spaced Concurrent, which reduces or eliminates pauses in the VM that are due to garbage collection. Because it trades memory throughput for reduced pause time, you generally need a larger heap size than with other GC types. If your ordinary Java threads create more garbage than this GC can collect, the VM will pause while the Java threads wait for the garbage collection to finish.■ Generational Concurrent, which creates a "nursery" space within the heap. New objects are created within the nursery. When the nursery is full, JRockit "stops-the-world," removes the dead objects from the nursery, and moves live objects to a different space within the heap. Another thread runs in the background to remove dead objects from the non-nursery space. This GC type has a higher memory throughput than a single spaced concurrent GC.■ Parallel, which allocates all objects to a single spaced heap. When the heap is full, all Java threads are stopped and every CPU is used to perform a complete garbage collection of the entire heap. This behavior causes longer pause times than for the concurrent collectors but maximizes memory throughput.	

Table 472-1

Attribute Label	Description	Value Constraints
Total Garbage Collection Count	Indicates the number of garbage collection runs that have occurred since the Virtual Machine was started. <i>MBean:</i> weblogic.management.runtime.JRockitRuntimeMBean <i>Attribute:</i> TotalGarbageCollectionCount	
Last GcEnd	Indicates the time at which the last garbage collection run ended. <i>MBean:</i> weblogic.management.runtime.JRockitRuntimeMBean <i>Attribute:</i> LastGcEnd	
Last GcStart	Indicates the time at which the last garbage collection run started. <i>MBean:</i> weblogic.management.runtime.JRockitRuntimeMBean <i>Attribute:</i> LastGcStart	
Total GcTime	Indicates the number of milliseconds that the Virtual Machine has spent on all garbage collection runs since the VM was started. <i>MBean:</i> weblogic.management.runtime.JRockitRuntimeMBean <i>Attribute:</i> TotalGcTime	

Table 472-1

Attribute Label	Description	Value Constraints
GCHandles Compaction	<p>Indicates whether the VM's garbage collector compacts the Java heap. Usually the heap is scattered throughout available memory. A garbage collector that compacts the heap defragments the memory space in addition to deleting unused objects.</p> <p><i>MBean:</i> <code>weblogic.management.runtime.JRockitRuntimeMBean</code></p> <p><i>Attribute:</i> <code>GCHandlesCompaction</code></p>	<p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false
Concurrent	<p>Indicates whether the VM's garbage collector runs in a separate Java thread concurrently with other Java threads.</p> <p><i>MBean:</i> <code>weblogic.management.runtime.JRockitRuntimeMBean</code></p> <p><i>Attribute:</i> <code>Concurrent</code></p>	<p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false
Generational	<p>Indicates whether the VM's garbage collector uses a nursery space.</p> <p>A nursery is the area of the Java heap that the VM allocates to most objects. Instead of garbage collecting the entire heap, generational garbage collectors focus on the nursery. Because most objects die young, most of the time it is sufficient to garbage collect only the nursery and not the entire heap.</p> <p><i>MBean:</i> <code>weblogic.management.runtime.JRockitRuntimeMBean</code></p> <p><i>Attribute:</i> <code>Generational</code></p>	<p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false

Table 472-1

Attribute Label	Description	Value Constraints
Incremental	<p>Indicates whether the VM's garbage collector collects (increments) garbage as it scans the memory space and dumps the garbage at the end of its cycle.</p> <p>With a non-incremental garbage collector, garbage is dumped as soon as it is encountered.</p> <p><i>MBean:</i> <code>weblogic.management.runtime.JRockitRuntimeMBean</code></p> <p><i>Attribute:</i> <code>Incremental</code></p>	<p><i>Valid values:</i></p> <ul style="list-style-type: none">■ <code>true</code>■ <code>false</code>
Parallel	<p>Indicates whether the VM's garbage collector is able to run in parallel on multiple processors if multiple processors are available.</p> <p><i>MBean:</i> <code>weblogic.management.runtime.JRockitRuntimeMBean</code></p> <p><i>Attribute:</i> <code>Parallel</code></p>	<p><i>Valid values:</i></p> <ul style="list-style-type: none">■ <code>true</code>■ <code>false</code>
Number Of Processors	<p>Displays the number of processors on the Virtual Machine's host computer. If this is not a Symetric Multi Processor (SMP) system, the value will be 1.</p> <p><i>MBean:</i> <code>weblogic.management.runtime.JRockitRuntimeMBean</code></p> <p><i>Attribute:</i> <code>NumberOfProcessors</code></p>	

Table 472-1

Attribute Label	Description	Value Constraints
All Processors Average Load	<p>Displays a snapshot of the average load of all processors in the host computer. If the computer has only one processor, this value is the same as JVM Processor Load.</p> <p>The value is returned as a double, where 1.0 represents 100% load (no idle time) and 0.0 represents 0% load (pure idle time).</p> <p><i>MBean:</i> <code>weblogic.management.runtime.JRockitRuntimeMBean</code></p> <p><i>Attribute:</i> <code>AllProcessorsAvgLoad</code></p>	
JVM Processor Load	<p>Displays a snapshot of the load that the Virtual Machine is placing on all processors in the host computer. If the host contains multiple processors, the value represents a snapshot of the average load.</p> <p>The value is returned as a double, where 1.0 represents 100% load (no idle time) and 0.0 represents 0% load (pure idle time).</p> <p><i>MBean:</i> <code>weblogic.management.runtime.JRockitRuntimeMBean</code></p> <p><i>Attribute:</i> <code>JVMProcessorLoad</code></p>	
Total Number Of Threads	<p>Indicates the number of Java threads (daemon and non-daemon) that are currently running in the Virtual Machine across all processors.</p> <p><i>MBean:</i> <code>weblogic.management.runtime.JRockitRuntimeMBean</code></p> <p><i>Attribute:</i> <code>TotalNumberOfThreads</code></p>	

Table 472-1

Attribute Label	Description	Value Constraints
Number Of Daemon Threads	Indicates the number of daemon Java threads currently running in the Virtual Machine across all processors. <i>MBean:</i> weblogic.management.runtime.JRockitRuntimeMBean <i>Attribute:</i> NumberOfDaemonThreads	
Uptime	Indicates the number of milliseconds that the Virtual Machine has been running. <i>MBean:</i> weblogic.management.runtime.JRockitRuntimeMBean <i>Attribute:</i> Uptime	



Server --> Logging --> Server

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The Server—>Logging—>Server tab configures a server's local message log. Use this tab to specify the name of the log file and any log file rotation criteria.

In addition, you can use this tab to specify the types of messages that the server sends to standard out.

The server message log does not contain HTTP requests, JDBC messages, or JTA transaction messages. Instead, you can configure separate log files for those requests and messages.

In addition to maintaining its local message log, by default, each server forwards all messages of severity ERROR and higher to the domain log.

Debug Messages

If you create applications to run on WebLogic Server, you can configure your applications to generate messages of severity DEBUG. These messages are never forwarded to the domain log and are intended to contain detailed information about the operation of an application or the server.

Tasks

“Viewing Server Logs” on page 253-9

“Changing the Name and Location of the Server Log File” on page 253-25

“Specifying Which Messages a Server Sends to Standard Out” on page 253-20

“Rotating Log Files” on page 253-16

“Specifying the Messages That a Server Forwards to the Domain Log” on page 81-1

“Viewing the Domain Log” on page 253-12

Related Topics

“Overview of WebLogic Server Log Messages and Log Files” on page 253-2

[Writing Debug Messages](#)

Attributes

Table 473-1

Attribute Label	Description	Value Constraints
Server File Name	<p>The name of the file that stores current log messages. If the pathname is not absolute, the path is relative to the server's root directory. For more information, refer to "A Server's Root Directory."</p> <p>If the name does not include a path element, then pathname is <code>./SERVER_NAME</code></p> <p>If neither file name nor pathname is specified, then the name is <code>./SERVER_NAME/SERVER_NAME.log</code></p> <p>To include a time or date stamp in the file name when the log file is rotated, add <code>java.text.SimpleDateFormat</code> variables. Surround each variable with percentage (%) characters.</p> <p>For example,</p> <p><code>myserver_%yyyy%_MM%_dd%_hh _%mm%.log</code></p> <p>If you do not include a time and date stamp, the rotated log files are numbered in order of creation <code>filenamennnnn</code>, where <i>filename</i> is the name configured for the log file.</p> <p><i>MBean</i>: <code>weblogic.management.configuration.LogMBean</code></p> <p><i>Attribute</i>: <code>FileName</code></p>	<p><i>Default</i>: <code>weblogic.log</code></p> <p><i>Configurable</i>: yes</p>

Table 473-1

Attribute Label	Description	Value Constraints
Log to Stdout	<p>Enables the server to send messages to standard out in addition to the log file. Use StdoutDebugEnabled and StdoutSeverityLevel to determine the type of messages that the server sends to standard out.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> StdoutEnabled</p>	<p><i>Default:</i> true</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false <p><i>Dynamic:</i> yes</p>
Debug to Stdout	<p>Determines whether the server sends messages of the DEBUG severity to standard out in addition to the log file. You must enable Log to Standard Out for this property to be relevant.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> StdoutDebugEnabled</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false <p><i>Dynamic:</i> yes</p>

Table 473-1

Attribute Label	Description	Value Constraints
Stdout Severity Threshold	<p>The minimum severity of a message that the server sends to standard out. You must enable <code>Log to Standard Out</code> for this value to be relevant.</p> <p>The ascending order of severities is as follows:</p> <ul style="list-style-type: none">■ <code>INFO (64)</code>. Used for reporting normal operations.■ <code>WARNING (32)</code>. A suspicious operation or configuration has occurred but it may not have an impact on normal operation.■ <code>ERROR (16)</code>. A user error has occurred. The system or application is able to handle the error with no interruption, and limited degradation, of service.■ <code>NOTICE (8)</code>. An <code>INFO</code> or <code>WARNING</code>-level message that is particularly important for monitoring the server. Only WebLogic Server subsystems write messages of this severity type.■ <code>CRITICAL (4)</code>. 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.■ <code>ALERT (2)</code>. A particular 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.■ <code>EMERGENCY (1)</code>. The server is in an unusable state. This severity indicates a severe system failure or panic. <p><i>MBean:</i> <code>weblogic.management.configuration.ServerMBean</code></p> <p><i>Attribute:</i> <code>StdoutSeverityLevel</code></p>	<p><i>Default:</i> <code>WARNING</code></p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ <code>INFO</code>■ <code>WARNING</code>■ <code>ERROR</code>■ <code>NOTICE</code>■ <code>CRITICAL</code>■ <code>ALERT</code>■ <code>EMERGENCY</code> <p><i>Dynamic:</i> yes</p>

Table 473-1

Attribute Label	Description	Value Constraints
Rotation Type	<p>Criteria for moving old log messages to a separate file:</p> <ul style="list-style-type: none">■ 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 <code>FileMinSize</code>, the server renames the file as <code>FileName.n</code>.■ By Time. At each time interval that you specify in <code>TimeSpan</code>, the server renames the file as <code>FileName.n</code>. <p>After the server renames a file, subsequent messages accumulate in a new file with the name that you specified in <code>FileName</code>.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.LogMBean</code></p> <p><i>Attribute:</i> <code>RotationType</code></p>	<p><i>Default:</i> By Size</p> <p><i>Dynamic:</i> yes</p>
Minimum File Size	<p>The size (1 - 65535 kilobytes) that triggers the server to move log messages to a separate file. After the log file reaches the specified minimum size, the next time the server checks the file size, it will rename the current log file as <code>FileName.n</code> and create a new one to store subsequent messages. (This field is relevant only if you set Rotation Type to By Size.)</p> <p><i>MBean:</i> <code>weblogic.management.configuration.LogMBean</code></p> <p><i>Attribute:</i> <code>FileMinSize</code></p>	<p><i>Units:</i> kilobytes</p> <p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 65535</p> <p><i>Default for Development Mode:</i> 500</p> <p><i>Default for Production Mode:</i> 5000</p> <p><i>Configurable:</i> yes</p> <p><i>Dynamic:</i> yes</p>

Table 473-1

Attribute Label	Description	Value Constraints
Rotation Time	<p>Determines the start time (hour and minute) for a time-based rotation sequence.</p> <p>At the time that this value specifies, the server renames the current log file as <i>FileName.n</i>. Thereafter, the server renames the log file at an interval that you specify in <i>FileTimeSpan</i>.</p> <p>Use the following format: <i>hh:mm</i>, where <i>hh</i> is the hour in a 24-hour format and <i>mm</i> is the minute.</p> <p>If the time that you specify has already past, then the server starts its file rotation immediately.</p> <p><i>MBean</i>: <code>weblogic.management.configuration.LogMBean</code></p> <p><i>Attribute</i>: <code>RotationTime</code></p>	<p><i>Default</i>: 00:00 (12:00 AM)</p> <p><i>Configurable</i>: yes</p> <p><i>Dynamic</i>: yes</p>
File Time Span	<p>The interval (in hours) at which the server saves old log messages to another file. This value is relevant only you use the time-based rotation type.</p> <p><i>MBean</i>: <code>weblogic.management.configuration.LogMBean</code></p> <p><i>Attribute</i>: <code>FileTimeSpan</code></p>	<p><i>Units</i>: hours</p> <p><i>Minimum</i>: 1</p> <p><i>Default</i>: 24</p> <p><i>Dynamic</i>: yes</p>
Limit Number of Retained Log Files	<p>Specifies whether the number of files that this WebLogic Server creates to store old messages should be limited. After the server reaches this limit, it overwrites the oldest file.</p> <p><i>MBean</i>: <code>weblogic.management.configuration.LogMBean</code></p> <p><i>Attribute</i>: <code>NumberOfFilesLimited</code></p>	<p><i>Default</i>: false</p>

Table 473-1

Attribute Label	Description	Value Constraints
Log Files to Retain	<p>The maximum number of log files that this WebLogic Server creates when it rotates the log. (This field is relevant only if you check the Number of Files Limited box.)</p> <p><i>MBean:</i> weblogic.management.configuration.LogMBean</p> <p><i>Attribute:</i> FileCount</p>	<p><i>Default:</i> 7</p>
Instrument Stack Traces	<p>Determines whether the server returns stack traces for RMI calls that generate exceptions.</p> <p>With this attribute enabled, if a client issues an RMI call to a server subsystem or to a module running within the server, and if the subsystem or module generates an exception that includes a stack trace, the server will return the exception as well as the stack trace. With this attribute disabled, the server will return exceptions without the stack trace details.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> InstrumentStackTraceEnabled</p>	<p><i>Default:</i> true</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false <p><i>Dynamic:</i> yes</p>

Server --> Deployments --> Web Modules

[Tasks](#) [Related Topics](#)

Overview

This page lists the Web Applications that have been configured for deployment to this server. Select the name of the Web Application to view the current deployment status, or to deploy, redeploy, or stop the application.

Tasks

[“Deploying a New Web Application” on page 559-2](#)

[“Stopping Deployed Web Applications” on page 559-4](#)

Related Topics

[Packaging and Deploying WebLogic Server Applications](#)



Server --> Deployments --> EJB Modules

[Tasks](#) [Related Topics](#)

Overview

This page lists the EJB modules that have been configured for deployment to this server. Select the name of the EJB module to view the current deployment status, or to deploy, redeploy, or stop the EJB.

Tasks

[“Configuring an EJB Module” on page 103-3](#)

[“Stopping or Redeploying an EJB Module” on page 103-7](#)

Related Topics

[Packaging and Deploying WebLogic Server Applications](#)



Server --> Deployments --> Connector

[Tasks](#) [Related Topics](#)

Overview

This page lists the Connector modules that have been configured for deployment to this server. Select the name of the Connector module to view the current deployment status, or to deploy, redeploy, or stop the module.

Tasks

[“Deploying New Connector \(Resource Adapter\) Modules” on page 57-2](#)

[“Stopping a Deployed Connectors” on page 57-4](#)

Related Topics

[Packaging and Deploying WebLogic Server Applications](#)



Server --> Deployments --> Applications

[Tasks](#) [Related Topics](#)

Overview

This page lists J2EE Applications and Web Services that have been configured for deployment to this server. Select the name of a deployed application or Web Service to view the current deployment status, or to deploy, redeploy, or stop the application.

Tasks

[“Configuring and Deploying a New Enterprise Application or Web Service” on page 26-2](#)

[“Deploying a New Web Application” on page 559-2](#)

[“Stopping Deployed Enterprise Applications” on page 26-3](#)

Related Topics

[Packaging and Deploying WebLogic Server Applications](#)



Server --> Control --> Remote Start Output

[Tasks](#) [Related Topics](#)

Overview

Use this page to view the standard output, standard error, and Node Manager output for a Managed Server that was started using the Node Manager application. You cannot use this page to view output for servers that were started without using Node Manager (for example, servers started at the command line).

Select View Server Output to see the standard output of a Managed Server that was started using Node Manager. This is the same content that is normally displayed on the command line when you start a server using scripts.

Select View Server Error to see any error messages the server generated when Node Manager attempted to start it.

Select View Node Manager Output to see Node Manager status messages written in response to Managed Server actions, such as starting or automatically shutting down the server.

Tasks

Related Topics

["Configuring Node Manager"](#)

["Starting and Stopping Node Manager"](#)

["Troubleshooting Node Manager"](#)



Server --> Control --> JTA Migration Config.

[Tasks](#) [Related Topics](#)

Overview

On the ~~Server~~–~~Control~~–JTA Migration Config tab in the Administration Console, you can limit the choices of the servers to use as a Transaction Recovery Service backup for a server in a cluster. For example, only certain servers in your cluster may have access to the transaction log files for a server. When you select target servers in the ~~Server~~–~~Control~~–JTA Migration Config tab, you can limit the list of destination servers available on the ~~Server~~–~~Control~~–Migrate JTA tab.

The ~~Server~~–~~Control~~–JTA Migration Config tab shows a list of Available servers (servers in the cluster) and a list of Chosen server. If you do not specify a list of chosen servers, you can migrate the Transaction Recovery Service to any server in the cluster.

Notes: You can only migrate the Transaction Recovery Service from one server in a cluster to another server in the same cluster that can access the transaction log of the failed server.

If you specify a list of chosen servers, you *must* include the current server in the list or you will not be able to migrate the service back to the server after transactions are recovered.

Tasks

[“Constraining the Servers to Which the Transaction Recovery Service can Migrate” on page 237-22](#)

[“Preparing to Migrate the Transaction Recovery Service” on page 237-21](#)

[“Migrating the Transaction Recovery Service to a Server in the Same Cluster” on page 237-21](#)

[“Viewing Current Owner of the Transaction Recovery Service” on page 237-23](#)

[“Manually Migrating the Transaction Recovery Service Back to the Original Server”](#)
on page 237-23

Related Topics

[“Transaction Recovery After a Server Fails”](#) on page 237-16

[“Setting the Transaction Log File Location \(Prefix\)”](#) on page 237-11

[“Migrating a Pinned Service to a Target Server Instance”](#)

[“Migrating When the Currently Active Host is Unavailable”](#)

Server --> Control --> JMS Migrate

[Tasks](#) [Related Topics](#)

Overview

Use this page to migrate a JMS server from a failed clustered server to another server in the cluster.

Based on the list of candidate servers (in the cluster) selected on the [Server —>Control —> JMS Migration Config.](#) page, use the Destination field to select a destination server to take over for the server currently hosting the JMS server. Select the server where you want to migrate the JMS server, and then click the Migrate button.

Notes: If your JMS application uses persistent messaging, you can only migrate the JMS server from one server in a cluster to another server in the same cluster that can access the persistent JMS store of the failed server.

If your JMS applications use JTA transactions, you must also migrate the Transaction Recovery Service to another server. For more information see, [Transaction Recovery After a Server Fails](#).

Tasks

[“JMS Server --> Target and Deploy” on page 196-1](#)

[“Server --> Control --> JMS Migration Config.” on page 481-1](#)

[“Server --> Control --> JTA Migration Config.” on page 479-1](#)

[“Server --> Control --> JTA Migrate” on page 482-1](#)

Related Topics

[“Migrating a Pinned Service to a Target Server Instance”](#)

[“Migrating When the Currently Active Host is Unavailable”](#)

[“Configuring JMS Migratable Targets”](#)

[“Transaction Recovery After a Server Fails”](#)

Attributes

Cluster—The name of the cluster to which the server belongs.

Current Server—The server that currently owns the JMS server. This may not necessarily be server that you are currently using to view the console (the server name at the top of the console page). In other words, if the JMS server was migrated to another server, that server is listed here. If the JMS server was not migrated, the current server is the server that is running the console.

Destination Server—A list of servers in the cluster. Select the server to which you want to migrate the JMS server. This list shows all servers in the cluster or a list of servers that you specify on the [Server →Control →JMS Migration Config.](#) page.

Server --> Control --> JMS Migration Config.

[Tasks](#) [Related Topics](#)

Overview

This page allows you to limit the choices of the servers in a cluster to use as a backup for the server currently hosting the JMS server. For example, only certain servers in your cluster may have access to the persistent store for a JMS server. When you select candidate servers on this page, you limit the list of destination servers available on the [Server → Control → JMS Migrate](#) tab.

The Available box shows a list of candidate servers (servers in the cluster) that can be added to the list of Chosen migratable target servers. If you do not specify a list of Chosen servers, you can migrate the JMS server to any server in the cluster on the [Server → Control → JMS Migrate](#) tab. If you do specify a list of Chosen servers, you must include the current server in the list or you will not be able to migrate the JMS server back to the server, if necessary.

Note: For implementations that use persistent messaging, make sure that the persistent JMS store is configured such that all the candidate servers in the Chosen list share access to the persistent store.

Tasks

[“JMS Server --> Target and Deploy” on page 196-1](#)

[“Server --> Control --> JMS Migrate” on page 480-1](#)

Related Topics

[“Migrating a Pinned Service to a Target Server Instance”](#)

[“Migrating When the Currently Active Host is Unavailable”](#)

[“Configuring JMS Migratable Targets”](#)

Server --> Control --> JTA Migrate

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

On the ~~Server~~—~~Control~~—JTA Migrate tab in the Administration Console, you can migrate the Transaction Recovery Service from a failed clustered server to another server in the cluster.

Note: You can only migrate the Transaction Recovery Service from one server in a cluster to another server in the same cluster that can access the transaction log of the failed server.

Tasks

[“Preparing to Migrate the Transaction Recovery Service” on page 237-21](#)

[“Migrating the Transaction Recovery Service to a Server in the Same Cluster” on page 237-21](#)

[“Constraining the Servers to Which the Transaction Recovery Service can Migrate” on page 237-22](#)

[“Viewing Current Owner of the Transaction Recovery Service” on page 237-23](#)

[“Manually Migrating the Transaction Recovery Service Back to the Original Server” on page 237-23](#)

Related Topics

[“Transaction Recovery After a Server Fails” on page 237-16](#)

[“Setting the Transaction Log File Location \(Prefix\)” on page 237-11](#)

[“Migrating a Pinned Service to a Target Server Instance”](#)

[“Migrating When the Currently Active Host is Unavailable”](#)

Attributes

Cluster—The name of the cluster to which the server belongs.

Current Server—The server that currently owns the Transaction Recovery Service for the selected server. If the Transaction Recovery Service was migrated to another server, that server is listed here. If the service was not migrated, the current server is the server for which you are viewing attributes (the server name at the top of the console page).

Destination Server—A list of servers in the cluster. Select the server to which you want to migrate the Transaction Recovery Service. This list shows all servers in the cluster or a list of servers that you specify. See [“Constraining the Servers to Which the Transaction Recovery Service can Migrate”](#) on page 14-3.

Note: You can only migrate the Transaction Recovery Service from one server in a cluster to another server in the same cluster that can access the transaction log of the failed server.

Servers --> Protocols --> Channels

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to configure a network channel for the server instance. A network channel allows you to segregate network traffic by Network Interface Card (NIC) and to listen for requests on multiple ports.

Tasks

“Configuring a Custom Network Channel for a Non-Clustered Server” on page 495-15

Related Topics

["Understanding Network Channels"](#)

["Configuring Network Channels"](#)

Attributes

Table 483-1

Attribute Label	Description	Value Constraints
Name	The name of this Network Channel. <i>MBean:</i> weblogic.management.configuration.NetworkAccessPointMBean <i>Attribute:</i> Name	<i>Default:</i> "<unknown>"
Protocol	The protocol supported by this Network Channel. <i>MBean:</i> weblogic.management.configuration.NetworkAccessPointMBean <i>Attribute:</i> Protocol	<i>Default:</i> "t3" <i>Valid values:</i> <ul style="list-style-type: none">■ "t3"■ "iiop"■ "com"■ "http"■ "t3s"■ "iiops"■ "https"
Listen Address	The IP address or DNS name associated with the Network Channel.	Default value is the listen address for the server instance.
Listen Port	The Listen Port associated with the Network Channel.	Default value is the standard listen port for the server instance.

Table 483-1

Attribute Label	Description	Value Constraints
External Listen Address	<p>The external address for the current server, which will be sent to clients. This will be required for the configurations in which need to cross a firewall doing Network Address Translation. This property supersedes <code>ExternalDNSName</code>. A value of null indicates that this value is inherited from the server.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.NetworkAccessPointMBean</code></p> <p><i>Attribute:</i> <code>PublicAddress</code></p>	<i>Default:</i> null
External Listen Port	<p>The external listen port for the channel. -1 implies that this value is inherited from the internal listen port.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.NetworkAccessPointMBean</code></p> <p><i>Attribute:</i> <code>PublicPort</code></p>	<i>Default:</i> -1

Advanced Attributes

Table 483-2

Attribute Label	Description	Value Constraints
Outbound Enabled	Whether or not new server-to-server connections may consider this channel when initiating a request. <i>MBean:</i> weblogic.management.configuration.NetworkAccessPointMBean <i>Attribute:</i> OutboundEnabled	<i>Default:</i> true, unless Protocol is COM, in which case default is false. <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false If Protocol is COM, this attribute cannot be changed from default of false.
Channel Weight	A weight to give this channel when creating server-to-server connections. <i>MBean:</i> weblogic.management.configuration.NetworkAccessPointMBean <i>Attribute:</i> ChannelWeight	<i>Minimum:</i> 1 <i>Maximum:</i> 100 <i>Default:</i> 50
Cluster Address	This channel's cluster address. If this is not set, the public address is used and if this is not set then the cluster address from the cluster configuration is used in its place. <i>MBean:</i> weblogic.management.configuration.NetworkAccessPointMBean <i>Attribute:</i> ClusterAddress	<i>Default:</i> null
Accept Backlog	Allowed backlog of connection requests on the listen port. -1 implies that this value is inherited from the channel. <i>MBean:</i> weblogic.management.configuration.NetworkAccessPointMBean <i>Attribute:</i> AcceptBacklog	<i>Minimum:</i> -1 <i>Default:</i> -1

Table 483-2

Attribute Label	Description	Value Constraints
Http Enabled For This Protocol	<p>Whether or not this port will accept HTTP requests. HTTP is generally required by binary protocols for downloading stubs and other resources.</p> <p><i>MBean:</i> weblogic.management.configuration.NetworkAccessPointMBean</p> <p><i>Attribute:</i> HttpEnabledForThisProtocol</p>	<p><i>Default:</i> true, except when Protocol is COM, in which case default is false.</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false <p>If Protocol is COM, this attribute cannot be changed from default of false.</p> <p>If Protocol is HTTP or HTTPS, this attribute cannot be changed from default of true.</p>
Login Timeout	<p>The login timeout for the server, in milliseconds. This value must be equal to or greater than 0. -1 implies that this value is inherited from the channel.</p> <p><i>MBean:</i> weblogic.management.configuration.NetworkAccessPointMBean</p> <p><i>Attribute:</i> LoginTimeoutMillis</p>	<p><i>Units:</i> milliseconds</p> <p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 100000</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>
Complete Message Timeout	<p>The maximum number of seconds spent waiting for a complete message to be received. This attribute helps guard against denial of service attacks in which a caller indicates that they will be sending a message of a certain size which they never finish sending. -1 implies that this value is inherited from the channel.</p> <p><i>MBean:</i> weblogic.management.configuration.NetworkAccessPointMBean</p> <p><i>Attribute:</i> CompleteMessageTimeout</p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 480</p> <p><i>Default:</i> -1</p> <p><i>Configurable:</i> yes</p> <p><i>Dynamic:</i> yes</p>

Table 483-2

Attribute Label	Description	Value Constraints
Idle Connection Timeout	<p>The maximum number of seconds an connection is allowed to be idle before it is closed by the server. This attribute helps guard against server deadlock through too many open connections. -1 implies that this value is inherited from the default channel.</p> <p><i>MBean:</i> weblogic.management.configuration.NetworkAccessPointMBean</p> <p><i>Attribute:</i> IdleConnectionTimeout</p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> -1</p> <p><i>Default:</i> -1</p> <p><i>Configurable:</i> yes</p> <p><i>Dynamic:</i> yes</p>
Tunneling Enabled	<p>http-tunneled clients enabled.</p> <p><i>MBean:</i> weblogic.management.configuration.NetworkAccessPointMBean</p> <p><i>Attribute:</i> TunnelingEnabled</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false <p>If Protocol is COM this attribute can not be changed from default of false.</p>
Tunneling Client Ping	<p>Interval (in seconds) at which to ping an http-tunneled client to see if its still alive. -1 implies that this value is inherited from the channel.</p> <p>This attributed is only has an effect if Tunneling Enabled is true.</p> <p><i>MBean:</i> weblogic.management.configuration.NetworkAccessPointMBean</p> <p><i>Attribute:</i> TunnelingClientPingSecs</p>	<p><i>Units:</i> seconds</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>

Table 483-2

Attribute Label	Description	Value Constraints
Tunneling Client Timeout	<p>Duration (in seconds) after which a missing http-tunneled client is considered dead. -1 implies that this value is inherited from the channel.</p> <p>This value only has an effect if Tunneling Enabled is true.</p> <p><i>MBean:</i> weblogic.management.configuration.NetworkAccessPointMBean</p> <p><i>Attribute:</i> TunnelingClientTimeoutSecs</p>	<p><i>Units:</i> seconds</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>
Maximum Message Size	<p>Specify the maximum message size allowable in a message header. This attribute attempts to prevent a denial of service attack whereby a caller attempts to force the server to allocate more memory than is available thereby keeping the server from responding quickly to other requests.</p> <p><i>MBean:</i> weblogic.management.configuration.NetworkAccessPointMBean</p> <p><i>Attribute:</i> MaxMessageSize</p>	<p><i>Units:</i> bytes</p> <p><i>Default:</i> 10000000</p> <p><i>Configurable:</i> yes</p> <p><i>Dynamic:</i> yes</p>



Server --> Protocols --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The Protocols --> General tab specifies defaults for all protocols that can be used to connect to a server. Any network channel that you create for this server can override the values on this tab.

Tasks

[“Configuring the T3 Protocol” on page 495-8](#)

[“Configuring the HTTP Protocol” on page 495-7](#)

[“Enabling and Configuring the IIOP Protocol” on page 495-9](#)

[“Enabling and Configuring jCOM” on page 495-10](#)

Related Topics

[Understanding the Default Network Configuration](#)

[Understanding Network Channels](#)

Attributes

Advanced Attributes

Table 484-1

Attribute Label	Description	Value Constraints
Complete Message Timeout	<p>The maximum number of seconds that this server waits for a complete message to be received. This attribute helps guard against denial of service attacks in which a caller indicates that it will be sending a message of a certain size which it never finishes sending.</p> <p>If you create network channels for this server, each channel can override this setting.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> CompleteMessageTimeout</p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 480</p> <p><i>Default:</i> 60</p> <p><i>Configurable:</i> yes</p> <p><i>Dynamic:</i> yes</p>
Idle Connection Timeout	<p>The maximum number of seconds that a connection is allowed to be idle before it is closed by the server. This attribute helps guard against server deadlock through too many open connections. The T3 and T3S protocols ignore this attribute.</p> <p>If you create network channels for this server, each channel can override this setting.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> IdleConnectionTimeout</p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> 0</p> <p><i>Default:</i> 65</p> <p><i>Configurable:</i> yes</p> <p><i>Dynamic:</i> yes</p>
Enable Tunneling	<p>Enables tunneling for the T3, T3S, HTTP, HTTPS, IIOP, and IIOPS protocols.</p> <p>If you create network channels for this server, each channel can override this setting.</p> <p><i>MBean:</i> weblogic.management.configuration.ServerMBean</p> <p><i>Attribute:</i> TunnelingEnabled</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false <p><i>Dynamic:</i> yes</p>

Table 484-1

Attribute Label	Description	Value Constraints
Tunneling Client Ping	Interval (in seconds) at which to ping a tunneled client to see if it is still alive. If you create network channels for this server, each channel can override this setting. <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> TunnelingClientPingSecs	<i>Units:</i> seconds <i>Minimum:</i> 1 <i>Default:</i> 45 <i>Dynamic:</i> yes
Tunneling Client Timeout	Duration (in seconds) after which a missing tunneled client is considered dead. If you create network channels for this server, each channel can override this setting. <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> TunnelingClientTimeoutSecs	<i>Units:</i> seconds <i>Minimum:</i> 1 <i>Default:</i> 40 <i>Dynamic:</i> yes
Maximum Message Size	The maximum message size allowable in a message header. This attribute attempts to prevent a denial of service attack whereby a caller attempts to force the server to allocate more memory than is available thereby keeping the server from responding quickly to other requests. If you create network channels for this server, each channel can override this setting. <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> MaxMessageSize	<i>Units:</i> bytes <i>Minimum:</i> 4096 <i>Maximum:</i> 2000000000 <i>Default:</i> 10000000 <i>Configurable:</i> yes <i>Dynamic:</i> yes



Specify Keystore Type

[Tasks](#) [Related Topics](#)

Overview

Use this page to choose a type of keystore (identity and trust) configuration. Identity keystores contain private keys for WebLogic Server. Trust keystores contain certificate authorities that WebLogic Server trusts.

The following options are available:

- **Demo Identity and Demo Trust**—The demonstration identity and trust keystores located in the `BEA_HOME\server\lib` directory and configured by default.
- **Custom Identity and Java Standard Trust**—A keystore you create and the trusted certificate authorities defined in the `cacerts` file in the `JAVA_HOME\jre\lib\security\cacerts` directory.
- **Custom Identity and Custom Trust**—Identity and trust keystores you create.
- **Custom Identity and Command-Line Trust**—An identity keystore you create and command-line arguments that specify the location of the trust keystore.

Tasks

“Configuring Keystores and SSL” on page 428-43

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation

Review SSL Private Key Settings

[Tasks](#) [Related Topics](#)

Overview

Changes to the Identity and Trust keystore configuration may require changes to the default settings for the SSL attributes. This page allows you to change the keystore-related SSL attributes.

Tasks

“Configuring Keystores and SSL” on page 428-43

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation



Configure Keystore Properties

[Tasks](#) [Related Topics](#)

Overview

Use this page to define attributes for the identity (private key) and trust (trusted certificate authorities). You can use demo (keystores provided by WebLogic Server), custom (keystores you create), and Java standard (keystores provided in the JDK).

Tasks

“Configuring Keystores and SSL” on page 428-43

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security page](#) in the WebLogic Server documentation



Restart Your Server

[Tasks](#) [Related Topics](#)

Overview

Any changes made to the default keystore configuration require you to restart the server. Click the Finish button, then restart your server to make the changes take effect.

Tasks

“Configuring Keystores and SSL” on page 428-43

Related Topics

[Introduction to WebLogic Security](#)

[Managing WebLogic Security](#)

[Securing WebLogic Resources](#)

[Programming WebLogic Security](#)

[Developing Security Providers for WebLogic Server](#)

[Securing a Production Environment](#)

The Security topics in the [WebLogic Server 8.1 Upgrade Guide](#)

[Security FAQ](#)

The [Security](#) page in the WebLogic Server documentation



Execute Queue --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Requests to a server instance are placed in an execute queue. Each request is assigned to a thread within the queue that performs the work.

By default, a new server instance is configured with a default execute queue, `weblogic.kernel.default`, that contains 15 threads. In addition, WebLogic Server provides two other pre-configured queues:

`weblogic.admin.HTTP`—Available only on Administration Servers, this queue is reserved for communicating with the Administration Console; you cannot reconfigure it.

`weblogic.admin.RMI`—Both Administration Servers and Managed Servers have this queue; it is reserved for administrative traffic; you cannot reconfigure it.

Unless you configure additional execute queues, and assign applications to them, web applications and RMI objects use `weblogic.kernel.default`.

Use this page to configure a new, user-defined execute queue for use with applications on this server, or to edit an existing execute queue.

Tasks

[“Creating Execute Queues”](#) in *WebLogic Server Performance and Tuning*.

[“Tuning Execute Queues for Overflow Conditions”](#) in *WebLogic Server Performance and Tuning*.

Related Topics

- [“Tuning the Default Execute Threads”](#) in *WebLogic Server Performance and Tuning*.
- [“Using Execute Queues to Control Thread Usage”](#) in *WebLogic Server Performance and Tuning*.
- [“Assigning Applications to Execute Queues”](#) in *WebLogic Server Performance and Tuning*.

Attributes

Table 489-1

Attribute Label	Description	Value Constraints
Name	The name of this execute queue. <i>MBean:</i> weblogic.management.configuration.ExecuteQueueMBean <i>Attribute:</i> Name	
Queue Length	The maximum number of simultaneous requests that the server instance can hold in the queue.	<i>Minimum:</i> 256 <i>Maximum:</i> 1073741824 <i>Default:</i> 65536
Queue Length Threshold Percent	The percentage (from 1-99) of the Queue Length size that can be reached before the server indicates an overflow condition for the queue. All actual queue length sizes below the threshold percentage are considered normal; sizes above the threshold percentage indicate an overflow. When an overflow condition is reached, WebLogic Server logs an error message and increases the number of threads in the queue by the value of the Threads Increase attribute to help reduce the workload.	<i>Minimum:</i> 1 <i>Maximum:</i> 99 <i>Default:</i> 90

Table 489-1

Attribute Label	Description	Value Constraints
Thread Count	The number of threads assigned to this queue. <i>MBean:</i> weblogic.management.configuration.ExecuteQueueMBean <i>Attribute:</i> ThreadCount	<i>Minimum:</i> 0 <i>Maximum:</i> 65536 <i>Default:</i> 15
Threads Increase	The number of threads to be added to the queue when an overflow condition occurs. If you specify zero threads (the default), the server instance changes its health state to “warning” in response to an overflow condition in the thread, but it does not increase the thread count.	<i>Minimum:</i> 0 <i>Maximum:</i> 65536 <i>Default:</i> 0
Threads Maximum	The maximum number of threads that this queue can have; this value prevents WebLogic Server from creating an overly high thread count in the queue in response to continual overflow conditions. <i>MBean:</i> weblogic.management.configuration.ExecuteQueueMBean <i>Attribute:</i> ThreadsMaximum	<i>Minimum:</i> 1 <i>Maximum:</i> 65536 <i>Default:</i> 400
Threads Minimum	The minimum number of threads that WebLogic Server will maintain in the queue.	<i>Minimum:</i> 0 <i>Maximum:</i> 65536 <i>Default:</i> 5
Thread Priority	The priority of the threads associated with this queue.	<i>Minimum:</i> 1 <i>Maximum:</i> 10 <i>Default:</i> 5



Execute Queue --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this page to enter optional descriptive notes for a user-defined execute queue.

Tasks

Enter free-form text notes to describe how applications use this execute queue. Click Apply to apply your changes.

Related Topics

["Tuning the Default Execute Threads"](#) in *WebLogic Server Performance and Tuning*.

["Using Execute Queues to Control Thread Usage"](#) in *WebLogic Server Performance and Tuning*.

["Assigning Applications to Execute Queues"](#) in *WebLogic Server Performance and Tuning*.

Attributes

Table 490-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration. ExecuteQueueMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

Active Execute Queue

As work enters a WebLogic Server, it is placed in an execute queue. This work is then assigned to a thread within the queue that performs the work.

This page displays current runtime characteristics and statistics for the server's active execute queues.

By default, a new server instance is configured with a default execute queue, `weblogic.kernel.default`, that contains 15 threads. In addition, WebLogic Server provides two other pre-configured queues:

`weblogic.admin.HTTP`—Available only on Administration Servers, this queue is reserved for communicating with the Administration Console; you cannot reconfigure it.

`weblogic.admin.RMI`—Both Administration Servers and Managed Servers have this queue; it is reserved for administrative traffic; you cannot reconfigure it.

Unless you configure additional execute queues, and assign applications to them, web applications and RMI objects use `weblogic.kernel.default`. For more information, see:

- [“Creating Execute Queues”](#) in *WebLogic Server Performance and Tuning*.
- [“Detecting “Stuck” Threads”](#) in *WebLogic Server Performance and Tuning*.



Execute Queue

As work enters a WebLogic Server, it is placed in an execute queue. This work is then assigned to a thread within the queue that performs the work.

By default, a new server instance is configured with a default execute queue, `weblogic.kernel.default`, that contains 15 threads. In addition, WebLogic Server provides two other pre-configured queues:

`weblogic.admin.HTTP`—Available only on Administration Servers, this queue is reserved for communicating with the Administration Console; you cannot reconfigure it.

`weblogic.admin.RMI`—Both Administration Servers and Managed Servers have this queue; it is reserved for administrative traffic; you cannot reconfigure it.

Unless you configure additional execute queues, and assign applications to them, web applications and RMI objects use `weblogic.kernel.default`.

This page displays the basic configured settings for the default execute queue and other configurable, user-defined execute queues on this server.

For more information, see:

- [Creating Execute Queues](#)
- [Detecting “Stuck” Threads](#)



Transactions By Name

The Transactions by Name table shows aggregate statistics about transactions coordinated by this server for each named transaction. Statistics for unnamed transactions are not included in this table. You can click column headings in the table to sort the information in the table. You can also click [Customize this view](#) to select the columns to display in the table.

- For more information about monitoring transactions, see [“Monitoring Transactions” on page 237-5](#).
- For more information about configuring WebLogic JTA in the Administration Console, see [“JTA” on page 237-1](#).

Attributes

The Transactions by Name table includes the following columns:

Name—The transaction name as specified in the application that created the transaction.

Transactions—The total number of transactions processed with this transaction name.

Commits—The total number of committed transactions with this transaction name.

Rollbacks—The total number of transactions with this transaction name that were rolled back.

Timeout Rollbacks—The number of transactions with this transaction name that were rolled back due to a timeout expiration.

Resource Rollbacks—The number of transactions with this transaction name that were rolled back due to a resource failure. A resource can be a database or other XA-compliant transaction participant.

Application Rollbacks—The number of transactions with this transaction name that were rolled back due to an application failure.

System Rollbacks—The number of transactions with this transaction name that were rolled back due to an internal system error.

Heuristics—The number of transactions with this transaction name that complete with a heuristic status. A heuristic completion (or heuristic decision) occurs when a resource makes a unilateral decision during the completion stage of a distributed transaction to commit or rollback updates. See [“Handling Heuristic Completions” on page 237-14](#).

Transaction Abandoned Total Count—The number of transactions with this transaction name that were abandoned. An abandoned transaction is a transaction during which the processing time for the second phase of the two-phase commit process exceeds the limit set in the JTA configuration. See [“Abandoning Transactions” on page 237-15](#) for more information.

Transactions by Resource

The Transactions by Resource table shows transaction statistics about for each transactional resource accessed on this server. You can click column headings in the table to sort the information in the table. You can also click Customize this view to select the columns to display in the table.

- For information about heuristic completions, see [“Handling Heuristic Completions” on page 237-14](#).
- For information about configuring WebLogic JTA in the Administration Console, see [“Configuring JTA” on page 237-2](#).
- To learn more about WebLogic JTA, see [“JTA” on page 237-1](#) and [Programming WebLogic JTA](#).

Attributes

The Transactions by Resource table includes the following columns:

Name—The resource name on the server. A resource is any XA-compliant resource that can participate in a distributed transaction, such as a connection pool or a JMS store.

Transactions—The number of transactions in which the resource participated.

Commits—The number of transactions in which the resource committed the transaction.

Rollbacks—The number of transactions in which the resource rolled back the transaction.

Heuristics—The number of transactions for which the resource reported a heuristic decision, either a heuristic commit, a heuristic rollback, a mixed heuristic decision, or a heuristic hazard decision. See [“Handling Heuristic Completions” on page 237-14](#).

Heuristic Commits—The number of transactions for which the resource reported a heuristic commit.

Heuristic Rollbacks—The number of transactions for which the resource reported a heuristic rollback.

Mixed Heuristics—The number of transactions for which the resource reported a mixed heuristic decision. In a mixed heuristic decision, the Transaction Manager is aware that a transaction resulted in a mixed outcome, where some participating resources committed and some rolled back.

Heuristic Hazards—The number of transactions for which the resource reported a heuristic hazard decision. In a heuristic hazard decision, the Transaction Manager is aware that a transaction might have resulted in a mixed outcome, where some participating resources committed and some rolled back, but system or resource failures make it impossible to know for sure whether a heuristic mixed outcome definitely occurred.

1 Creating, Configuring, and Monitoring Servers

[“Attributes and Console Screen Reference for Servers” on page 496-1]

If a domain’s Administration Server is running, you can use the Administration Console to add and remove servers in the domain and to configure all of the domain’s properties. In addition, you can use the Administration Console to monitor the performance and overall health of a domain.

If you want to create a new domain, use the Configuration Wizard. You can also use the Configuration Wizard to modify many features of a domain’s configuration without starting server instances in the domain. The Configuration Wizard can configure only a subset of a domain’s features. For more information, refer to [“Creating and Configuring Domains Using the Configuration Wizard.”](#)

The following sections describe how to create, configure, and monitor servers from the Administration Console:

- “Adding and Removing Servers in an Existing Domain” on page 495-2
- “Configuring the Default Network Connections” on page 495-6
- “Configuring a Custom Network Channel for a Non-Clustered Server” on page 495-15
- “Transitioning Domains from Development to Production Environments: Main Steps” on page 495-16
- “Other Configuration Tasks” on page 495-18

- “Monitoring a Server” on page 495-20

Adding and Removing Servers in an Existing Domain

A domain can include multiple WebLogic Server instances. A minimal domain contains only one WebLogic Server instance, which functions both as an Administration Server and as a Managed server—such a domain can be useful while developing applications, but is not recommended for use in a production environment.

A **Managed Server** is a WebLogic Server instance that retrieves its configuration data from the domain’s Administration Server. There can be many Managed Servers in a domain, but only one Administration Server. Usually, you create and start server instances as Managed Servers to run your business applications in a production environment. In this standard scenario, the server instance that you start as the Administration Server does not run business applications. Instead, it only manages resources in the domain. To improve reliability and performance, you can install the WebLogic Server software on several computers and run the servers that you create on the various WebLogic Server hosts. For more information about Managed Servers and Administration Servers, refer to "[WebLogic Server Domains](#)."

Any server instance that you define for a domain can run as an Administration Server or a Managed Server. There is no attribute within a server’s configuration that designates it as an Administration Server or Managed Server. Instead, the first server instance that you start in a domain always functions as the Administration Server. If you start additional servers in a domain, you must start them as Managed Servers. For more information, refer to “Starting and Stopping Servers” on page 497-1.

The following sections describe how to add and remove servers:

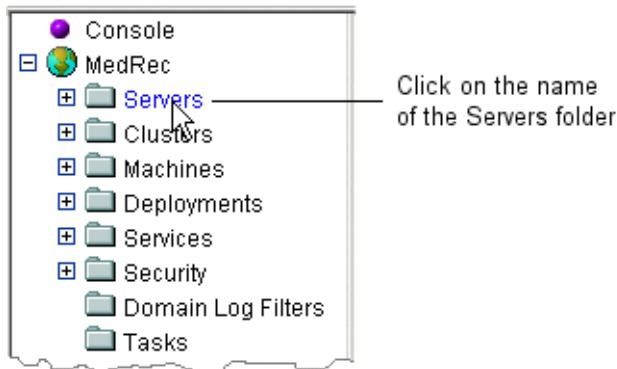
- “Creating a Managed Server in an Existing Domain” on page 495-3
- “Cloning a Server” on page 495-4
- “Deleting a Managed Server” on page 495-5
- “Deleting an Administration Server” on page 495-5

Creating a Managed Server in an Existing Domain

To create a Managed Server in an existing domain:

1. Start the Administration Server.
See “Starting Administration Servers” on page 497-2.
2. In the left pane of the Administration Console, click on the name of the Servers folder. (See Figure 495-1.)

Figure 495-1 Click on the Name of the Servers Folder



3. On the Servers page, click Configure a new Server.

The Administration Console displays the Servers →Create page in the right pane.

4. On the Create page, enter the name of the server in the Name field.

Each server within a domain must have a name that is unique for all configuration objects in the domain. Within a domain, each server, machine, cluster, JDBC connection pool, virtual host, and any other resource type must be named uniquely and must not use the same name as the domain.

The server name is not used as part of the URL for applications that are deployed on the server. It is for your identification purposes only. The server name displays in the Administration Console, and if you use WebLogic Server command-line utilities or APIs, you use this name to identify the server.

5. For information about modifying the default values on the General tab, refer to “Attributes” on page 434-2.

6. Click Create.

The new server appears under the Servers node in the left pane. The Administration Console updates the domain's `config.xml` file with the new server configuration data.

Cloning a Server


Cloning a server creates a new server instance with the same attributes as the original server.

1. Start the Administration Server.

See “Starting Administration Servers” on page 497-2.

2. In the left pane of the Administration Console, click on the name of the Servers folder. (See Figure 495-1.)

The Servers page displays a list of servers that have already been defined in the domain.

3. On the Servers page, click the Clone icon  in the row of the server you want to clone.

A dialog displays the tabs associated with cloning a new server.

4. Enter a new name for the server.
5. 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 server appears under the Servers node in the left pane. The Administration Console updates the domain's `config.xml` file with the new server configuration data.

Deleting a Managed Server


When you delete a server, WebLogic Server removes its associated configuration data from the domain's configuration file (`config.xml`). To see which data will be deleted, select the server in the left pane of the Administration Console. All of the data in the right pane will be deleted. For example, any network channels that you created for the server are deleted, but applications and EJBs that are deployed on the server will not be deleted.

You cannot delete a server that is currently active; therefore, you cannot use the Administration Console to delete the Administration Server. (The Administration Console runs on the Administration Server.) For more information, see "Deleting an Administration Server" on page 495-5.

You cannot delete a server if it is running a pinned service. Before you can delete such a server, you must migrate the service to a migratable target. See "[Migration for Pinned Services](#)."

To delete a Managed Server:

1. Start the Administration Server.
See "Starting Administration Servers" on page 497-2.
2. In the left pane of the Administration Console, click on the name of the Servers folder. (See Figure 495-1.)

The Servers page displays a list of servers that have already been defined in the domain.
3. On the Servers page, click the Delete icon  in the row of the server you want to delete.
4. Click Yes to confirm your deletion request.

Deleting an Administration Server

To delete the server instance that you are using as an Administration Server:

1. Shut down all server instances in the domain, including the Administration Server.

2. Start another server in the domain as the Administration Server by entering the following command:

```
java -Dweblogic.Name=serverName weblogic.Server
```

See “Starting an Administration Server With the java weblogic.Server Command” on page 497-4.

3. Use the Administration Console to delete the server instance that you were previously using as the Administration Server.

Configuring the Default Network Connections

Each WebLogic Server instance provides default settings for the protocols, listen addresses, and listen ports through which it can be reached. These settings are referred to collectively as the default network channel. This default network channel provides two listen ports through which it receives requests: one for non-SSL requests and the other for SSL requests. You can disable one of these ports, but at least one must be enabled.

The following sections describe how to configure the default network channel for a server instance:

- “Configuring Protocols” on page 495-7
- “Configuring the Listen Address” on page 495-11
- “Configuring the Listen Ports” on page 495-13

You can configure additional network channels to meet various connection requirements and improve the utilization of your system and network resources. For information on configuring additional network channels, refer to "[Configuring Network Resources](#)."

Configuring Protocols

Servers can be reached through the following URL:

protocol://listen-address:listen-port

The default network channel supports multiple protocols for communicating with a server instance. By default, clients can contact a server instance through the HTTP and HTTPS protocols. BEA utilities (such as the `weblogic.Admin` command-line utility) can also connect to servers using the proprietary T3 and T3S protocols.

The following sections describe how to enable and configure communications protocols for a WebLogic Server instance:

- “Configuring the HTTP Protocol” on page 495-7
- “Configuring the T3 Protocol” on page 495-8
- “Enabling and Configuring the IIOP Protocol” on page 495-9
- “Enabling and Configuring jCOM” on page 495-10

Configuring the HTTP Protocol

The server instance for which you configure the HTTP protocol does not need to be running. If it is running, and if you change settings on the HTTP tab, you must restart it. Changes on the Protocols → General tab take effect without requiring you to restart the server.

To configure the HTTP protocol:

1. Start the Administration Server.
See “Starting Administration Servers” on page 497-2.
2. In the left pane of the Administration Console, expand the Servers folder and select a server. (See Figure 495-3.)
3. Select Protocols → HTTP.
4. Modify the default settings as desired. See “Attributes” on page 465-1.
5. Click Apply.
6. To enable tunneling of connections:

- a. Select Protocols →General.
 - b. In the Advanced Options bar, click Show.
 - c. Click Enable Tunneling and provide values for Tunneling Client Ping and Tunneling Client Timeout. For information about each setting, refer to “Attributes” on page 484-1.

Note: These settings apply to all protocols in the server’s default network configuration that support tunneling.

Also see “[Setting Up WebLogic Server for HTTP Tunneling.](#)”
 - d. Click Apply.
7. If the server instance is running, and if you made changes on the HTTP tab, restart the server.

Configuring the T3 Protocol

The server instance for which you configure the T3 protocol does not need to be running. If it is running, all modifications to the T3 protocol settings take effect immediately.

To configure the T3 protocol:

1. Start the Administration Server.

See “Starting Administration Servers” on page 497-2.
2. In the left pane of the Administration Console, expand the Servers folder and select a server. (See Figure 495-3.)
3. Select Protocols →General.
4. In the Advanced Options bar, click Show.
5. Modify the values for Complete Message Timeout, and Maximum Message Size. See “Attributes” on page 484-1.

Note: These settings apply to all protocols in the server’s default network configuration. See “[The Default Network Channel.](#)”
6. To enable tunneling of connections, click Enable Tunneling and provide values for Tunneling Client Ping and Tunneling Client Timeout. See “Attributes” on page 484-1.

Note: These settings apply to all protocols in the server's default network configuration that support tunneling.

7. Click Apply.

Enabling and Configuring the IIOP Protocol

The IIOP (Internet Inter-ORB Protocol) protocol makes it possible for distributed programs written in different programming languages to communicate over the Internet. For information about using RMI-IIOP in your applications, refer to the ["Programming WebLogic RMI over IIOP"](#) guide.

The server instance for which you enable and configure the IIOP protocol does not need to be running. If it is running, you must restart it after you complete these steps.

To enable and configure the IIOP protocol:

1. Start the Administration Server.
See "Starting Administration Servers" on page 497-2.
2. In the left pane of the Administration Console, expand the Servers folder and select a server. (See Figure 495-3.)
3. Select Protocols →IIOP.
4. Enable the IIOP protocol by checking the Enable IIOP check box.
5. To modify the default configuration, in the Advanced Options bar, click Show.
6. If you want to specify a default IIOP user name and password:
 - a. In the Default IIOP User field, enter a user name.
 - b. Click Apply.
 - c. In Default IIOP Password, enter a password.
 - d. In Confirm Default IIOP Password, enter the password again.
7. To configure the other options on the IIOP tab, see "Attributes" on page 464-1.
8. Click Apply.
9. To configure maximum messages size and timeout settings:

- a. Click the Protocols—General tab.
 - b. In the Advanced Options bar, click Show.
 - c. Modify the values for Complete Message Timeout, Idle Connection Timeout, and Maximum Message Size. For information about each setting, refer to “Attributes” on page 484-1.
- Note:** These settings apply to all protocols in the server’s default network configuration. See “[The Default Network Channel](#).”
- d. Click Apply.
10. To enable tunneling of connections:
- a. Select Protocols →General.
 - b. In the Advanced Options bar, click Show.
 - c. Click Enable Tunneling and provide values for Tunneling Client Ping and Tunneling Client Timeout. See “Attributes” on page 484-1.
- Note:** These settings apply to all protocols in the server’s default network configuration that support tunneling.
- d. Click Apply.
11. If the server instance is running, restart it.

Enabling and Configuring jCOM

WebLogic jCOM is a software bridge that allows bidirectional access between Java/J2EE objects deployed in WebLogic Server, and Microsoft ActiveX components available within Microsoft Office family of products, Visual Basic and C++ objects, and other Component Object Model /Distributed Component Object Model (COM/DCOM) environments. For more information about WebLogic jCOM, refer to the “[Programming WebLogic jCOM](#)” guide.

The server instance for which you enable and configure the jCOM protocol does not need to be running. If it is running, you must restart it after you complete these steps.

To enable and configure the jCOM protocol:

1. Start the Administration Server.
See “Starting Administration Servers” on page 497-2.

2. In the left pane of the Administration Console, expand the Servers folder and select a server. (See Figure 495-3.)
3. Select Protocols → jCOM.
4. Enable the jCOM protocol by checking the Enable COM check box.
5. Use the other items on the jCOM tab to configure jCOM. See “Attributes” on page 461-2.
6. Click Apply.
7. If the server instance is running, restart it.

Configuring the Listen Address

Servers can be reached through the following URL:

`protocol://listen-address:listen-port`

By default, a server’s listen address attribute is undefined, which enables you to reach the server through any of the following listen addresses:

- The primary IP address of the computer that hosts the server
- The host DNS name
- The `localhost` string (valid only for requests that are issued from the computer on which the server is running)

If the server instance must be accessible as `localhost` (for example, if you create administrative scripts that connect to `localhost`), and must also be accessible by remote processes, leave the listen address blank. Otherwise, if you want to limit the valid listen address for a server, refer to Table 495-2 for guidelines on specifying listen addresses.

Note: On multi-homed Windows NT machines, if you leave the listen address undefined or if you specify a DNS name, a server instance binds to all available IP addresses.

Table 495-2 Setting the Listen Address

If the Listen Address is set to...	Then the following is true...
IP address or DNS name	<ul style="list-style-type: none">■ To connect to the server instance, processes can specify either the IP address or the corresponding DNS name.■ Processes that specify <code>localhost</code> will fail to connect.■ You must update existing processes that use <code>localhost</code> to connect to the server instance.■ Connections that specify the IP address for the listen address and a secured port for the listen port must disable host name verification.
<code>localhost</code>	<ul style="list-style-type: none">■ Processes must specify <code>localhost</code> to connect to the server instance.■ Only processes that reside on the machine that hosts the server instance (local processes) will be able to connect to that server instance.

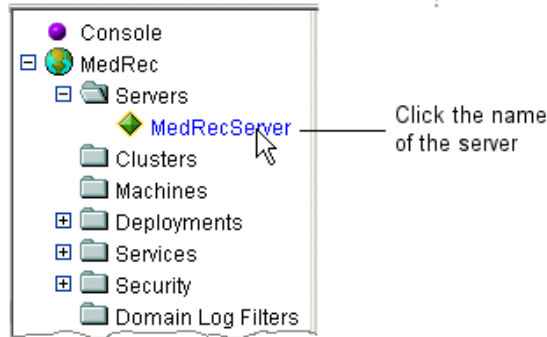
Configuring the Listen Address from the Administration Console

The server instance for which you configure the listen address does not need to be running. If it is running, you must restart it after you complete these steps.

To configure the listen address from the Administration Console:

1. Start the Administration Server.
See “Starting Administration Servers” on page 497-2.
2. In the left pane of the Administration Console, expand the Servers folder and select a server. (See Figure 495-3.)

Figure 495-3 Click on a Server



3. In the right pane, select Configuration → General.
4. Enter a value in the Listen Address box. Refer to Table 495-2 for guidelines.
5. Click Apply.
6. If the server is running, restart it.

Configuring the Listen Ports

Servers can be reached through the following URL:

protocol://listen-address:listen-port

Each WebLogic Server instance defines two listen ports in its default network channel: one for regular, non-secure requests (via such protocols as HTTP and T3) and the other for secure requests (via such protocols as HTTPS and T3S).

Note: By default, a server instance uses demonstration certificates to authenticate requests from the secure port. In a production environment, you must configure SSL to use certificates from a certificate authority. See "[Configuring the SSL Protocol](#)" in the *Managing WebLogic Security* guide.

You can disable either the default non-SSL or the default SSL listen port, but at least one must be enabled, even if you create one or more network channels for the server.

Although you can specify any valid port number, if you specify port 80, you can omit the port number from the HTTP request used to access resources over HTTP. For example, if you define port 80 as the listen port, you can use the URL

`http://hostname/myfile.html` instead of
`http://hostname:portnumber/myfile.html`.

On some operating systems, port 80 can be accessed only by processes that run under a privileged user or group ID. In this case, you can assign the server instance to a UNIX Machine that has defined a Post-Bind UID or GID. For more information, refer to “Machines.”

If you run multiple instances of WebLogic Server on a single computer, each instance must use a unique listen port/listen address combination. On a multihomed computer (a computer that can be accessed through multiple IP addresses), you can use the same listen port but configure each server to use a unique IP address as its listen address. If your computer does not support multiple IP addresses, you must use a different listen port for each active instance.

Configuring the Listen Ports from the Administration Console

The server instance for which you configure the listen ports does not need to be running. If it is running, you must restart it after you complete these steps.

To configure the listen ports from the Administration Console:

1. Start the Administration Server.
See “Starting Administration Servers” on page 497-2.
2. In the left pane of the Administration Console, expand the Servers folder and select a server. (See Figure 495-3.)
3. In the right pane, select Configuration → General.
4. If you want to disable the non-SSL listen port so that the server listens only on the SSL listen port, remove the checkmark from the Listen Port Enabled box.

If you want to disable the SSL listen port so that the server listens only on the non-SSL listen port, remove the checkmark from the Enable SSL Listen Port box.

Note: You cannot disable both the non-SSL listen port and the SSL listen port. At least one port must be active.

5. If you are using the non-SSL listen port and you want to modify the default port number, change the default number in the Listen Port box.
6. If you want to modify the default SSL listen port number change the value in the SSL Listen Port box.
7. Click Apply.
8. If the server is running, restart it.

Configuring a Custom Network Channel for a Non-Clustered Server

You can configure custom network channels to meet varying connection requirements and improve utilization of your systems and network resources. See "[Configuring Network Resources](#)."

Note: If this server belongs to a cluster, refer to "[Configuring Network Channels with a Cluster](#)."

The server instance for which you configure a custom network channel does not need to be running. If it is running, you must restart it after you complete these steps.

To configure a custom network channel from the Administration Console:

1. Start the Administration Server.
See "Starting Administration Servers" on page 497-2.
2. In the left pane of the Administration Console, expand the Servers folder and select a server. (See Figure 495-3.)
3. In the right pane, select Configuration ⇒ Protocols ⇒ Channels.
4. Click Configure a new Network Channel.
5. Enter the Name, Protocol, Listen Address and Listen Port for the new network channel and click Create.

6. To support Network Address Translation (NAT) firewalls, in External Listen Address enter the IP address or DNS name that clients use to access applications on the server instance.
7. To configure additional attributes for this network channel:
 - a. In the Advanced Options bar, click Show.
 - b. Modify the values in advanced options. See “Advanced Attributes” on page 483-4.
8. Click Apply.
9. If the server is running, restart it.

Transitioning Domains from Development to Production Environments: Main Steps

In a production environment, the security requirements are typically much stricter than in a development environment. The network environment is typically more complex, and the need for monitoring and availability is greater.

The following steps outline a logical order for transitioning a server that you originally configured for a development environment to a production environment:

1. Secure the hardware, WebLogic Server software, and your application in the production environment. See "[Securing a Production Environment](#)."
2. If you want to use multiple Network Interface Cards (NICs) and/or multiple port numbers in your domain to improve performance and solve common networking problems, set up Network Channels. For more information, refer to "[Configuring Network Resources](#)."
3. Configure servers to start in production mode. For more information, refer to “Changing the Runtime Mode” on page 495-17.
4. Configure servers to use a JVM that is optimized for production-level performance. For more information, refer to "[Changing the JVM that Runs Servers](#)."

On a Windows or Linux platform, BEA recommends using the JVM that the WebLogic JRockit SDK provides. This JVM provides optimal running performance but initial startup cycles can require more time than other JVMs.

Note that when you change JVMs, you most likely need to adjust the memory available to the JVM.

5. If you want to use the Administration Console to start the Managed Servers in your domain, do the following:
 - a. Configure a Node Manager to run on a specific computer (machine). For more information, refer to [“Configuring a Machine.”](#)
 - b. Start the Node Manager. The Node Manager is a small program that can run on WebLogic Server hosts. In addition to enabling you to start server instances from the Administration Console, the Node Manager can also restart servers that report themselves to be in an unhealthy state. For more information, refer to [“Starting and Stopping Node Manager.”](#)
 - c. Start the Managed Server from the Administration Console. For more information, refer to [“Starting Managed Servers from the Administration Console.”](#)
6. If you want to use the Node Manager to automatically restart unhealthy servers, refer to [“Configure Monitoring, Shutdown, and Restart for Managed Servers.”](#)
7. If you want to use a WebLogic Server instance as a Web server, refer to [“Configuring WebLogic Server Web Components.”](#)

The WebLogic Server Web server component can handle high volume Web sites, serving static files such as HTML files and image files as well as servlets and JavaServer Pages (JSP).

Changing the Runtime Mode

After you create a domain, do the following to change the mode in which all servers in a domain run:

1. To change the runtime mode for all domains that run on a WebLogic Server host, in a text editor, open `WL_HOME\common\bin\commEnv.cmd` (Windows) or `WL_HOME\common\bin\commEnv.sh` (UNIX) where `WL_HOME` is the directory in which you installed WebLogic Server.

To change the runtime mode only for a specific domain, in a text editor, open `domain-name\StartWebLogic.cmd` (Windows) or `domain-name\StartWebLogic.sh` (UNIX).

where `domain-name` is the directory which you created the domain.

2. In the script, change the value for the `PRODUCTION_MODE` variable.
Specify `TRUE` if you want servers to start in production mode.
3. Shut down any servers that are currently running.
4. If you changed the mode from development mode to production mode, see “Transitioning Domains from Development to Production Environments: Main Steps” on page 495-16.
5. Invoke the domain’s `StartWebLogic` script.

The Administration Server starts in the new mode.

6. If the domain contains Managed Servers, start the Managed Servers.

As each Managed Server starts, it refers to the mode of the Administration Server to determine its runtime mode.

For information on the differences between development mode and production mode, refer to “[Differences Between Configuration Startup Modes](#).”

Other Configuration Tasks

The following sections describe miscellaneous configuration tasks:

- “Replicating a Domain’s Configuration Files for Managed Server Independence” on page 495-19
- “Disabling Managed Server Independence” on page 495-19
- “[Creating Execute Queues](#)” in *WebLogic Server Performance and Tuning*.

Replicating a Domain's Configuration Files for Managed Server Independence

The server instance for which you configure Managed Server Independence (MSI) replication does not need to be running. If it is running, you must restart it after you complete these steps.

To configure a Managed Server to replicate a domain's configuration files:

1. Start the Administration Server.
See "Starting Administration Servers" on page 497-2.
2. In the left pane of the Administration Console, expand the Servers folder and select a server. (See Figure 495-3.)
3. In the right pane, select Configuration → Tuning.
4. In the Advanced Options bar, click Show.
5. Under Advanced Options, make sure that the Managed Server Independence Enabled check box is checked.
6. Click the MSI File Replication Enabled check box.
7. Click Apply.
8. If the server is running, restart it.

Disabling Managed Server Independence

When a Managed Server starts, it tries to contact the Administration Server to retrieve its configuration information. If a Managed Server cannot connect to the Administration Server during startup, it can retrieve its configuration by reading configuration and security files directly. A Managed Server that starts in this way is running in *Managed Server Independence (MSI)* mode. For more information about MSI mode, refer to "[Managed Server Independence Mode](#)."

By default, MSI mode is enabled.

The server instance for which you want to disable MSI does not need to be running. If it is running, you must restart it after you complete these steps.

To disable MIS:

1. Start the Administration Server.
See “Starting Administration Servers” on page 497-2.
2. In the left pane of the Administration Console, expand the Servers folder and select a server. (See Figure 495-3.)
3. In the right pane, select Configuration → Tuning.
4. In the Advanced Options bar, click Show.
5. Under Advanced Options, clear the Managed Server Independence Enabled check box.
6. Click Apply.
7. If the server is running, restart it.

Monitoring a Server

The following tasks monitor the performance and health state of servers:

- “Monitoring a Server Instance from the Administration Console” on page 495-21
- “Determining the Platform on Which a Server Is Running” on page 495-21
- “Monitoring the JRockit Virtual Machine” on page 495-22
- “Configure Self-Health Monitoring, Shutdown, and Restart for Managed Servers” on page 495-22

For more information about monitoring WebLogic Server, refer to "[Monitoring a WebLogic Server Domain](#)."

Monitoring a Server Instance from the Administration Console

The WebLogic Server Administration Console provides visibility into a broad array of configuration and status information of a server instance.

The server instance that you want to monitor **must be running**. WebLogic Server does not archive performance statistics.

To monitor a server instance from the Administration Console:

1. Start the Administration Server and the server instance you want to monitor.
See “Starting and Stopping Servers” on page 497-1.
2. In the left pane of the Administration Console, expand the Servers folder and select a server. (See Figure 495-3.)
3. In the right pane, select Configuration → Monitoring.
4. Refer to "[Server Monitoring Pages](#)."

Determining the Platform on Which a Server Is Running

To determine the platform on which a server instance is running:

1. Start the Administration Server.
See “Starting Administration Servers” on page 497-2.
2. In the left pane of the Administration Console, expand the Servers folder and select a server. (See Figure 495-3.)
3. In the right pane, select Configuration → Monitoring.
4. In the Advanced Options bar, click Show.
5. Under Advanced Options, the Administration Console indicates the software platform on which the server is running.

Monitoring the JRockit Virtual Machine

If you run a server with the JRockit Virtual Machine (VM), you can view runtime data about the underlying JRockit VM and the memory and processors on the computer that is hosting the VM.

To monitor the JRockit VM:

1. Use the JRockit VM to start the server instance.
2. Start the Administration Server.
See “Starting Administration Servers” on page 497-2.
3. In the left pane of the Administration Console, expand the Servers folder and select a server that is using the JRockit VM. (See Figure 495-3.)
4. In the right pane, select Monitoring ⇒ JRockit.

The JRockit tab displays monitoring information.

To view additional data about the VM, such as how long it spends in a specific method, use the JRockit Management Console. Note that if you want to use the JRockit Management Console, you must include the `-XManagement` startup option when you start the server. (You do not need this option to use the WebLogic Server Administration Console to monitor the VM.) For more information, refer to "[JRockit Documentation Home Page](#)."

Configure Self-Health Monitoring, Shutdown, and Restart for Managed Servers

A server can also monitor key aspects of its subsystems and report when a subsystem is not functioning properly. If the server is running under a Node Manager, the Node Manager can automatically restart a server with an unhealthy subsystem.

The server instance for which you want to self-health monitoring does not need to be running. If it is running, you must use the Node Manager to restart it after you complete these steps.

Follow these steps to configure Node Manager features for monitoring, shutting down, and restarting a Managed Server:

1. Start the Administration Server.
See “Starting Administration Servers” on page 497-2.
2. In the left pane of the Administration Console, expand the Servers folder and select a server. (See Figure 495-3.)
3. In the right pane, select Configuration → Health Monitoring.
4. On the Health Monitoring tab, edit the values. For information about each attribute, refer to “Attributes” on page 435-2.
5. Click Apply.
6. If the server is running, shut it down.
7. Use the Node Manager to start the server. Node Manager can perform automatic monitoring and shutdown only for servers that it starts.

Attributes and Console Screen Reference for Servers

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

Server --> Configuration

[“Server --> Configuration --> General” on page 434-1](#)

[“Server --> Configuration --> Cluster” on page 432-1](#)

[“Servers-->Configuration-->Keystores and SSL” on page 436-1](#)

[“Server --> Configuration --> Deployment” on page 433-1](#)

[“Server --> Configuration --> Tuning” on page 438-1](#)

[“Server --> Configuration --> Health Monitoring” on page 435-1](#)

[“Server --> Configuration --> Remote Start” on page 437-1](#)

SSL/Keystore Configuration

[“Specify Keystore Type” on page 485-1](#)

[“Configure SSL-->Determine Compatibility Level” on page 431-1](#)

[“SSL Identity Configuration” on page 444-1](#)

[“Deprecated SSL Identity and Trust Configuration” on page 440-1](#)

[“Configure Keystore Properties” on page 487-1](#)

[“Review SSL Private Key Settings” on page 486-1](#)

[“Restart Your Server” on page 488-1](#)

Servers --> Protocols

“Server --> Protocols --> General” on page 484-1

“Server --> Protocols --> HTTP” on page 465-1

“Server --> Protocols --> jCOM” on page 461-1

“Server --> Protocols --> IIOP” on page 464-1

“Servers --> Protocols --> Channels” on page 466-1

Servers --> Protocols --> Channels

Servers --> Logging

“Server --> Logging --> Server” on page 473-1

“Server --> Logging --> Domain” on page 445-1

“Server --> Logging --> HTTP” on page 446-1

“Server --> Logging --> JDBC” on page 447-1

“Server --> Logging --> JTA” on page 448-1

Servers --> Monitoring

“Server --> Monitoring --> General” on page 471-1

“Server --> Monitoring --> Performance” on page 468-1

“Server --> Monitoring --> JRockit” on page 472-1

“Server --> Monitoring --> Security” on page 467-1

“Server --> Monitoring --> JMS” on page 470-1

“Server --> Monitoring --> JTA” on page 469-1

Servers --> Control

- “Server --> Control --> Start-Stop” on page 439-1
- “Server --> Control --> Remote Start Output” on page 478-1
- “Server --> Control --> JMS Migrate” on page 480-1
- “Server --> Control --> JMS Migration Config.” on page 481-1
- “Server --> Control --> JTA Migrate” on page 482-1
- “Server --> Control --> JTA Migration Config.” on page 479-1

Servers --> Deployments

- “Server --> Deployments --> Applications” on page 477-1
- “Server --> Deployments --> EJB Modules” on page 475-1
- “Server --> Deployments --> Web Modules” on page 474-1
- “Server --> Deployments --> Connector” on page 476-1
- “Server --> Deployments --> Startup/Shutdown” on page 443-1

Server --> Services

- “Server --> Services --> JDBC” on page 453-1
- “Server --> Services --> JMS” on page 454-1
- “Server --> Services --> Web Services” on page 463-1
- “Server --> Services --> Bridge” on page 451-1
- “Server --> Services --> XML” on page 458-1
- “Server --> Services --> WTC” on page 457-1
- “Server --> Services --> Jolt” on page 455-1

“Server --> Services --> Virtual Hosts” on page 462-1

“Server --> Services --> Mail” on page 456-1

“Server --> Services --> File T3” on page 452-1

Server --> Notes

“Server --> Notes” on page 430-1

Miscellaneous

View “Connection” on page 441-1

“Active Execute Queue” on page 491-1

“Execute Queue” on page 492-1

“Execute Thread” on page 442-1

“Execute Queue --> Configuration” on page 489-1

“Socket” on page 449-1

“Server” on page 459-1

1 Starting and Stopping Servers

WebLogic Server provides several ways to start and stop server instances. The method that you choose depends on whether you prefer using a graphical or command-line interface, and on whether you are using the Node Manager to manage a server's lifecycle.

No matter how you start a server, the end result passes a set of configuration options to initialize a Java Virtual Machine (JVM). The server instance runs within the JVM, and the JVM can host only one server instance.

The following sections describe starting and stopping server instances:

- “Version Requirements for Starting Servers” on page 497-2
- “Starting Administration Servers” on page 497-2
- “Alternate Ways to Start Administration Servers” on page 497-3
- “Starting Managed Servers from the Administration Console” on page 497-5
- “Starting Managed Servers From a WebLogic Server Script” on page 497-8
- “Alternate Ways to Start Managed Servers” on page 497-10
- “Providing Usernames and Passwords to Start and Stop a Server” on page 497-14
- “Other Startup Tasks” on page 497-22
- “Shutting Down Instances of WebLogic Server” on page 497-28

For a quick overview of starting and stopping servers, refer to "[Starting and Stopping WebLogic Server Instances: Quick Reference](#)."

Version Requirements for Starting Servers

The Administration Server and all Managed Servers in a domain must be the same WebLogic Server version. The Administration Server must be either at the same service-pack level or at a later service-pack level than the Managed Servers. For example, if the Managed Servers are at version 8.1, then the Administration Server can be either version 8.1 or 8.1 SP1. However, if the Managed Servers are at SP1, then the Administration Server must be at SP1.

Starting Administration Servers

An Administration Server is a WebLogic Server instance that maintains configuration data for a domain. In a development environment, it is usually sufficient to start an Administration Server and deploy your applications directly onto the Administration Server. In a production environment, you create Managed Servers to run applications. For more information about Administration Servers and Managed Servers, refer to "[Overview of WebLogic Server Domains](#)."

To start an Administration Server:

1. If you have not already done so, use the Configuration Wizard to create a domain. For more information, refer to "[Creating Domains and Servers](#)."
2. Open a shell (command prompt) on the computer on which you created the domain.
3. Change to the directory in which you located the domain.
By default, this directory is
`BEA_HOME\user_projects\domains\domain-name.`
4. Run one of the following scripts:
 - `startWebLogic.cmd` (Windows)
 - `startWebLogic.sh` (UNIX and Windows. On Windows, this script supports the MKS and Cygnus BASH UNIX shell emulators.)

Note: If you use a Configuration Wizard template that is provided by WebLogic Server, your domain directory includes a start script named `startWebLogic`. If you use a domain template from another source, the wizard might not create a start script, or it might create a script with a different name. The template designer determines whether the wizard creates a start script and the name of the script.

The WebLogic startup script does the following:

1. Sets environment variables by invoking `WL_HOME\common\bin\commEnv.cmd` (`commEnv.sh` on UNIX), where `WL_HOME` is the location in which you installed WebLogic Server.
2. Invokes the `java weblogic.Server` command, which starts a JVM that is configured to run a WebLogic Server instance.

When the server successfully completes its startup process, it writes the following message to standard out (which, by default, is the command window):

```
<Notice> <WebLogicServer> <000360> <Server started in RUNNING mode>
```

Alternate Ways to Start Administration Servers

The following sections describe alternate ways to start an Administration Server:

- “Starting an Administration Server from the Windows Start Menu” on page 497-4
- “Starting an Administration Server When the Host Computer Boots” on page 497-4
- “Starting an Administration Server With the `java weblogic.Server` Command” on page 497-4

You cannot use the Node Manager to start an Administration Server.

Starting an Administration Server from the Windows Start Menu

When you create an Administration Server on a Windows computer, the Configuration Wizard prompts you to install the server in the Windows Start Menu. If you choose yes, you can start the server instance from the Windows Start Menu.

The command that the Configuration Wizard adds to the Start menu opens a command window and calls the startup script that is described in “Starting Administration Servers.” When the server has successfully completed its startup process, it writes the following message to standard out (which, by default, is the command window):

```
<Notice> <WebLogicServer> <000360> <Server started in RUNNING mode>
```

Starting an Administration Server When the Host Computer Boots

If you want an Administration Server to start automatically when you boot a computer, you can set up the server as a UNIX daemon or a Windows service. Refer to the documentation for your UNIX operating system or to "[Setting Up a WebLogic Server Instance as a Windows Service](#)."

Starting an Administration Server With the java weblogic.Server Command

The `weblogic.Server` class is the main class for a WebLogic Server instance. You can start a server instance by directly invoking `weblogic.Server` in a Java command or by creating your own scripts that invoke the `weblogic.Server` class. (The WebLogic Server startup scripts invoke `weblogic.Server` in a Java command.)

For information about invoking `weblogic.Server` in a Java command, refer to "[weblogic.Server Command Line Reference](#)."

Starting Managed Servers from the Administration Console

A Managed Server is a WebLogic Server instance that runs deployed applications. It refers to the Administration Server for all of its configuration and deployment information. Usually, you use Managed Servers to run applications in a production environment. For more information about Managed Servers and Administration Servers, refer to "[Overview of WebLogic Server Domains](#)."

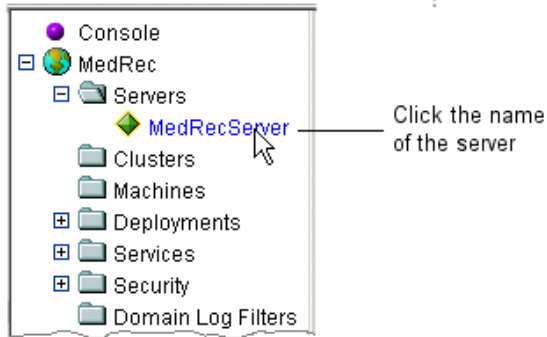
To use the Administration Console to start a Managed Server:

1. If you have not already done so, create a Managed Server. See "[Creating Domains and Servers](#)" or "Adding and Removing Servers in an Existing Domain" on page 495-2.
2. Configure the Managed Server to communicate with a Node Manager. See "Configuring a Machine" on page 268-1 and "Configure Startup Arguments for Managed Servers" on page 497-7.

The **Node Manager** is a standalone Java program provided with each WebLogic Server installation. You use it to start and stop Managed Servers, and to monitor and automatically restart Managed Servers based on server health. You cannot use the Node Manager to start Administration Servers. For more information on the Node Manager, refer to "[Overview of Node Manager](#)."

3. Start the Node Manager on the computer that you want to host the Managed Server. See "[Starting Node Manager](#)."
4. Start the Administration Server.
5. In the Administration Console, in the left pane, expand the Servers node and select a server. (See Figure 497-1.)

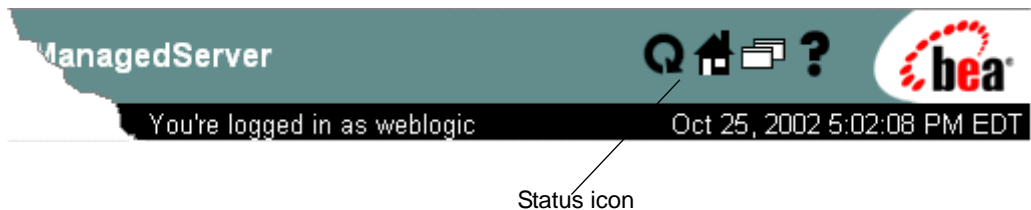
Figure 497-1 Click the Name of a Server



6. In the right pane, select Control →Start/Stop.
7. On the Start/Stop tab, select Start this server.
8. Click Yes to confirm.

The Administration Console displays an animated status icon while the Node Manager starts the server on the target machine. (See Figure 497-2.)

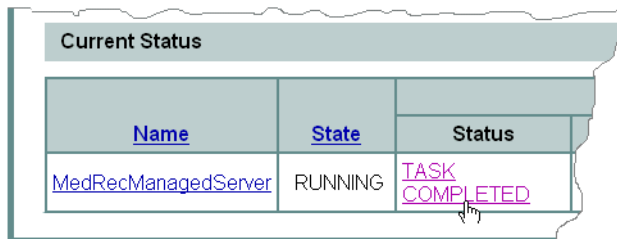
Figure 497-2 Status Icon



When the Node Manager finishes its start sequence, the status icon is no longer displayed and the server's state is indicated in the Current Status table, under the State column. (See Figure 497-3.)

9. To view messages that the Node Manager generated while starting the server, click the Task Completed link in the Current Status table.

Figure 497-3 View the Node Manager Output



Current Status		
Name	State	Status
MedRecManagedServer	RUNNING	TASK COMPLETED

These messages are also written to the Node Manager log file for that server, as described in "[Managed Server Log Files](#)."

Configure Startup Arguments for Managed Servers

In most environments, the Node Manager can start a server without requiring you to specify startup options. However, if you have modified your environment; for example, if you have added classes to the WebLogic Server classpath, you must specify startup options before you use the Administration Console to start a server.

To configure the startup options that Node Manager uses to start a Managed Server:

1. In the Administration Console, in the left pane, expand the Servers node and select a server. (See Figure 497-1.)
2. In the right pane, select Configuration → Remote Start.
3. On the Remote Start tab, the User Name and Password fields contain values that you supplied when you used the Administration Console or the Configuration Wizard to create the server.

If you want the server instance to run under a different WebLogic Server user account, enter the name of an existing user. The user must be in a role that has permission to start servers. For information on roles and permissions, refer to "[Security Roles](#)."

4. Use the remaining fields on this tab only if you want to override the default values that the Node Manager provides. See "[Configuring, Starting, and Stopping Node Manager](#)."

The Administration Console **replaces** the Node Manager defaults with the values you provide; it does not append the values to the Node Manager defaults.

If you provide values for the Classpath field, make sure that you provide the full class path required to start the Managed Server.

Note: All paths refer to paths on the Node Manager machine.

For more information about the values to enter in these fields, refer to “Attributes” on page 437-2.

5. Click Apply.
6. If the server is running, restart it.

Starting Managed Servers From a WebLogic Server Script

A Managed Server is a WebLogic Server instance that runs deployed applications. It refers to the Administration Server for all of its configuration and deployment information. Usually, you use Managed Servers to run applications in a production environment. For more information about Managed Servers and Administration Servers, refer to "[Overview of WebLogic Server Domains](#)."

If you use one of the Configuration Wizard templates that WebLogic Server provides, your domain directory includes a start script named `startManagedWebLogic` that you can use to start Managed Servers.

This script does not use the Node Manager to start and manage the server. Instead, it uses a Java command to invoke the `weblogic.Server` class, which is the main class for a WebLogic Server instance. For information about invoking `weblogic.Server` in a Java command, refer to "[weblogic.Server Command Line Reference](#)."

To use the WebLogic Server scripts to start a Managed Server:

1. If you have not already done so, create a Managed Server. See “[Creating Domains and Servers](#)” or “Adding and Removing Servers in an Existing Domain” on page 495-2.
2. Start the domain’s Administration Server.
3. If you want to run the Managed Server on a remote WebLogic Server host:

- a. Establish a session with a remote computer on which you have installed WebLogic Server. For example, you can use telnet or VNC to establish a connection with a remote WebLogic Server host.
- b. Create a directory or change to the directory that you want to use as the server's root directory. See "[A Server's Root Directory](#)."

- c. Copy the following script to the remote computer:

`domain-name\startManagedWebLogic.cmd` (Windows)

`domain-name/startManagedWebLogic.sh` (UNIX)

where *domain-name* is the directory in which you located the domain. By default, this directory is

`BEA_HOME\user_projects\domains\domain-name`

- d. In a text editor on the remote computer, open the `startManagedWebLogic` script.

- e. Edit the command that calls the WebLogic Server environment script (`WL_HOME\common\bin\commEnv.cmd` or `commEnv.sh`) so that the pathname matches the pathname on the remote computer.

For example, if you installed WebLogic Server on the D drive of the remote computer, make sure the `call` command specifies the D drive:

```
@rem set up common environment
call "D:\bea\weblogic\common\bin\commEnv.cmd"
```

- f. Save the modified `startManagedWebLogic` script.

4. In a shell (command prompt) on the computer that you want to host the Managed Server, change to the directory that contains the `startManagedWebLogic` script.

5. Enter one of the following commands:

- `startManagedWebLogic.cmd managed_server_name admin_url` (Windows)

- `startManagedWebLogic.sh managed_server_name admin_url` (UNIX)

where *managed_server_name* specifies the name of the Managed Server and *admin_url* specifies the listen address (host name or IP address) and port number of the domain's Administration Server.

For example, the following command uses `startManagedWebLogic.cmd` to start a Managed Server named `myManagedServer`. The listen address for the domain's Administration Server is `AdminHost:7001`:

1 *Starting and Stopping Servers*

```
c:\user_domains\mydomain\startManagedWebLogic.cmd myManagedServer  
http://AdminHost:7001
```

For more information on configuring a connection to the Administration Server, refer to “Configuring a Connection to the Administration Server” on page 497-22.

The WebLogic startup script does the following:

1. Sets environment variables by invoking `WL_HOME\common\bin\commEnv.cmd` (`commEnv.sh` on UNIX), where `WL_HOME` is the location in which you installed WebLogic Server.
2. Invokes the `java weblogic.Server` command, which starts a JVM that is configured to run a WebLogic Server instance.

When the server successfully completes its startup process, it writes the following message to standard out (which, by default, is the command window):

```
<Notice> <WebLogicServer> <000360> <Server started in RUNNING mode>
```

Alternate Ways to Start Managed Servers

The following sections describe alternate ways to start a Managed Server:

- “Starting All Managed Servers in a Domain” on page 497-11
- “Starting a Managed Server in the STANDBY State” on page 497-12
- “Creating Scripts That Use the Node Manager” on page 497-13
- “Starting a Managed Server With the `java weblogic.Server` Command” on page 497-13
- “Starting a Managed Server When the Host Computer Boots” on page 497-14
- “Starting a Managed Server If the Administration Server is Unavailable” on page 497-14

Starting All Managed Servers in a Domain

The Administration Console provides an operation that starts all Managed Servers that have been configured to communicate with a Node Manager.

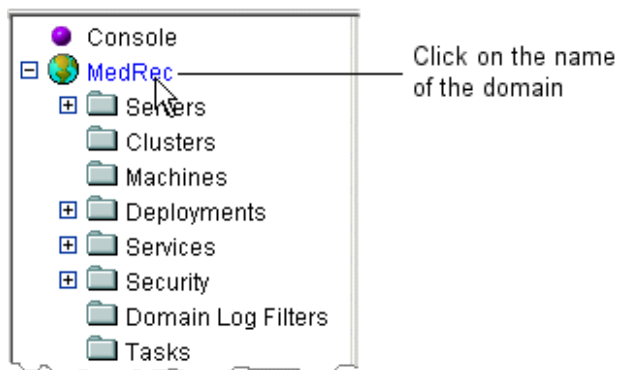
To start all Managed Servers in a domain.:

1. Configure the Managed Servers to communicate with a Node Manager. See "Configuring a Machine" on page 268-1.

The **Node Manager** is a standalone Java program provided with each WebLogic Server installation. You cannot use the Node Manager to start Administration Servers. See "[Overview of Node Manager](#)."

2. Start the Node Manager on all computers that host Managed Servers. For more information, refer to "[Starting Node Manager](#)."
3. Start the Administration Server.
4. In the left pane of the Administration Console, click on the name of the domain. (See Figure 497-4.)

Figure 497-4 Click on the Name of the Domain



5. In the right pane, select the Control tab.
6. On the Control tab, select Start all Managed Servers.
7. When the Administration Console prompts you to confirm the command, click Yes.

The Administration Console displays an animated status icon while the Node Manager starts each server on its target machine. (See Figure 497-2.)

Figure 497-5 Status Icon



Status icon

When the Node Manager finishes the start sequence for all servers, the status icon is no longer displayed and the state of each server is indicated in the Current Status table, under the State column.

8. To view messages that the Node Manager generated while starting the servers, click the Task Completed link in the Current Status table.

These messages are also written to the Node Manager log file for that server, as described in "[Managed Server Log Files](#)."

Starting a Managed Server in the STANDBY State

You can configure a Managed Server so that at the end of its startup cycle, the server is in the **STANDBY** state. In this state, the server listens for administrative requests only on the domain-wide administration port. When you are ready for the server to receive other types of requests on other listen ports, you resume it as described in "Resuming a Server" on page 497-24.

To configure a Managed Server so that it starts in the **STANDBY** state:

1. Start the domain's Administration Server.
2. Enable the domain-wide administration port. Refer to "Enabling the Domain-Wide Administration Port" on page 74-1.
3. In the left pane of the Administration Console, expand the Servers folder and select a server. (See Figure 497-1.)
4. In the right pane, select Configuration → General.

5. In the Advanced Options bar, click Show.
6. Under Advanced Options, in the Startup Mode list, select STANDBY.

The Startup Mode list determines the startup behavior for a server instance. If you select STANDBY, all future startup cycles for this server will end in the STANDBY state.
7. Click Apply.
8. Start the server as described in “Starting Managed Servers from the Administration Console” on page 497-5 or “Starting Managed Servers From a WebLogic Server Script” on page 497-8.

When you are ready for this server to receive non-administrative requests, refer to “Resuming a Server” on page 497-24.

Creating Scripts That Use the Node Manager

You can create your own scripts that use the Node Manager to start Managed Servers. The scripts must incorporate the `weblogic.Admin START` command. For more information on `weblogic.Admin` commands, refer to the “[weblogic.Admin Command-Line Reference](#).”

Starting a Managed Server With the `java weblogic.Server` Command

The `weblogic.Server` class is the main class for a WebLogic Server instance. You can start a server instance by directly invoking `weblogic.Server` in a Java command or by creating your own scripts that invoke the `weblogic.Server` class. (The scripts that WebLogic Server creates invoke `weblogic.Server` in a Java command.)

See “[weblogic.Server Command Line Reference](#).”

Starting a Managed Server When the Host Computer Boots

If you want a Managed Server to start automatically when you boot a computer, you can set up the server as a UNIX daemon or a Windows service. See "[Setting Up a WebLogic Server Instance as a Windows Service](#)."

Starting a Managed Server If the Administration Server is Unavailable

Usually, a Managed Server contacts the Administration Server during its startup sequence to retrieve its configuration information. For information on starting Managed Servers when the Administration Server is unavailable, refer to "[Starting a Managed Server When the Administration Server Is Not Available](#)."

Note: The first time you start a Managed Server, it must be able to contact the Administration Server. Thereafter you can configure Managed Servers to start even if the Administration Server is unavailable.

Providing Usernames and Passwords to Start and Stop a Server

To start and stop a WebLogic Server instance, you must provide the credentials of a user who is permitted to start and stop servers. For information on user credentials, roles, and permissions, refer to "[Security Roles](#)."

This section describes the following tasks:

- "Specifying an Initial Administrative User for a Domain" on page 497-15
- "Boot Identity Files" on page 497-15

- “Specifying User Credentials When Starting a Server with the Node Manager” on page 497-21

Specifying an Initial Administrative User for a Domain

When you create a domain, the Configuration Wizard prompts you to provide the username and password for an initial administrative user. The Configuration Wizard does the following with this information:

1. Assigns the user to the Administrators security group.

The Administrators group grants the highest level of privileges for starting and managing WebLogic Server. For information on administrative privileges, refer to "[Security Roles](#)."

2. Adds the user to the `myrealm` security realm.

A **security realm** is a collection of components (providers) that authenticate usernames, determine the type of resources that the user can access, and provide other security-related services for WebLogic resources. WebLogic Server installs the `myrealm` security realm and uses it by default.

You can use the Administration Console to add users to security realms. If you use an Authentication provider other than the one that WebLogic Server installs, you must use the provider's administration tools to create at least one user with administrative privileges.

3. If you are creating a domain in development mode, the wizard creates a boot identity file, which contains an encrypted version of the username and password. For more information, refer to "Boot Identity Files."

Boot Identity Files

A boot identity file is a text file that contains user credentials for starting and stopping an instance of WebLogic Server. An Administration Server can refer to this file for user credentials instead of prompting you to provide them. Because the credentials are encrypted, using a boot identity file is much more secure than storing unencrypted credentials in a startup or shutdown script.

If you start a Managed Server from a script that invokes the `java weblogic.Server` command (or if you invoke the `java weblogic.Server` command directly), a Managed Server can also refer to a boot identity file. However, if you use the Node Manager to start a Managed Server, the Managed Server does not refer to a boot identity file. Instead, it refers to user credentials that are encrypted and stored in the domain's configuration file (`config.xml`). For more information, refer to “Specifying User Credentials When Starting a Server with the Node Manager” on page 497-21.

The following sections describe working with boot identity files:

- “Creating a Boot Identity File for an Administration Server” on page 497-16
- “Creating a Boot Identity File for a Managed Server” on page 497-18
- “Using a Boot Identity File to Start a Server Instance” on page 497-18
- “Using a Boot Identity File to Stop a Server Instance” on page 497-20
- “Removing a Boot Identity File After Startup” on page 497-20

Creating a Boot Identity File for an Administration Server

If you use the Configuration Wizard to create a domain in development mode, the Configuration Wizard creates an encrypted boot identity file in the root directory of the Administration Server. For more information about root directories, refer to “[A Server's Root Directory](#).”

If a boot identity file for an Administration Server does not already exist, and if you want to bypass the prompt for username and password, create one as follows:

1. Start the Administration Server at least once and provide the user credentials on the command line.

During the Administration Server's initial startup process, it generates security files that must be in place before a server can use a boot identity file.

2. Place the following two lines in a text file:

```
username=username  
password=password
```

The username and password values must match an existing user account in the Authentication provider for the default security realm and must belong to a role that has permission to start and stop a server. For information on roles and permissions, refer to “[Security Roles](#).”

3. Save the file.

If you save the file as `boot.properties` and locate it in the server's root directory, the server automatically uses this file during its subsequent startup cycles. For more information, refer to "Using a Boot Identity File to Start a Server Instance" on page 497-18.

The first time you use this file to start a sever, the server reads the file and then overwrites it with an encrypted version of the username and password.

Alternative Technique for Creating a Boot Identity File for an Administration Server

If you invoke the `weblogic.Server` class directly on the command line, instead of following the steps in the previous section, you can create a boot identity file by including the following options in the Java command:

```
-Dweblogic.management.username=username  
-Dweblogic.management.password=password  
-Dweblogic.system.StoreBootIdentity=true
```

These options cause the server instance to boot with the supplied user credentials and then store them in a file named `boot.properties`.

For example, the following command starts an Administration Server named `myAdminServer` and creates a boot identity file:

```
java -Dweblogic.management.username=username  
-Dweblogic.management.password=password  
-Dweblogic.system.StoreBootIdentity=true  
-Dweblogic.Name=myAdminServer weblogic.Server
```

For more information about invoking the `weblogic.Server` class directly from a command line, refer to "[weblogic.Server Command-Line Reference](#)."

Note: If you use a script to start an Administration Server, BEA recommends that you do **not** use the technique described in this section for the following reasons:

- It requires you to store an unencrypted password in the startup script.
- Each time you run the script, the server boots with the supplied user credentials and then creates a new boot identity file.

Creating a Boot Identity File for a Managed Server

If a Managed Server uses the same root directory as the Administration Server, it can use the same boot properties file as the Administration Server. For information about a server's root directory, refer to "[A Server's Root Directory](#)."

If you use a Node Manager to start a Managed Server, you do not need to create a boot identity file. For more information, refer to "Configure Startup Arguments for Managed Servers" on page 497-7.

To create a boot identity file for a Managed Server:

1. Start the domain's Administration Server to make sure that the required security files are in the root directory of the Administration Server. If the files are not present, the Administration Server generates them.
2. Copy the `SerializedSystemIni.dat` file from the Administration Server's root directory to the Managed Server's root directory.
3. Place the following two lines in a text file:

```
username=username  
password=password
```

The username and password values must match an existing user account in the Authentication provider for the default security realm and must belong to a role that has permission to start a server. For information on roles and permissions, refer to "[Security Roles](#)."

4. Save the file.

If you save the file as `boot.properties` and locate it in the server's root directory, the server automatically uses this file during its subsequent startup cycles. For more information, refer to "Using a Boot Identity File to Start a Server Instance."

The first time you use this file to start a sever, the server reads the file and then overwrites it with an encrypted version of the username and password.

Using a Boot Identity File to Start a Server Instance

A server instance uses a boot identity file during its startup process as follows:

- If a server's root directory contains a valid `boot.properties` file, it uses this file during its startup process by default. For information about a server's root directory, refer to "[A Server's Root Directory](#)."
- If you want to specify a different file (or if you do not want to store boot identity files in a server's root directory), you can include the following argument in the server's `weblogic.Server` startup command:

```
-Dweblogic.system.BootIdentityFile=filename
```

where *filename* is the fully qualified pathname of a valid boot identity file.

To specify this argument in the `startWebLogic` script, add

`-Dweblogic.system.BootIdentityFile` as a value of the `JAVA_OPTIONS` variable. For example:

```
set
JAVA_OPTIONS=-Dweblogic.system.BootIdentityFile=C:\BEA\user_domains\mydomain\myidentity.prop
```

- If you do **not** want a server instance to use a boot identity file during its startup cycle, include the following options in the server's `weblogic.Server` startup command:

```
-Dweblogic.management.username=username
-Dweblogic.management.password=password
```

These options cause a server instance to ignore any boot identity files and override other startup options that cause a server to use boot identity files during its startup cycle.

Note: If you use a script to start a server instance, BEA recommends that you do **not** use this technique because it requires you to store an unencrypted password in the startup script. Use this technique only if you invoke the `weblogic.Server` class directly from the command line. For more information, see "[weblogic.Server Command-Line Reference](#)."

- If a server is unable to access its boot identity file during its startup cycle, it displays the username and password prompt in its command shell and writes a message to the log file.

For a given server instance, use only the boot identity file that the instance has created. WebLogic Server does not support copying a boot identity file from one server root directory to another.

For example, if you use ServerA to generate a boot identity file, use only that boot identity file with ServerA. Do not copy ServerA's boot identity file into the root directory of ServerB. Instead, create a boot identity file for ServerB as described in "Creating a Boot Identity File for an Administration Server" on page 497-16 or "Creating a Boot Identity File for a Managed Server" on page 497-18.

Using a Boot Identity File to Stop a Server Instance

The `weblogic.Admin SHUTDOWN` or `FORCESHUTDOWN` commands use a boot identity file as follows:

- If you invoke the `weblogic.Admin SHUTDOWN` or `FORCESHUTDOWN` command from a server's root directory, and if the server's root directory contains a valid `boot.properties` file, it uses this file by default. For information about a server's root directory, refer to "[A Server's Root Directory](#)."
- If you invoke the `weblogic.Admin SHUTDOWN` or `FORCESHUTDOWN` command from a server's root directory, but the server's boot identity file is not in the server's root directory or is not named `boot.properties`, include the following argument in the command:

```
-Dweblogic.system.BootIdentityFile=filename
```

where *filename* is the fully qualified pathname of a valid boot identity file.

- If you do not invoke the `weblogic.Admin SHUTDOWN` or `FORCESHUTDOWN` command from a server's root directory, include both of the following arguments in the command:

```
-Dweblogic.system.BootIdentityFile=filename
```

```
-Dweblogic.RootDirectory=path
```

where *filename* is the fully qualified pathname of a valid boot identity file and *path* is the relative or fully-qualified name of the server's root directory.

For a given server instance, use only the boot identity file that the instance has created. WebLogic Server does not support copying a boot identity file from one server root directory to another.

Removing a Boot Identity File After Startup

If you want to remove the boot identity file after a server starts, you can include the following argument in the server's `weblogic.Server` startup command:


```
-Dweblogic.system.RemoveBootIdentity=true
```

This argument removes only the file that the server used to start. For example, if you specify `-Dweblogic.system.BootIdentityFile=c:\secure\boot.MyServer`, only `boot.MyServer` is removed, even if the server's root directory contains a file named `boot.properties`.

To specify this argument in the `startWebLogic` script, add

```
-Dweblogic.system.RemoveBootIdentity=true
```

 as a value of the `JAVA_OPTIONS` variable. For example:

```
set JAVA_OPTIONS=-Dweblogic.system.RemoveBootIdentity=true
```

Specifying User Credentials When Starting a Server with the Node Manager

If you use the Node Manager to start a Managed Server, you must provide user credentials on the Remote Start tab of the Administration Console. If you do not provide these credentials, the Node Manager throws an exception when it tries to start the server.

When you use the Administration Console or the Configuration Wizard to create a Managed Server, WebLogic Server adds your credentials to the server's Remote Start tab.

If you want the server instance to run under a different WebLogic Server user account:

1. In the left pane of the Administration Console, expand the Servers folder and select a server.
2. In the right pane, select Configuration → Remote Start.
3. In the Username field, enter the name of an existing user.

The user must be in a role that has permission to start servers. See "[Security Roles](#)."

4. In the Password and Confirm Password fields, enter the password for the user that you specified.
5. Click Apply.
6. Restart the server in order to use these credentials.

Other Startup Tasks

The following sections describe miscellaneous startup tasks:

- “Configuring a Connection to the Administration Server” on page 497-22
- “Resuming a Server” on page 497-24
- “Specifying Java Options for a WebLogic Server Instance” on page 497-25
- “Changing the JVM that Runs Servers” on page 497-27

Configuring a Connection to the Administration Server

If you start a Managed Server from a script that invokes the `java weblogic.Server` command, or if you invoke the `java weblogic.Server` command directly, you must make sure that the Managed Server specifies the correct listen address of the Administration Server. A Managed Server uses this address to retrieve its configuration from the Administration Server.

Use the following format to specify the listen address:

```
[protocol://]Admin-host:port
```

1. For *protocol*, specify any of the following:

- `t3`
- `t3s`
- `http`
- `https`

If you do not specify a value, the servers use T3.

Note: Regardless of which protocol you use, the initial download of a Managed Server’s configuration is over HTTP or HTTPS. After the RMI subsystem initializes, the server instance can use the T3 or T3S protocol.

2. For *Admin-host*, specify any of the following by default:

- `localhost`.

Valid only if you are starting the Managed Server on the same computer as the Administration Server.

- The DNS name of the computer that is hosting the Administration Server.
- The IP address of the computer that is hosting the Administration Server.

Because of the following security issue, BEA System recommends that you do not use IP addresses for *Admin-host* in a production environment:

To connect to the Administration Server through an SSL port, the Managed Server verifies that the Administration Server's host name matches the host name that is specified in the URL. If you specify an IP address, and if host name verification is enabled, the connection fails because the IP address, which is a series of numbers, does not match the name of the host, which is a string of characters.

In a development environment, where security is less of a concern, you can disable host name verification on the Managed Server so SSL connections that specify an IP address will succeed. Refer to [“Using a Hostname Verifier.”](#)

If the Administration Server has been configured to use some other listen address, you must specify the configured listen address. See “Configuring the Listen Address” on page 495-11.

3. For *port*, specify any of the following:

- The domain-wide administration port.

If you have enabled the domain-wide administration port, you must specify this port. You must specify either the T3S or HTTPS protocol to use this port.

- The non-SSL listen port for the Administration Server's default network configuration (7001 by default).

If this listen port has been disabled for the Administration Server, you must use one of the other listen ports described in this list. You must specify either the T3 or HTTP protocol to use this port.

- The SSL listen port for the Administration Server's default network configuration (7002 by default).

If this listen port has been disabled for the Administration Server, you must use one of the other listen ports described in this list. You must specify either the T3S or HTTPS protocol to use this port.

- The port number that is associated with an optional, custom network channel.

If the port is secured with SSL, you must specify either the T3S or HTTPS protocol.

4. To verify the host IP address, name, and default listen port of the Administration Server, start the Administration Server in a shell (command prompt). When the server successfully finishes its startup cycle, it prints to standard out messages that are similar to the following (among other messages):

```
<Apr 19, 2002 9:24:19 AM EDT> <Notice> <WebLogicServer>
<000355> <Thread "Listen Thread.Default" listening on port
7001, ip address 11.12.13.141>
```

...

```
<Apr 19, 2002 9:24:19 AM EDT> <Notice> <WebLogicServer>
<000331> <Started WebLogic Admin Server "MedRecServer" for
domain "MedRec" running in Development Mode>
```

For information on enabling SSL, refer to "[Configuring the SSL Protocol](#)." For more information on Administration Ports, refer to "Enabling the Domain-Wide Administration Port" on page 74-1.

Resuming a Server

If you have started a server in the `STANDBY` state, when you are ready for the server to receive requests other than administration requests:

1. In the left pane of the Administration Console, expand the Servers folder and select a server that is in the `STANDBY` state.
2. In the right pane, select Control →Start/Stop.
3. Select Resume this server, then click Yes to confirm and resume the server.

For information on how the server transitions from `STANDBY` to the `RUNNING` state, refer to "[RESUMING](#)."

Specifying Java Options for a WebLogic Server Instance

You use Java options to configure operating parameters for the JVM that runs a WebLogic Server instance. For example, you use Java options to tune the performance and monitoring capabilities of the JRockit JVM.

You can also use Java options to override a server's configuration temporarily. The Java options apply only to the current instance of the server. They are not saved in the domain's `config.xml` file and they are not visible from the Administration Console. For example, if a server is configured to listen on port 7201, you can use a Java option to start the server so that it listens on port 7555. The Administration Console will still indicate that the server is configured to listen on port 7201. If you do not use the Java option the next time you start the server, it will listen on port 7201.

The following sections describe how to specify Java options for the JVM that runs a WebLogic Server instance:

- “Specifying Java Options for a WebLogic Server Startup Script” on page 497-25
- “Specifying Java Options for a Managed Server that the Node Manager Starts” on page 497-26

Specifying Java Options for a WebLogic Server Startup Script

If you use a WebLogic Server script to start servers, do the following:

1. Create a backup copy of the WebLogic Server start scripts:
 - For scripts that start an Administration Server, back up
`domain-name\startWebLogic.cmd` (`startWebLogic.sh` on UNIX)
 - For scripts that start a Managed Server, back up
`domain-name\startManagedWebLogic.cmd` (`startManagedWebLogic.sh` on UNIX)

where *domain-name* is the directory in which you located the domain. By default, this directory is `BEA_HOME\user_projects\domains\domain-name`

2. Open the start scripts in a text editor.

3. Edit the `set JAVA_OPTIONS` command to specify the Java options. If you specify multiple options, separate each option by a space, and place quotes around the entire set of options. For example:

```
set JAVA_OPTIONS="-Xgc:gencopy -Xns:30"
```

For more information, refer to:

- [“weblogic.Server Command-Line Reference”](#) for information on the Java options that set runtime behavior of a WebLogic Server instance.
 - [“JRockit Java Virtual Machine User Guide”](#) for information on the Java options that the JRockit Virtual Machine supports.
 - The documentation that the JVM vendor provides for information on the Java options that other JVMs support.
4. Save the start script.
 5. Restart the server.

Specifying Java Options for a Managed Server that the Node Manager Starts

If you use the Node Manager to start Managed Servers, do the following for each server:

1. In left pane of the Administration Console, select the server.
2. In the right pane, select Configuration → Remote Start.
3. In the Arguments field, specify the Java options. If you specify multiple options, separate each option by a space.

For more information, refer to:

- [“weblogic.Server Command-Line Reference”](#) for information on the Java options that set runtime behavior of a WebLogic Server instance.
 - [“JRockit Java Virtual Machine User Guide”](#) for information on the Java options that the JRockit Virtual Machine supports.
 - The documentation that the JVM vendor provides for information on the Java options that other JVMs support.
4. Click Apply.
 5. Restart the Managed Server.

Changing the JVM that Runs Servers

When you create a domain, if you choose to customize the configuration, the Configuration Wizard presents a list of SDKs that WebLogic Server installed. From this list, you choose the JVM that you want to run your domain and the wizard configures the BEA start scripts based on your choice.

After you create a domain, if you want to use a different JVM, you can modify the scripts as follows:

1. To change the JVM for all server instances that run on a WebLogic Server host, open `WL_HOME\common\bin\commEnv.sh` where `WL_HOME` is the directory in which you installed WebLogic Server.

To change the JVM only for a specific domain's Administration Server, open `domain-name\StartWebLogic.cmd` (Windows) or `domain-name\StartWebLogic.sh` (UNIX).

To change the JVM only for a specific domain's Managed Servers, open `domain-name\StartManagedWebLogic.cmd` (Windows) or `domain-name\StartManagedWebLogic.sh` (UNIX).

where `domain-name` is the directory which you created the domain.

2. Change the value for the `JAVA_HOME` variable.

Specify an absolute pathname to the top directory of the SDK that you want to use. For example, `c:\bea\jrockit81`

On a Windows or Linux platform, BEA recommends the following JVMs:

- For development mode, the Sun SDK with the HotSpot Client JVM.
- For production mode, the WebLogic JRockit SDK. This SDK provides optimal running performance but initial startup cycles can require more time than other SDKs.

3. Change the value for the `JAVA_VENDOR` variable.

Specify the vendor of the SDK. Valid values depend on the platform on which you are running. For more information, refer to the Supported Platforms page at the following URL: <http://e-docs.bea.com/platform/docs81/support/index.html>.

For example:

- BEA indicates that you are using the JRockit SDK. It is valid only on platforms that support JRockit.
 - Sun indicates that you are using the Sun SDK.
 - HP and IBM indicate that you are using SDKs that Hewlett Packard or IBM have provided. These values are valid only on platforms that support HP or IBM SDKs.
4. Restart any servers that are currently running.

Shutting Down Instances of WebLogic Server

You can do any of the following to shut down a WebLogic Server instance:

- Using the Administration Console:
 - “Shutting Down a Server” on page 497-28
 - “Shutting Down All Managed Servers in a Domain” on page 497-29
- Using the `weblogic.Admin` utility:
 - [SHUTDOWN](#)
 - [FORCESHUTDOWN](#)
- “Killing the JVM” on page 497-31

Shutting Down a Server

To shut down a server from the Administration Console:

1. In the left pane, expand the Servers folder and select a server. (See Figure 497-1.)
2. In the right pane, select Control →Start/Stop.
3. Select one of the following:

- Shutdown this server.

This command initiates a graceful shutdown, which gives WebLogic Server subsystems time to complete certain application processing currently in progress. For more information, refer to “Controlling Graceful Shutdowns” on page 497-30.

- Force shutdown this server.

This command initiates a forced shutdown, in which the server instructs subsystems to immediately drop in-work requests. For more information, refer to “[Forced Shutdown](#).”

4. Click Yes to confirm and shut down the server.

If you shut down the Administration Server, the Administration Console is no longer active.

Shutting Down All Managed Servers in a Domain

To shut down all Managed Servers:

1. In the left pane of the Administration Console, click the name of the domain. (See Figure 497-4.)
2. In the right pane, select the Control tab.
3. Select one of the following:

- Graceful shutdown of all Managed Servers.

This command initiates a graceful shutdown of all Managed Servers, which causes each Managed Server to notify its subsystems to complete all in-work requests. A graceful shutdown gives WebLogic Server subsystems time to complete certain application processing currently in progress. For information, refer to “Controlling Graceful Shutdowns” on page 497-30 and “[Graceful Shutdown](#).”

- Force shutdown of all Managed Servers.

This command initiates a forced shut down. When you initiate a forced shutdown, each Managed Server instructs subsystems to immediately drop in-work requests. For more information, refer to “[Forced Shutdown](#).”

4. When the Administration Console prompts you to confirm the command, click Yes.
5. To confirm that all Managed Servers have been shut down, view the table at the bottom of the Control tab. The table displays a list of all servers and indicates their current state.

For information on shutting down the domain's Administration Server, refer to "Shutting Down a Server" on page 497-28.

Controlling Graceful Shutdowns

A graceful shutdown gives WebLogic Server subsystems time to complete certain application processing currently in progress. See "[Graceful Shutdown](#)."

To control the length of the graceful shutdown process:

1. In the left pane of the Administration Console, expand the Servers folder and select a server. (See Figure 497-1.)
2. In the right pane, select Control →Start/Stop.
3. If you want the server's graceful shutdown to drop all HTTP sessions immediately instead of waiting for them to complete or timeout, place a checkmark in the Ignore Sessions During Shutdown box.

Waiting for abandoned HTTP sessions to timeout can significantly lengthen the graceful shutdown process because the default session timeout is 24 hours.

4. The Graceful Shutdown Timeout field specifies a time limit for a server instance to complete a graceful shutdown. If you supply a timeout value, and the server instance does not complete a graceful shutdown within that period, WebLogic Server performs a forced shutdown on the server instance.
5. To limit the amount of time the server takes to complete a graceful shutdown, enter the maximum number of seconds in the Graceful Shutdown Timeout box.

If you supply a timeout value, and the server instance does not complete a graceful shutdown within that period, WebLogic Server performs a forced shutdown on the server instance.

If you do not supply a timeout value, the server waits indefinitely to complete a graceful shutdown.

Killing the JVM

Each WebLogic Server instance runs in its own JVM. If you are unable to shut down a server instance using the methods described in the previous sections, you can use an operating system command to kill the JVM.

Caution: If you kill the JVM, the server immediately stops all processing. Any session data is lost. If you kill the JVM for an Administration Server while the server is writing to the `config.xml` file, you can corrupt the `config.xml` file.

Some common ways to kill the JVM are as follows:

- If the shell (command prompt) in which you start the server is still open, you can type `Ctrl-C`.
- On a Windows computer, you can use the Task Manager to kill a JVM. If the server is running as a Windows service, you must use the Services Control Panel to kill the JVM.
- On a UNIX computer, you can use the `ps` command to list all running processes. Then you can use the `kill` command to kill the JVM.

SMNP --> Traps --> Attribute Change --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to configure Attribute Changes.

Tasks

“Configuring an Attribute Change” on page 516-15

Related Topics

[WebLogic SNMP Management Guide](#)

[WebLogic SNMP MIB Reference](#)

Attributes

Table 498-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this attribute change monitor.</p> <p>Each monitor within a domain must have a unique name.</p> <p>The name displays in the Administration Console, and if you use WebLogic Server command-line utilities or APIs, you use this name to identify the attribute change monitor.</p> <p>After you have created a attribute change monitor, you cannot change its name. Instead, clone the monitor and provide a new name for the clone. For more information, refer to “Cloning Configuration Objects” on page 6-11.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPAttributeChangeMBean</p> <p><i>Attribute:</i> Name</p>	
Attribute MBean Type	<p>The MBean type that defines the attribute you want to monitor.</p> <p>For more information, refer to “Determining the Names of Other Attributes” on page 516-10.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPAttributeChangeMBean</p> <p><i>Attribute:</i> AttributeMBeanType</p>	

Table 498-1

Attribute Label	Description	Value Constraints
Attribute MBean Name	<p>The name of the MBean instance that you want to monitor. If you leave this field empty, WebLogic Server monitors all instances of the MBean type that you specify in Monitored MBean Type.</p> <p>For more information, refer to “Determining the Names of Other Attributes” on page 516-10.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPGaugeMonitorMBean</p> <p><i>Attribute:</i> MonitoredMBeanName</p>	
Attribute Name	<p>The name of the MBean instance that contains the attribute you want to monitor.</p> <p>For more information, refer to “Determining the Names of Other Attributes” on page 516-10.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPAttributeChangeMBean</p> <p><i>Attribute:</i> AttributeName</p>	

Table 498-1

Attribute Label	Description	Value Constraints
Enabled Servers	<p>Defines a list of target servers for trap generation. If no server is specified, no trap will be generated.</p> <p>When you assign an Attribute Change monitor to a server, you deploy a JMX listener and filter to the server. The listener will forward a notification to the WebLogic SNMP agent only if an event matches the criteria in the Attribute Change monitor.</p> <p>For example, if you create a Attribute Change monitor that observes the <code>State</code> attribute of <code>ServerA</code>'s <code>ServerRuntimeMBean</code>, and if you target this monitor to <code>ServerB</code>, the monitor on <code>ServerB</code> will never generate a trap, because <code>ServerB</code> does not have access to the state of <code>ServerA</code>.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.SNMPCounterMonitorMBean</code></p> <p><i>Attribute:</i> <code>EnabledServers</code></p>	

SMNP --> Traps --> Attribute Change

WebLogic Server SNMP agents generate several types of trap notifications for SNMP managers. Attribute Change traps are generated when an attribute you select changes value.

To create new Attribute Changes, click the [Configure a new Attribute Change](#) link.

- For information on how to create an attribute change, see “[Configuring an Attribute Change](#)” on page 516-15.
- For information on how to clone an attribute change, see “[Cloning Configuration Objects](#)” on page 6-11.
- For information on how to delete an attribute change, see “[Deleting Configuration Objects](#)” on page 6-10.



SMNP --> Traps --> Counter Monitor --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to configure Counter Monitors, which periodically check the value of an integer attribute and generate a trap when the value exceeds a threshold.

You can also configure a Counter Monitor to increase the threshold by an offset value after it sends a trap.

Tasks

“Configuring a Counter Monitor” on page 516-20

Related Topics

[WebLogic SNMP Management Guide](#)

[WebLogic SNMP MIB Reference](#)

Attributes

Table 500-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this counter monitor.</p> <p>Each monitor within a domain must have a unique name.</p> <p>The name displays in the Administration Console, and if you use WebLogic Server command-line utilities or APIs, you use this name to identify the string monitor.</p> <p>After you have created a counter monitor, you cannot change its name. Instead, clone the counter monitor and provide a new name for the clone. For more information, refer to “Cloning Configuration Objects” on page 6-11.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPCounterMonitorMBean</p> <p><i>Attribute:</i> Name</p>	
Monitored MBean Type	<p>The MBean type that defines the attribute you want to monitor.</p> <p>For more information, refer to “Determining the Names of Other Attributes” on page 516-10.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPCounterMonitorMBean</p> <p><i>Attribute:</i> MonitoredMBeanType</p>	

Table 500-1

Attribute Label	Description	Value Constraints
Monitored MBean Name	<p>The name of the MBean instance that you want to monitor. If you leave this field empty, WebLogic Server monitors all instances of the MBean type that you specify in Monitored MBean Type.</p> <p>For more information, refer to “Determining the Names of Other Attributes” on page 516-10.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPGaugeMonitorMBean</p> <p><i>Attribute:</i> MonitoredMBeanName</p>	
Monitored Attribute Name	<p>The name of an attribute to monitor.</p> <p>For more information, refer to “Determining the Names of Other Attributes” on page 516-10.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPCounterMonitorMBean</p> <p><i>Attribute:</i> MonitoredAttributeName</p>	
Polling Interval	<p>The frequency (in seconds) that WebLogic Server checks the attribute value.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPCounterMonitorMBean</p> <p><i>Attribute:</i> PollingInterval</p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 65535</p>

Table 500-1

Attribute Label	Description	Value Constraints
Threshold	<p>Specifies a value that triggers the Counter Monitor to generate a trap.</p> <p>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.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.SNMPCounterMonitorMBean</code></p> <p><i>Attribute:</i> <code>Threshold</code></p>	<i>Minimum:</i> 0
Offset	<p>Specifies an integer value to be added to the threshold if the threshold is crossed.</p> <p>For example, if you set <code>Threshold</code> to 1000 and <code>Offset</code> 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.</p> <p>The threshold can increase up to a maximum specified by the <code>Modulus</code> attribute.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.SNMPCounterMonitorMBean</code></p> <p><i>Attribute:</i> <code>Offset</code></p>	<i>Minimum:</i> 0

Table 500-1

Attribute Label	Description	Value Constraints
Modulus	<p>Specifies the maximum value for the threshold.</p> <p>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.</p> <p>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.</p> <p>If you specify 0, the Counter Monitor does not use the modulus and the threshold value can grow indefinitely.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.SNMPCounterMonitorMBean</code></p> <p><i>Attribute:</i> <code>Modulus</code></p>	<i>Minimum:</i> 0



SMNP --> Traps --> Counter Monitor --> Servers

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to select servers from a list of available servers that will monitor the attribute specified in the Monitored Attribute Name field.

Tasks

“Configuring a Counter Monitor” on page 516-20

Related Topics

[WebLogic SNMP Management Guide](#)

[WebLogic SNMP MIB Reference](#)

Attributes

Table 501-1

Attribute Label	Description	Value Constraints
Enabled Servers	<p>Defines a list of target servers for trap generation. If no server is specified, no trap will be generated.</p> <p>When you assign a Counter Monitor to a server, you deploy a JMX listener and filter to the server. The listener will forward a notification to the WebLogic SNMP agent only if an event matches the criteria in the String Monitor.</p> <p>For example, if you create a Counter Monitor that observes the <code>AcceptBacklog</code> attribute of <code>ServerA</code>'s <code>Server MBean</code>, and if you target this monitor to <code>ServerB</code>, the monitor on <code>ServerB</code> will never generate a trap, because <code>ServerB</code> does not have access to the configuration data of <code>ServerA</code>.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.SNMPCounterMonitorMBean</code></p> <p><i>Attribute:</i> <code>EnabledServers</code></p>	

SMNP --> Traps --> Counter Monitor

The WebLogic SNMP agent allows you to configure Java Management Extension (JMX) monitors to poll WebLogic resources at a specified interval to check for the occurrence of conditions or the crossing of thresholds. A counter monitor defines a threshold that is an integer value. A trap is generated if the agent detects that attribute equals or exceeds the threshold value.

- For information on how to configure a counter monitor, see “Configuring a Counter Monitor” on page 516-20.
- For information on how to clone a counter monitor, see “Cloning Configuration Objects” on page 6-11.
- For information on how to delete a counter monitor, see “Deleting Configuration Objects” on page 6-10.



SMNP --> Traps --> Gauge Monitor --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to configure Gauge Monitors, which periodically check the value of an integer or floating-point attribute and generate a trap if the value exceeds a threshold.

Tasks

“Configuring a Gauge Monitor” on page 516-18

Related Topics

[WebLogic SNMP Management Guide](#)

[WebLogic SNMP MIB Reference](#)

Attributes

Table 503-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this gauge monitor.</p> <p>Each monitor within a domain must have a unique name.</p> <p>The name displays in the Administration Console, and if you use WebLogic Server command-line utilities or APIs, you use this name to identify the gauge monitor.</p> <p>After you have created a gauge monitor, you cannot change its name. Instead, clone the gauge monitor and provide a new name for the clone. For more information, refer to “Cloning Configuration Objects” on page 6-11.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPGaugeMonitorMBean</p> <p><i>Attribute:</i> Name</p>	
Monitored MBean Type	<p>The MBean type that defines the attribute you want to monitor.</p> <p>For more information, refer to “Determining the Names of Other Attributes” on page 516-10.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPGaugeMonitorMBean</p> <p><i>Attribute:</i> MonitoredMBeanType</p>	

Table 503-1

Attribute Label	Description	Value Constraints
Monitored MBean Name	<p>The name of the MBean instance that you want to monitor. If you leave this field empty, WebLogic Server monitors all instances of the MBean type that you specify in Monitored MBean Type.</p> <p>For more information, refer to “Determining the Names of Other Attributes” on page 516-10.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPGaugeMonitorMBean</p> <p><i>Attribute:</i> MonitoredMBeanName</p>	
Monitored Attribute Name	<p>The name of an attribute to monitor.</p> <p>For more information, refer to “Determining the Names of Other Attributes” on page 516-10.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPGaugeMonitorMBean</p> <p><i>Attribute:</i> MonitoredAttributeName</p>	
Polling Interval	<p>The frequency (in seconds) that WebLogic Server checks the attribute value.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPGaugeMonitorMBean</p> <p><i>Attribute:</i> PollingInterval</p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 65535</p>
Threshold High	<p>The high threshold at which a trap is generated. A trap is generated if the monitored value is equal to or greater than this value.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPGaugeMonitorMBean</p> <p><i>Attribute:</i> ThresholdHigh</p>	

Table 503-1

Attribute Label	Description	Value Constraints
Threshold Low	<p>The low threshold at which a trap is generated. A trap is generated if the monitored value is less than or equal to this value.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPGaugeMonitorMBean</p> <p><i>Attribute:</i> ThresholdLow</p>	

SNMP --> Traps --> Gauge Monitor --> Servers

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to select servers from a list of available servers that will monitor the attribute specified in the Monitored Attribute Name field.

Tasks

“Configuring a Gauge Monitor” on page 516-18

Related Topics

[WebLogic SNMP Management Guide](#)

[WebLogic SNMP MIB Reference](#)

Attributes

Table 504-1

Attribute Label	Description	Value Constraints
Enabled Servers	<p>Defines a list of target servers for trap generation. If no server is specified, no trap will be generated.</p> <p>When you assign a Gauge Monitor to a server, you deploy a JMX listener and filter to the server. The listener will forward a notification to the WebLogic SNMP agent only if an event matches the criteria in the String Monitor.</p> <p>For example, if you create a Gauge Monitor that observes the <code>AcceptBacklog</code> attribute of <code>ServerA</code>'s <code>Server MBean</code>, and if you target this monitor to <code>ServerB</code>, the monitor on <code>ServerB</code> will never generate a trap, because <code>ServerB</code> does not have access to the configuration data of <code>ServerA</code>.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.SNMPGaugeMonitorMBean</code></p> <p><i>Attribute:</i> <code>EnabledServers</code></p>	

SMNP --> Traps --> Gauge Monitor

The WebLogic SNMP agent allows you to configure Java Management Extension (JMX) monitors to poll WebLogic resources at a specified interval to check for the occurrence of conditions or the crossing of thresholds. A gauge monitor defines a high and a low threshold that is an integer value. A trap is generated if the agent detects that attribute is equal to or exceeds the high threshold or is equal to or less than the low threshold value.

- For information on how to configure a gauge monitor, see “Configuring a Gauge Monitor” on page 516-18.
- For information on how to clone a gauge monitor, see “Cloning Configuration Objects” on page 6-11.
- For information on how to delete a gauge monitor, see “Deleting Configuration Objects” on page 6-10.



SMNP --> Traps --> SNMP Monitors

The WebLogic SNMP agent allows you to configure Java Management Extension (JMX) monitors to poll WebLogic resources at a specified interval to check for the occurrence of conditions or the crossing of thresholds. When a user-defined monitor detects the specified condition, a trap notification is sent to the SNMP manager. This feature allows you to off load polling of WebLogic resources from the SNMP management station to the WebLogic Administration Server.

- For more information on gauge monitors, see “Configuring a Gauge Monitor” on page 516-18.
- For more information on string monitors, see “Configuring a String Monitor” on page 516-16.
- For more information on counter monitors, see “Configuring a Counter Monitor” on page 516-20.



SMNP --> Traps --> Log Filter --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to set up the WebLogic SNMP agent to listen for log messages that a server instance broadcasts. When the agent receives a message, it generates an SNMP log notification trap.

Tasks

“Configuring a Notification Log Filter” on page 516-14

Related Topics

[WebLogic SNMP MIB Reference](#)

[WebLogic SNMP Management Guide](#)

Attributes

Table 507-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this log filter.</p> <p>Each log filter within a domain must have a unique name.</p> <p>The name displays in the Administration Console, and if you use WebLogic Server command-line utilities or APIs, you use this name to identify the log filter.</p> <p>After you have created a log filter, you cannot change its name. Instead, clone the filter and provide a new name for the clone. For more information, refer to “Cloning Configuration Objects” on page 6-11.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPLogFilterMBean</p> <p><i>Attribute:</i> Name</p>	

Table 507-1

Attribute Label	Description	Value Constraints
Severity Level	<p>The minimum severity of a message that causes WebLogic Server to generate a trap. The ascending order of severities:</p> <ul style="list-style-type: none">■ 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. A warning message. A suspicious operation or configuration has occurred which may not affect the normal operation of 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 particular 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. <p>If you enter INFO, then WebLogic Server can generate a trap when a server instance prints a message of any severity.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPLogFilterMBean</p> <p><i>Attribute:</i> SeverityLevel</p>	<p><i>Default:</i> WARNING</p>

Table 507-1

Attribute Label	Description	Value Constraints
Subsystem Names	<p>Defines a list of subsystems whose messages are selected by this filter.</p> <p>If none are specified, then WebLogic Server can generate a trap when any subsystem prints a message.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPLogFilterMBean</p> <p><i>Attribute:</i> SubsystemNames</p>	
User Ids	<p>Defines a list of user IDs that causes WebLogic Server to generate a trap. Every message includes the user ID from the security context in which the message was generated.</p> <p>If the user ID field for a message matches one of the user IDs you specify in the filter, WebLogic Server generates a trap.</p> <p>If no IDs are specified, WebLogic Server can generate a trap for messages from all user IDs.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPLogFilterMBean</p> <p><i>Attribute:</i> UserIds</p>	

Table 507-1

Attribute Label	Description	Value Constraints
Message Ids	<p>Defines a list of message IDs or ID ranges that cause WebLogic Server to generate a trap.</p> <p>To see the ID of a message, view a server's message log. For more information, refer to "Viewing Server Logs from the Administration Console" on page 253-9.</p> <p>If no IDs are specified, this filter selects all message IDs.</p> <p>Example list: 20,50-100,300</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPLogFilterMBean</p> <p><i>Attribute:</i> MessageIds</p>	
Message Substring	<p>Defines a string that is searched for in the message text. Only messages that contain the string are selected. If a string is not specified, all messages are selected.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPLogFilterMBean</p> <p><i>Attribute:</i> MessageSubstring</p>	
Enabled Servers	<p>Defines a list of servers whose messages can cause WebLogic Server to generate a trap.</p> <p>If no server is specified, no traps will be generated.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPLogFilterMBean</p> <p><i>Attribute:</i> EnabledServers</p>	



SMNP --> Traps --> Log Filter

The WebLogic SNMP agent can register a log message filter which notifies the agent of selected log information.

To create a new Log Filter, click the Configure a new Log Filter link.

- For information on how to create a log filter, see “Configuring a Notification Log Filter” on page 516-14.
- For information on how to clone a log filter, see “Cloning Configuration Objects” on page 6-11.
- For information on how to delete a log filter, see “Deleting Configuration Objects” on page 6-10.



SMNP --> Proxies --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The WebLogic SNMP agent can act as a proxy for other SNMP agents. It listens for requests from SNMP managers. If any of the OIDs in the requests fall under the control of a proxy that you have defined, WebLogic Server forwards the request to the SNMP agent that is associated with the proxy.

Use this tab to provide the connection information and the OIDs that WebLogic Server uses to forward SNMP manager requests to another SNMP agent.

Tasks

“Configuring an SNMP Proxy” on page 516-23

Related Topics

[WebLogic SNMP Management Guide](#)

[WebLogic SNMP MIB Reference](#)

Attributes

Table 509-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this proxy configuration.</p> <p>This should be descriptive of the agent that the requests will be forwarded to, such as “SQLDBAgent.”</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPProxyMBean</p> <p><i>Attribute:</i> Name</p>	
Port	<p>The port number on which the proxied SNMP agent is listening.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPProxyMBean</p> <p><i>Attribute:</i> Port</p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 65535</p>
Oid Root	<p>The root of the OID tree that the proxied SNMP agent controls.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPProxyMBean</p> <p><i>Attribute:</i> OidRoot</p>	
Community	<p>The community name to be passed on for all requests to the proxied agent.</p> <p>The default value, na, causes WebLogic Server to pass the community name that is contained in incoming SNMP requests.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPProxyMBean</p> <p><i>Attribute:</i> Community</p>	<p><i>Default:</i> "na"</p>

Table 509-1

Attribute Label	Description	Value Constraints
Timeout	<p>The number of milliseconds that WebLogic Server waits for a response to requests forwarded to another SNMP agent. If the interval elapses without a response, WebLogic Server sends an error to the requesting manager.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPProxyMBean</p> <p><i>Attribute:</i> Timeout</p>	<p><i>Units:</i> milliseconds</p> <p><i>Minimum:</i> 0</p> <p><i>Default:</i> 5000</p>



SMNP --> Proxies

WebLogic Server uses an SNMP master agent that acts as a proxy for other SNMP agents.

To create a new proxy, click the Configure a new Proxy link.

- For information on how to create a proxy, see “Configuring an SNMP Proxy” on page 516-23
- For information on how to clone a proxy, see “Cloning Configuration Objects” on page 6-11
- For information on how to delete a proxy, see “Deleting Configuration Objects” on page 6-10



SMNP --> Traps --> String Monitor --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to configure String Monitors, which periodically check the value of a `String` attribute and generate a trap if the value is equal to or different from another specified string.

Tasks

“Configuring a String Monitor” on page 516-16

Related Topics

[WebLogic SNMP Management Guide](#)

[WebLogic SNMP MIB Reference](#)

Attributes

Table 511-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this string monitor.</p> <p>Each monitor within a domain must have a unique name.</p> <p>The name displays in the Administration Console, and if you use WebLogic Server command-line utilities or APIs, you use this name to identify the string monitor.</p> <p>After you have created a string monitor, you cannot change its name. Instead, clone the string monitor and provide a new name for the clone. For more information, refer to “Cloning Configuration Objects” on page 6-11.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPStringMonitorMBean</p> <p><i>Attribute:</i> Name</p>	
Monitored MBean Type	<p>The MBean type that defines the attribute you want to monitor.</p> <p>For more information, refer to “Determining the Names of Other Attributes” on page 516-10.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPStringMonitorMBean</p> <p><i>Attribute:</i> MonitoredMBeanType</p>	

Table 511-1

Attribute Label	Description	Value Constraints
Monitored MBean Name	<p>The name of the MBean instance that you want to monitor. If you leave this field empty, WebLogic Server monitors all instances of the MBean type that you specify in Monitored MBean Type.</p> <p>For more information, refer to “Determining the Names of Other Attributes” on page 516-10.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPGaugeMonitorMBean</p> <p><i>Attribute:</i> MonitoredMBeanName</p>	
Monitored Attribute Name	<p>The name of an attribute to monitor.</p> <p>For more information, refer to “Determining the Names of Other Attributes” on page 516-10.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPStringMonitorMBean</p> <p><i>Attribute:</i> MonitoredAttributeName</p>	
Polling Interval	<p>The frequency (in seconds) that WebLogic Server checks the attribute value.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPStringMonitorMBean</p> <p><i>Attribute:</i> PollingInterval</p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 65535</p>
String To Compare	<p>Defines a string to be compared to the value of the Monitored Attribute Name. A trap is generated when the criteria specified by Notify Match or Notify Differ is satisfied.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPStringMonitorMBean</p> <p><i>Attribute:</i> StringToCompare</p>	

Table 511-1

Attribute Label	Description	Value Constraints
Notify Differ	<p>Defines that the criteria used to generate a trap is that the value of Monitored Attribute Name and the value of String to Compare do not match.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPStringMonitorMBean</p> <p><i>Attribute:</i> NotifyDiffer</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false
Notify Match	<p>Defines that the criteria used to generate a trap is that the value of Monitored Attribute Name and the value of String to Compare match.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPStringMonitorMBean</p> <p><i>Attribute:</i> NotifyMatch</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false

SMNP --> Traps --> String Monitor --> Servers

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to select servers from a list of available servers that will monitor the attribute specified in the Monitored Attribute Name field.

Tasks

“Configuring a String Monitor” on page 516-16

Related Topics

[WebLogic SNMP Management Guide](#)

[WebLogic SNMP MIB Reference](#)

Attributes

Table 512-1

Attribute Label	Description	Value Constraints
Enabled Servers	<p>Defines a list of target servers for trap generation. If no server is specified, no trap will be generated.</p> <p>When you assign a String Monitor to a server, you deploy a JMX listener and filter to the server. The listener will forward a notification to the WebLogic SNMP agent only if an event matches the criteria in the String Monitor.</p> <p>For example, if you create a String Monitor that observes the <code>State</code> attribute of <code>ServerA</code>'s <code>ServerRuntime</code> MBean, and if you target this monitor to <code>ServerB</code>, the monitor on <code>ServerB</code> will never generate a trap, because <code>ServerB</code> does not have access to the state of <code>ServerA</code>.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.SNMPStringMonitorMBean</code></p> <p><i>Attribute:</i> <code>EnabledServers</code></p>	

SMNP --> Traps --> String Monitor

The WebLogic SNMP agent allows you to configure Java Management Extension (JMX) monitors to poll WebLogic resources at a specified interval to check for the occurrence of conditions or the crossing of thresholds. A string monitor does a compare between a string you provide and the value of a chosen attribute. A trap is generated if the agent detects that attribute matches the string you provide or you can specify that the trap is generated if the attribute does not match the string you provide.

- For information on how to configure a string monitor, see “Configuring a String Monitor” on page 516-16.
- For information on how to clone a string monitor, see “Cloning Configuration Objects” on page 6-11.
- For information on how to delete a string monitor, see “Deleting Configuration Objects” on page 6-10.



SNMP --> Trap Destinations --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to provide the information that WebLogic Server needs to connect with an SNMP manager.

Tasks

“Creating a Trap Destination” on page 516-4

Related Topics

[WebLogic SNMP Management Guide](#)

[WebLogic SNMP MIB Reference](#)

Attributes

Table 514-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this Trap Destination.</p> <p>This value is for your identification purposes only.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPTrapDestinationMBean</p> <p><i>Attribute:</i> Name</p>	
Community	<p>The SNMP trap community name. The community name functions as a password for sending trap notifications to the target SNMP manager.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPTrapDestinationMBean</p> <p><i>Attribute:</i> Community</p>	<i>Default:</i> "public"
Host	<p>The DNS name or IP address of the computer on which the SNMP manager is running.</p> <p>The WebLogic SNMP agent sends trap notifications to the host and port that you specify on this tab.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPTrapDestinationMBean</p> <p><i>Attribute:</i> Host</p>	<i>Default:</i> localhost

Table 514-1

Attribute Label	Description	Value Constraints
Port	<p>The port (1 - 65535) on which the SNMP manager is listening.</p> <p>The WebLogic SNMP agent sends trap notifications to the host and port that you specify on this tab.</p> <p>To determine the correct value for this attribute, refer to the settings of your SNMP manager.</p> <p><i>MBean:</i> weblogic.management.configuration.SNMPTrapDestinationMBean</p> <p><i>Attribute:</i> Port</p>	<p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 65535</p> <p><i>Default:</i> 162</p>



SNMP --> Trap Destinations

WebLogic Server uses Trap Destination to specify the SNMP management station and the community name used by the SNMP agent to send trap notifications.

To create a new Trap Destination, click the [Configure a new Trap Destination](#) link.

- For information on how to create a trap destination, see “Creating a Trap Destination” on page 516-4
- For information on how to clone a trap destination, see “Cloning Configuration Objects” on page 6-11
- For information on how to delete a trap destination, see “Deleting Configuration Objects” on page 6-10



1

2

3

4 Configuring SNMP and WebLogic Server

[“Attributes and Console Screen Reference for SNMP” on page 517-1]

WebLogic Server software includes the ability to communicate with enterprise-wide management systems using Simple Network Management Protocol (SNMP). The WebLogic Server SNMP capability enables you to integrate management of WebLogic Servers into an SNMP-compliant management system. Such a system provides a single view of the various software and hardware resources of a complex, distributed system.

The following sections describe using SNMP with WebLogic Server:

- “Configuring SNMP and WebLogic Server: Main Steps” on page 516-2
- “Enabling and Configuring the WebLogic SNMP Agent” on page 516-3
- “Creating a Trap Destination” on page 516-4
- “Determining Which WebLogic Server Attributes to Monitor” on page 516-6
- “Configuring WebLogic Server to Generate Traps” on page 516-12
- “Disabling Trap Generation” on page 516-22
- “Configuring an SNMP Proxy” on page 516-23

For background information on the WebLogic SNMP agent and WebLogic SNMP traps, refer to the following sections:

- ["Introduction to the WebLogic SNMP Agent"](#)
- ["WebLogic Trap Notifications"](#)

Configuring SNMP and WebLogic Server: Main Steps

To configure SNMP and WebLogic Server:

1. In your SNMP management system, load the WebLogic MIB. WebLogic Server installs its MIB as `WL_HOME\server\lib\BEA-WEBLOGIC-MIB.asn1`, where `WL_HOME` is the directory in which you installed WebLogic Server.

For information about loading a MIB, refer to the documentation that the vendor of your SNMP management system supplies.
2. Start the Administration Server for your domain.
3. Enable and configure the WebLogic SNMP agent. For more information, refer to “Enabling and Configuring the WebLogic SNMP Agent” on page 516-3.
4. If you want WebLogic Server to monitor some of its attributes and send traps to SNMP managers when the attribute values change, do the following:
 - a. Configure an SNMP trap destination. For more information, refer to “Creating a Trap Destination” on page 516-4.
 - b. Determine the names of the WebLogic Server attributes that you want to monitor. For more information, refer to “Determining Which WebLogic Server Attributes to Monitor” on page 516-6.
 - c. Configure WebLogic Server to generate one or more traps. For more information, refer to “Configuring WebLogic Server to Generate Traps” on page 516-12.

Enabling and Configuring the WebLogic SNMP Agent

In a WebLogic domain, the Administration Server hosts the SNMP agent. After you enable and configure the SNMP agent functionality, SNMP managers can send requests to the WebLogic SNMP agent. In addition, the WebLogic SNMP agent can be configured to gather and send data (trap notifications) about managed resources to your SNMP manager (trap destination).

To enable and configure the SNMP agent:

1. In the WebLogic Server Administration Console, in the left pane, expand the Services folder. Then click on the name of the SNMP folder.
2. In the right pane, on the SNMP tab, check the Enabled box.
3. At a minimum, make the following changes to the default values:
 - a. In the SNMP Port field, enter the port number on which the WebLogic SNMP agent listens for requests from SNMP managers.

Most SNMP managers can ping SNMP agents and some SNMP managers can request the status of specific attributes.

If an SNMP manager is running on the same computer as the Administration Server, make sure that the listen port you specify in this field and the listen port that you specify in the trap destination are different. (The trap destination's listen port specifies the port on which the SNMP **manager** listens for trap notifications.)

- b. In the Community Prefix field, enter the SNMP community (password) that SNMP managers must specify when sending requests to the WebLogic SNMP agent.

To secure access to the values of the WebLogic attributes, BEA Systems recommends that you use some value other than the default `public`.

For more information about the community prefix, refer to "[Using Community Names to Specify Target Servers in Management Requests](#)."

- c. From the Trap Version list, specify whether you want WebLogic Server to generate traps that conform to the SNMPv1 or SNMPv2 protocol. Choose a protocol that matches the capabilities of your SNMP managers.
4. Modify any of the default values for the remainder of attributes. For information about these attributes, refer to “Attributes” on page 69-2.
5. Click Apply.
6. Restart the Administration Server.

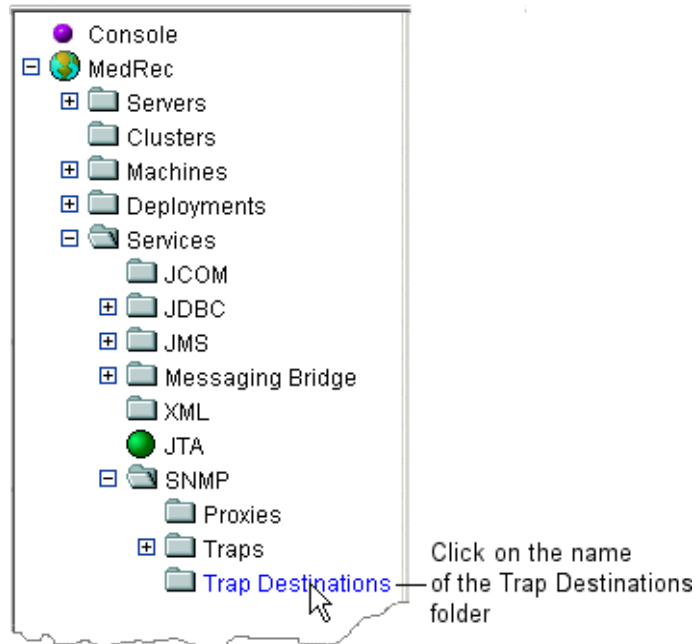
Creating a Trap Destination

A Trap Destination provides the information that the WebLogic SNMP agent needs to send trap notifications to an SNMP manager. For information about cloning or deleting a trap destination, refer to “Cloning Configuration Objects” on page 6-11 and “Deleting Configuration Objects” on page 6-10.

To create a Trap Destination:

1. In the Administration Console, in the left pane, expand the Services folder. Then expand the SNMP folder.
2. Click on the name of the Trap Destinations folder. (See Figure 516-1.)

Figure 516-1 Click on Trap Destinations

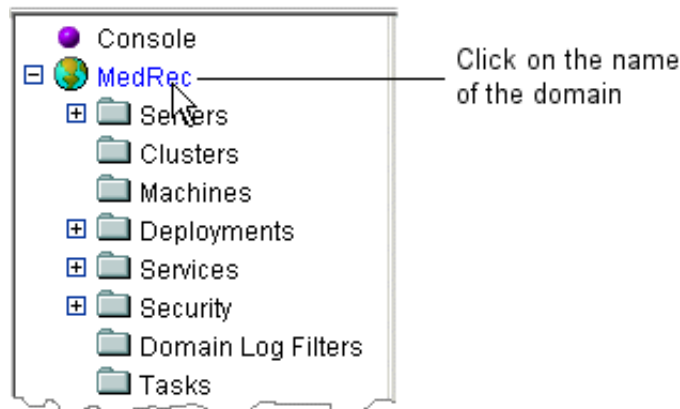


3. In the right pane, click the Configure a new SNMP Trap Destination link.
4. On the Create a new SNMPTrapDestination page, enter values in the attribute fields.

Make sure that the community name and port number match the name and number that your SNMP manager specifies, and that the host name correctly specifies the SNMP manager's host. For more information, refer to "Attributes" on page 514-2.

5. Click the Create button.
6. In the left pane, click on the name of the domain. (See Figure 516-2.)

Figure 516-2 Click on the Name of the Domain



7. In the right pane, click the Configuration tab. Then click the SNMP tab.
8. On the SNMP tab, in Targeted Trap Destinations, move the Trap Destination that you created from the Available to the Chosen column. Then click Apply.
9. Restart the Administration Server.

Determining Which WebLogic Server Attributes to Monitor

WebLogic Server provides information to your SNMP manager by monitoring server attributes and generating traps when the values change. WebLogic Server exposes these server attributes through Managed Beans (MBeans), which are part of the Java Management Extensions (JMX) specification. For more information, refer to ["WebLogic Server Managed Resources and MBeans."](#)

For any attribute that you want to monitor, you need the following information:

- The type of MBean in which the attribute is defined.
- The name of the attribute.

- Depending on the scope of the SNMP monitor, you might also need the name of a specific MBean instance in which the attribute resides.

The following sections describe how to determine the values that you must supply to configure WebLogic Server to generate SNMP traps:

- “Determining the Scope of an SNMP Monitor” on page 516-7
- “Commonly Monitored Attributes” on page 516-7
- “Determining the Names of Other Attributes” on page 516-10

Determining the Scope of an SNMP Monitor

You can configure WebLogic Server to observe an attribute in all MBeans of a specific type or to observe an attribute of only a single instance of an MBean.

For example, you can create a monitor that observes the runtime state of all server instances in a domain. If the state of any server changes to a specific value (such as `ShuttingDown`), the WebLogic SNMP agent generates a trap. The trap specifies the name of the server instance whose state has changed to `ShuttingDown`.

Or, you can create a monitor that observes the heap size of a specific JVM that is running a specific server instance. If the heap size grows beyond a specified value, the WebLogic SNMP agent generates a trap. You could create one of these monitors for each server instance in a domain and specify a different threshold for each. For example, `ServerA` might run on a computer that has more physical memory than `ServerB`, so you need different thresholds for each server instance.

If you want to monitor an attribute in a specific MBean instance, you must know the name of the MBean instance. The following sections provide information on determining the names of MBean instances.

Commonly Monitored Attributes

The following attributes are commonly monitored by SNMP and other management systems:

Table 516-3 Commonly Monitored WebLogic Server Attributes

Name	Description
<p>MBean Type: <code>ServerRuntime</code> Attribute Name: <code>State</code> MBean Instance Name: <code>server-name</code> For example, <code>MedRecServer</code></p>	<p>Indicates whether the server is in an Initializing, Suspended, Running, or ShuttingDown state.</p> <p>If you want to use one monitor for multiple server instances, do not supply a value for MBean instance name. If any the state of any server changes, WebLogic Server generates a trap.</p>
<p>MBean Type: <code>ExecuteQueueRuntime</code> Attribute Name: <code>ExecuteThreadCurrentIdleCount</code> MBean Instance Name: <code>default</code></p>	<p>Displays the number of threads in a server's default execute queue that are taking up memory space but are not being used to process data.</p> <p>You can create multiple execute queues on a server instance to optimize the performance of critical applications, but the default execute queue is available by default. For more information, refer to "Using Execute Queues to Control Thread Usage."</p>
<p>MBean Type: <code>ExecuteQueueRuntime</code> Attribute Name: <code>PendingRequestCurrentCount</code> MBean Instance Name: <code>default</code></p>	<p>Displays the number of requests waiting in a server's default execute queue.</p>
<p>MBean Type: <code>JVMRuntime</code> Attribute Name: <code>HeapSizeCurrent</code> MBean Instance Name: <code>server-name</code> For example, <code>MedRecServer</code></p>	<p>Displays the amount of memory (in bytes) that is currently available in the server's JVM heap.</p> <p>If you want to use one monitor for the heap size of multiple servers, do not supply a value for MBean instance name. If any server passes the threshold, WebLogic Server generates a trap.</p> <p>For more information, refer to "Tuning Java Virtual Machines (JVMs)."</p>

4 Configuring SNMP and WebLogic Server

Table 516-3 Commonly Monitored WebLogic Server Attributes

Name	Description
MBean Type: <code>ServerRuntime</code> Attribute Name: <code>OpenSocketsCurrentCount</code> MBean instance name: <i>server-name</i> For example, <code>MedRecServer</code>	Use these two attributes together to compare the current activity on the server's listen ports to the total number of requests that can be backlogged on the ports. Note that the attributes are located in two separate MBeans:
MBean Type: <code>Server</code> Attribute Name: <code>AcceptBacklog</code> MBean Instance Name: <i>server-name</i> For example, <code>MedRecServer</code>	<ul style="list-style-type: none">■ <code>OpenSocketsCurrentCount</code> is in the <code>ServerRuntime</code> MBean.■ <code>AcceptBacklog</code> is in the <code>Server</code> configuration MBean.
MBean Type: <code>JDBCConnectonPoolRuntime</code> Attribute Name: <code>ActiveConnectionsCurrentCount</code> MBean Instance Name: <i>connection-pool-name</i> For example, <code>MedRecPool</code>	Displays the current number of active connections in a JDBC connection pool. If you want to use one monitor for all JDBC Connection Pools in a domain, do not supply a value for MBean instance name. For more information, refer to " How JDBC Connection Pools Enhance Performance ."
MBean Type: <code>JDBCConnectonPoolRuntime</code> Attribute Name: <code>ActiveConnectionsHighCount</code> MBean Instance Name: <i>connection-pool-name</i> For example, <code>MedRecPool</code>	The high water mark of active connections in a JDBC connection pool. The count starts at zero each time the connection pool is instantiated.
MBean Type: <code>ServletRuntime</code> Attribute Name: <code>InvocationTotalCount</code> MBean Instance Name: <i>null</i> (do not specify a value)	The number of times each servlet has been invoked. Use this attribute to generate a trap when any servlet has been invoked more than a specific number of times. For example, if any servlet is invoked more than 10,000 times, the WebLogic SNMP agent generates a trap that indicates which servlet has been invoked over the threshold. WebLogic Server does not support monitoring specific servlets. For example, you cannot create a monitor that generates a trap only when a servlet named <code>MyServlet</code> is invoked beyond a threshold.

Determining the Names of Other Attributes

Most, but not all, WebLogic Server attributes can be monitored through SNMP. The WebLogic Server MIB lists all attributes that can be monitored through SNMP.

To determine the name of any attribute that can be monitored through SNMP:

1. Determine the name of the MBean type and attribute name. For more information, refer to “Determining the MBean Type and Attribute Name” on page 516-10.
2. If you want to monitor a specific instance of an MBean, determine the name of the MBean instance that exposes the attribute you want to monitor. For more information, refer to “Determining the Name of the MBean Instance” on page 516-11.

Determining the MBean Type and Attribute Name

To determine the MBean type and attribute name:

1. Browse through the [WebLogic Server MIB Reference](#) until you find an attribute that you want to monitor.

Within the MIB, tables that include the word *Runtime* in their title expose attributes that describe a server’s runtime state. All other tables expose attributes describe a server’s configuration. The values of the configuration attributes are all saved in a domain’s `config.xml` file.

2. To determine the corresponding MBean type and attribute name:
 - a. Determine the name of the MIB table in which the entry resides.
 - b. Remove any spaces and the word "table" from the table title to determine the name of the corresponding MBean type.
 - c. Remove the name of the table from the entry to determine the name of the MBean attribute.

For example:

- Under `serverRuntimeTable`, the `serverRuntimeState` entry corresponds to the `State` attribute of the `ServerRuntime` MBean, which indicates the current state of a server instance.

- Under `serverTable`, the `serverAcceptBacklog` entry corresponds to the `AcceptBacklog` attribute of the `Server MBean`, which is the total number of requests that can be backlogged on the server's ports.

Determining the Name of the MBean Instance

To determine the name of the MBean instance in which the attribute resides:

1. In a command prompt, run the following script:

```
WL_HOME\server\bin\setWLSEnv.cmd (Windows)
WL_HOME/server/bin/setWLSEnv.sh (UNIX)
```

where `WL_HOME` is the directory in which you installed WebLogic Server.

2. Enter the following command:

```
java weblogic.Admin -adminurl AdminHost:ListenPort
-username username -password password
GET -pretty -type MBean-type -property Attribute-Name
```

where:

- `-adminurl` specifies the listen address and listen port of the domain's Administration Server.
- `-username` and `-password` specify the credentials of a user with administrative privileges.
- `-type` specifies the MBean type that you determined in "Commonly Monitored Attributes" on page 516-7 or "Determining the Names of Other Attributes" on page 516-10.
- `-property` specifies the name of the MBean attribute that you want to monitor.

For example, the following command returns the names of all `JDBCConnectionPoolRuntime` MBeans that are currently instantiated in the domain:

```
java weblogic.Admin -adminurl localhost:7001
-username weblogic -password weblogic
GET -pretty -type JDBCConnectionPoolRuntime
-property ActiveConnectionsCurrentCount
```

This example command returns the following:


```
MBeanName:
```

```
"MedRec:Location=MedRecServer,Name=MedRecPool,ServerRuntime=Med  
RecServer,Type=JDBCConnectionPoolRuntime"
```

```
ActiveConnectionsCurrentCount: 1
```

3. Review the output of the `weblogic.Admin` command. The output provides the name of the MBean instance as `Name=name`.

In the previous example, the name of the `JDBCConnectionPoolRuntime` MBean is `MedRecPool`.

Configuring WebLogic Server to Generate Traps

A trap is the message (in a standard SNMP format) that WebLogic Server sends to the SNMP manager when an attribute changes in a specific way.

If you enable the SNMP service for a domain, the WebLogic SNMP agent automatically generates traps to notify an SNMP manager that a server instance has started. The WebLogic SNMP agent also automatically generates a trap if an SNMP manager sends an incorrect community string. For more information, refer to "[Automatically Generated WebLogic SNMP Traps](#)."

You can configure WebLogic Server to generate the following types of traps:

Table 516-4 Types of Traps

Configure This Type...	When You Want To...	For More Information, Refer To
Notification Log Filter	Generate a trap when a server instance prints a specific log message.	"Configuring a Notification Log Filter" on page 516-14

Table 516-4 Types of Traps

Configure This Type...	When You Want To...	For More Information, Refer To
Attribute Change	<p>Generate a trap when an attribute in a configuration MBean is changed.</p> <p>WebLogic Server provides two types of MBeans:</p> <ul style="list-style-type: none"> ■ Configuration MBeans, which contain static configuration data that changes only when you issue a command through the Administration Console, the <code>weblogic.Admin</code> utility, or an API. ■ Runtime MBeans, which contain dynamic information about an attribute's runtime state. <p>Runtime MBeans always include the word <code>Runtime</code> in their names. For example, the <code>ServerRuntime</code> MBean provides access to runtime attributes while the <code>Server</code> MBean provides access to configuration attributes.</p> <p>WebLogic Server does not support using Attribute Change trap notifications to monitor runtime attributes. Instead, use a String Monitor, Gauge Monitor, or Counter Monitor.</p>	"Configuring an Attribute Change" on page 516-15
String Monitor	<p>Periodically check the value of a <code>String</code> attribute and generate a trap if the value is equal to or different from another specified string.</p> <p>This monitor type can monitor both configuration and runtime MBeans.</p>	"Configuring a String Monitor" on page 516-16
Gauge Monitor	<p>Periodically check the value of an integer or floating-point attribute and generate a trap if the value exceeds a threshold.</p> <p>This monitor type can monitor both configuration and runtime MBeans.</p>	"Configuring a Gauge Monitor" on page 516-18

Table 516-4 Types of Traps

Configure This Type...	When You Want To...	For More Information, Refer To
Counter Monitor	<p>Periodically check the value of an integer attribute and generate a trap when the value exceeds a threshold.</p> <p>You can also configure a Counter Monitor to increase the threshold by an offset value after it sends a trap.</p> <p>This monitor type can monitor both configuration and runtime MBeans.</p>	“Configuring a Counter Monitor” on page 516-20

Configuring a Notification Log Filter

You can set up the WebLogic SNMP agent to listen for log messages that a server instance broadcasts. When the agent receives a message, it generates an SNMP log notification trap.

1. In the left pane of the Administration Console, expand the following folders: Services—SNMP—Traps. Then click on the name of the Log Filters folder.

The SNMP Log Filter page displays. It lists all of the SNMP filters that you have created in the current domain.
2. In the right pane, on the SNMP Log Filter page, click the Configure a new Log Filter link.
3. On the Create a new SNMP Log Filter page, enter values in the fields:
 - Use the Name field to specify the name of this log filter.
 - The other attribute fields correspond to fields within WebLogic Server log messages. For more information about these fields, refer to “Attributes” on page 507-2.

For example, any time you change the configuration of a server instance, the server prints a log message with a message ID of BEA-140009 and severity INFO. (You can see this message ID and severity level by changing a value in the Administration Console and then viewing the server’s log file. For more information, refer to “Viewing Server Logs” on page 253-9.)

If you want WebLogic Server to generate a trap when a server instance prints this log message, in the Severity field enter `INFO`, and in the Message ID field, enter `BEA-140009` or `140009`.

Or, if you want the SNMP agent to receive all log messages from the Security Service, in the Severity field enter `Info`, and in the Subsystem Names field enter `Security`.

4. Assign the filter to the server instances from which you want to receive messages.

The Enabled Servers, Available list contains all server instances that have been defined for the domain. Move the server instances from which you want to receive messages to the Chosen list.

5. Click the Create button to create and register the log filter.
6. Restart the Administration Server so that your changes can take effect.

Configuring an Attribute Change

An Attribute Change detects a change in a configuration attribute and sends a trap to the configured SNMP managers.

Note: WebLogic Server does not support using Attribute Change trap notifications to monitor run-time attributes. Runtime MBeans always include the word `Runtime` in their names. For example, the `ServerRuntime` MBean provides access to runtime attributes while the `Server` MBean provides access to configuration attributes. To monitor changes in an MBean that includes `Runtime` in its name, use a String Monitor, Gauge Monitor, or Counter Monitor.

To detect changes in configuration attributes:

1. In the Administration Console, in the left pane, expand the following folders: `Services`—`SNMP`—`Traps`. Then click on the name of the Attribute Changes folder.

The SNMP Attribute Change page displays. This page lists all the counter monitors that are configured for the domain.

2. On the SNMP Attribute Change page, select the Configure a new Attribute Change link.

3. On the Create a new SNMP Attribute Change page, enter a name for the attribute change in the Name field.

BEA Systems recommends that you choose a name that indicates the resource that is being monitored.

4. Enter values in the Attribute MBean Type, Attribute MBean Name, and Attribute Name fields. For more information, refer to “Determining Which WebLogic Server Attributes to Monitor” on page 516-6.

For example, if you want to monitor the `AcceptBacklog` attribute of the `Server` MBean for `MedRecServer`:

- In the MBean Type field, enter `Server`.
- In the MBean Name field, enter `MedRecServer`.
- In the Attribute Name field, enter `AcceptBacklog`.

The new instance is added under the String Monitors node in the left pane.

5. Next to Enabled Servers move the servers on which the MBean or MBeans are located from the Chosen to the Available column.

If you are configuring a monitor for a domain-wide resource, such as a JDBC Connection Pool, move the Administration Server to the Available column.

Note: When you assign an Attribute Change monitor to a server, you deploy a JMX listener and filter to the server. The listener will forward a notification to the WebLogic SNMP agent only if an event matches the criteria in the Attribute Change monitor.

6. Click the Create button.

The new instance is added under the Attribute Changes node in the left pane.

7. Restart the Administration Server so that your changes can take effect.

Configuring a String Monitor

A String Monitor periodically checks the value of a `String` attribute and generates a trap if the value is equal to or different from another specified string.

To configure a string monitor:

1. In the Administration Console, in the left pane, expand the following folders: Services—SNMP—Traps. Then click on the name of the String Monitors folder.
The SNMP String Monitor page displays. This page lists all the string monitors that are configured for the domain.
2. On the SNMP String Monitor page, select the Create a new String Monitor link.
3. On the String Monitor page, enter a name for the monitor instance in the Name field.

BEA Systems recommends that you choose a name that indicates the resource that is being monitored.

4. Enter values in the Monitored MBean Type, Monitored Attribute Name, and (optionally) Monitored MBean Name fields. For more information, refer to “Determining Which WebLogic Server Attributes to Monitor” on page 516-6.

For example, if you want to monitor the `State` attribute of the `ServerRuntime` MBean for `MedRecServer`:

- In the MBean Type field, select `Server Runtime`.
 - In the MBean Name field, enter `MedRecServer`.
 - In the Attribute Name field, enter `State`.
5. In the Polling Interval field, enter the frequency in seconds at which you want WebLogic Server to check the attribute’s value.

For testing purposes, consider entering a small value, such as 10.

A value of 0 means that the monitor **never** polls the attribute, effectively disabling this monitor.

6. In the String to Compare field, enter a string.
7. To send a trap when the observed attribute value differs from the string, select Notify Differ. For example, if you want to be notified when a server’s life cycle state is anything other than `Running`, enter `Running` in the String to Compare field and check the Notify Differ check box.

To send a trap when the observed attribute value matches the string, select Notify Match.

8. Click Create.

The new instance is added under the String Monitors node in the left pane.

9. Click the Servers tab and place a check mark next to servers on which the MBean or MBeans are located.

If you are configuring a monitor for a domain-wide resource, such as a JDBC Connection Pool, select the Administration Server.

Note: When you assign a String Monitor to a server, you deploy a JMX listener and filter to the server. The listener will forward a notification to the WebLogic SNMP agent only if an event matches the criteria in the String Monitor.

For example, if you create a String Monitor that observes the `State` attribute of `ServerA`'s `ServerRuntime` MBean, and if you target this monitor to `ServerB`, the monitor on `ServerB` will never generate a trap, because `ServerB` does not have access to the state of `ServerA`.

10. Click Apply.
11. Restart the Administration Server.

Configuring a Gauge Monitor

A gauge monitor periodically check the value of an integer or floating-point attribute and generate a trap if the value exceeds a threshold.

To configure a gauge monitor:

1. In the Administration Console, in the left pane, expand the following folders: `Services—SNMP—Traps`. Then click on the name of the Gauge Monitors folder.
The SNMP Gauge Monitor page displays. This page lists all the gauge monitors that are configured for the domain.
2. On the SNMP Gauge Monitor page, select the Create a new Gauge Monitor link.
3. On the Gauge Monitor page, enter a name for the monitor instance in the Name field.

BEA Systems recommends that you choose a name that indicates the resource that is being monitored.

4. Enter values in the Monitored MBean Type, Monitored Attribute Name, and (optionally) Monitored MBean Name fields. For more information, refer to “Determining Which WebLogic Server Attributes to Monitor” on page 516-6.

For example, if you want to monitor the `ActiveConnectionsHighCount` attribute of the `JDBCConnectionPoolRuntime` MBean for a JDBC connection pool named `MedRecPool`:

- In the MBean Type field, select `JDBCConnectionPool Runtime`.
 - In the MBean Name field, enter `MedRecPool`.
 - In the Attribute Name field, enter `ActiveConnectionsHighCount`.
5. In the Polling Interval field, enter the frequency in seconds at which you want WebLogic Server to check the attribute’s value.

For testing purposes, consider entering a small value, such as 10.

A value of 0 means that the monitor **never** polls the attribute, effectively disabling this monitor.

6. To send a trap when the observed attribute value is equal to or greater than a specific value, enter the value in the High Threshold field.

To send a trap when the observed attribute value is equal to or less than a specific value, enter the value in the Low Threshold field.

7. Click Create.

The new instance is added under the Gauge Monitors node in the left pane.

8. Click the Servers tab and place a check mark next the servers that you want to enable for trap generation.

If you are configuring a monitor for a domain-wide resource, such as a JDBC Connection Pool, select the Administration Server.

Note: When you assign a Gauge Monitor to a server, you deploy a JMX listener and filter to the server. The listener will forward a notification to the WebLogic SNMP agent only if an event matches the criteria in the String Monitor.

For example, if you create a Gauge Monitor that observes the `AcceptBacklog` attribute of `ServerA`’s `Server` MBean, and if you target this monitor to `ServerB`, the monitor on `ServerB` will never generate a trap, because `ServerB` does not have access to the configuration data of `ServerA`.

9. Click Apply.
10. Restart the Administration Server.

Configuring a Counter Monitor

A counter monitor periodically checks the value of an integer attribute and generates a trap when the value exceeds a threshold. You can also configure a Counter Monitor to increase the threshold by an offset value after it sends a trap.

To configure a counter monitor:

1. In the Administration Console, in the left pane, expand the following folders: Services—SNMP—Traps. Then click on the name of the Counter Monitors folder.

The SNMP Counter Monitor page displays. This page lists all the counter monitors that are configured for the domain.

2. On the SNMP Counter Monitor page, select the Create a new Counter Monitor link.
3. On the Counter Monitor page, enter a name for the monitor instance in the Name field.

BEA Systems recommends that you choose a name that indicates the resource that is being monitored.

4. Enter values in the Monitored MBean Type, Monitored Attribute Name, and (optionally) Monitored MBean Name fields. For more information, refer to “Determining Which WebLogic Server Attributes to Monitor” on page 516-6.

For example, if you want to monitor the `ActiveConnectionsHighCount` attribute of the `JDBCConnectionPoolRuntime` MBean for a JDBC connection pool name `MedRecPool`:

- In the MBean Type field, select `JDBCConnectionPool Runtime`.
 - In the MBean Name field, enter `MedRecPool`.
 - In the Attribute Name field, enter `ActiveConnectionsHighCount`.
5. In the Polling Interval field, enter the frequency in seconds at which you want WebLogic Server to check the attribute’s value.

For testing purposes, consider entering a small value, such as 10.

A value of 0 means that the monitor **never** polls the attribute, effectively disabling this monitor.

6. Enter data in the remaining fields as described in the next section, "Typical Configurations for Counter Monitors."
7. Click Create.

The new instance is added under the Counter Monitors node in the left pane.

8. Click the Servers tab and place a check mark next the servers that you want to enable for trap generation.

If you are configuring a monitor for a domain-wide resource, such as a JDBC Connection Pool, select the Administration Server.

Note: When you assign a Counter Monitor to a server, you deploy a JMX listener and filter to the server. The listener will forward a notification to the WebLogic SNMP agent only if an event matches the criteria in the String Monitor.

For example, if you create a Counter Monitor that observes the `AcceptBacklog` attribute of `ServerA's Server MBean`, and if you target this monitor to `ServerB`, the monitor on `ServerB` will never generate a trap, because `ServerB` does not have access to the configuration data of `ServerA`.

9. Click Apply.
10. Restart the Administration Server.

Typical Configurations for Counter Monitors

The following list describes how to achieve typical configurations of a Counter Monitor instance by entering data on the Counter Monitor page:

- To send a trap when the observed attribute exceeds a threshold, enter a threshold values in the Threshold field.
- To send a trap when the observed attribute exceeds the threshold and then increase the threshold by an offset value, enter a threshold in the Threshold field and an offset value in the Offset field.

Each time the observed attribute exceeds the new threshold, the threshold is increased by the offset value. 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.

To specify a maximum value for the threshold, enter a value in the Modulus field. When the threshold reaches the value specified by the modulus, the threshold is returned to the value that was specified through the latest call to the monitor's `setThreshold` method, 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.

Disabling Trap Generation


To disable the generation of Log Filter and Attribute Change traps, you must delete the trap configuration.

To disable the generation of String Monitor, Gauge Monitor, or Counter Monitor traps, you can either delete the trap or you can set its polling rate to 0.

To delete a trap configuration:

1. In the Administration Console, in the left pane, expand the following folders: Services—SNMP—Traps. Then click on the name of the folder that contains the trap.

For example, to delete an Attribute Change trap configuration, expand Services—SNMP—Traps. Then click on the name of the Attribute Changes folder.

2. In the right pane, in the table that lists the traps that you have configured, click the delete icon  in the table row that represents the trap configuration.

To prevent a String Monitor, Gauge Monitor, or Counter Monitor trap from being generated without deleting the trap configuration:

1. In the Administration Console, in the left pane, expand the following folders: Services—SNMP—Traps—Monitors. Then expand the folder that contains the trap.
2. Click on the trap configuration.
3. In the right pane, on the General tab, in the Polling Interval field, enter 0.
4. Restart the Administration Server.

Configuring an SNMP Proxy

The WebLogic SNMP agent can act as a proxy for other SNMP agents. It listens for requests from SNMP managers. If any of the OIDs in the requests fall under the control a proxy that you have defined, WebLogic Server forwards the request to the SNMP agent that is associated with the proxy. For more information, refer to "[SNMP Proxies](#)."

Use the following tasks to configure WebLogic Server to act as a proxy for an SNMP agent:

1. In the Administration Console, in the left pane, expand the following folders: Services—SNMP. Then click on the name of the Proxies folder.
2. In the right pane, click the Configure a new Proxy text link.
3. On the Create a new SNMPProxy page, enter values in the attribute fields. For more information, refer to "Attributes" on page 509-2.
4. Click Apply to create the new proxy.
5. Restart the Administration Server.

Attributes and Console Screen Reference for SNMP

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“Domain --> Configuration --> SNMP” on page 69-1](#)

[“SMNP --> Proxies” on page 510-1](#)

[“SMNP --> Proxies --> General” on page 509-1](#)

[“SMNP --> Traps --> SNMP Monitors” on page 506-1](#)

[“SMNP --> Traps --> Gauge Monitor” on page 505-1](#)

[“SMNP --> Traps --> Gauge Monitor --> General” on page 503-1](#)

[“SMNP --> Traps --> Gauge Monitor --> Servers” on page 504-1](#)

[“SMNP --> Traps --> String Monitor” on page 513-1](#)

[“SMNP --> Traps --> String Monitor --> General” on page 511-1](#)

[“SMNP --> Traps --> String Monitor --> Servers” on page 512-1](#)

[“SMNP --> Traps --> Counter Monitor” on page 502-1](#)

[“SMNP --> Traps --> Counter Monitor --> Configuration” on page 500-1](#)

[“SMNP --> Traps --> Counter Monitor --> Servers” on page 501-1](#)

[“SMNP --> Traps --> Log Filter” on page 508-1](#)

[“SMNP --> Traps --> Log Filter --> General” on page 507-1](#)

[“SMNP --> Traps --> Attribute Change” on page 499-1](#)

[“SMNP --> Traps --> Attribute Change --> Configuration” on page 498-1](#)

[“SNMP --> Trap Destinations” on page 515-1](#)

[“SNMP --> Trap Destinations --> General” on page 514-1](#)



Startup or Shutdown Class

Startup and shutdown classes are Java programs that you create and deploy to WebLogic Server.

To create a new configuration for a startup or shutdown class, click **Configure a new Startup Class** or **Configure a new Shutdown Class**. For more information, see “**Configuring a Server to Use a Startup or Shutdown Class: Main Steps**” on page 525-2.

For information on changing the information that this tab displays, refer to “**Customizing Table Views**” on page 6-15.



Shutdown Class --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The Shutdown Class—Configuration tab defines a named configuration for a shutdown class. You use this tab while creating a new configuration or modifying an existing one.

A shutdown class is automatically loaded and executed when a WebLogic Server is shut down either from the Administration Console or using the `weblogic.admin.shutdown` command.

If you are creating a new configuration, you must enter information on the Configuration tab and click Apply before you can access other tabs. Once you create a configuration, you cannot change its name. Instead, you must create a new configuration under a different name.

You can create multiple named configurations for any given shutdown class. For each configuration that you create, WebLogic Server uses a managed bean (MBean) to configure the shutdown class and persist the configuration.

Tasks

“Configuring a Server to Use a Startup or Shutdown Class: Main Steps” on page 525-2

“Modify an Existing Startup or Shutdown Configuration” on page 525-8

“Delete a Startup or Shutdown Class Configuration” on page 525-8

Related Topics

[“Starting and Stopping Servers” on page 497-1](#)

Attributes

Table 519-1

Attribute Label	Description	Value Constraints
Name	<p>An alpha-numeric value that identifies the shutdown class configuration. This name attribute is for your identification purposes only.</p> <p>After you have created a shutdown class configuration, you cannot change its name. Instead, delete the configuration and create a new one with a new name.</p> <p><i>MBean:</i> weblogic.management.configuration.ShutdownClassMBean</p> <p><i>Attribute:</i> Name</p>	
ClassName	<p>The fully qualified name of the Java class that you have created and compiled. The class must be on the server's classpath.</p> <p><i>MBean:</i> weblogic.management.configuration.ShutdownClassMBean</p> <p><i>Attribute:</i> ClassName</p>	<i>Required:</i> yes

Table 519-1

Attribute Label	Description	Value Constraints
Deployment Order	<p>A priority that the server uses to determine when it deploys an item. The priority is relative to other deployable items of the same type. For example, the server prioritizes and deploys all EJBs before it prioritizes and deploys startup classes.</p> <p>Items with the lowest Deployment Order value are deployed first. There is no guarantee on the order of deployments with equal Deployment Order values. There is no guarantee of ordering across clusters.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ShutdownClassMBean</code></p> <p><i>Attribute:</i> <code>DeploymentOrder</code></p>	<p><i>Minimum:</i> <code>MIN_ORDER</code></p> <p><i>Maximum:</i> <code>MAX_ORDER</code></p> <p><i>Default:</i> <code>DEFAULT_ORDER</code></p> <p><i>Dynamic:</i> yes</p>
Arguments	<p>Arguments that the server uses to initialize a class. Separate multiple arguments with a comma. For example:</p> <pre>attribute1=value1,attribute2=value2</pre> <p>Do not use spaces to separate the arguments; use only a comma.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ShutdownClassMBean</code></p> <p><i>Attribute:</i> <code>Arguments</code></p>	



Shutdown Class --> Target and Deploy

Tasks

Overview

This tab shows all of the servers and clusters that are in the domain. Use it to assign or unassign this shutdown-class configuration for one or more of those servers or clusters.

When you assign a shutdown class to a cluster, the Administration Server makes sure that the assignment succeeds for all servers in the cluster. If the assignment fails for one server, then it fails for the entire cluster.

Tasks

“Assign a Startup or Shutdown Configuration to Servers or Clusters” on page 525-5



Shutdown Class --> Notes

[Tasks](#) [Attributes](#)

Overview

This tab provides a space to maintain administrative notes on the current configuration of a shutdown class.

Tasks

“Configure a Startup or Shutdown Class” on page 525-2

“Modify an Existing Startup or Shutdown Configuration” on page 525-8

Attributes

Table 521-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.ShutdownClassMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes



Startup Class --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The Startup Class—Configuration tab defines a named configuration for a startup class. You use this tab while creating a new configuration or modifying an existing one.

If you are creating a new configuration, you must enter information on the Configuration tab and click Apply before you can access other tabs. Once you create a configuration, you cannot change its name. Instead, you must create a new configuration under a different name.

You can create multiple named configurations for any given startup class. For each configuration that you create, WebLogic Server uses a managed bean (MBean) to configure the startup class and persist the configuration.

Tasks

“Configuring a Server to Use a Startup or Shutdown Class: Main Steps” on page 525-2

“Modify an Existing Startup or Shutdown Configuration” on page 525-8

“Delete a Startup or Shutdown Class Configuration” on page 525-8

Related Topics

["Ordering Startup Class Execution and Deployment"](#)

Attributes

Table 522-1

Attribute Label	Description	Value Constraints
Name	<p>An alpha-numeric value that identifies the startup class configuration. This name attribute is for your identification purposes only.</p> <p>After you have created a startup class configuration, you cannot change its name. Instead, delete the configuration and create a new one with a new name.</p> <p><i>MBean:</i> weblogic.management.configuration.StartupClassMBean</p> <p><i>Attribute:</i> Name</p>	
ClassName	<p>The fully qualified name of the Java class that you have created and compiled. The class must be on the server's classpath.</p> <p><i>MBean:</i> weblogic.management.configuration.StartupClassMBean</p> <p><i>Attribute:</i> ClassName</p>	<i>Required:</i> yes

Table 522-1

Attribute Label	Description	Value Constraints
Deployment Order	<p>A priority that the server uses to determine when it deploys an item. The priority is relative to other deployable items of the same type. For example, the server prioritizes and deploys all EJBs before it prioritizes and deploys startup classes.</p> <p>Items with the lowest Deployment Order value are deployed first. There is no guarantee on the order of deployments with equal Deployment Order values. There is no guarantee of ordering across clusters.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.StartupClassMBean</code></p> <p><i>Attribute:</i> <code>DeploymentOrder</code></p>	<p><i>Minimum:</i> <code>MIN_ORDER</code></p> <p><i>Maximum:</i> <code>MAX_ORDER</code></p> <p><i>Default:</i> <code>DEFAULT_ORDER</code></p> <p><i>Dynamic:</i> yes</p>
Arguments	<p>Arguments that the server uses to initialize a class. Separate multiple arguments with a comma. For example:</p> <p><code>attribute1=value1,attribute2=value2</code></p> <p>Do not use spaces to separate the arguments; use only a comma.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.StartupClassMBean</code></p> <p><i>Attribute:</i> <code>Arguments</code></p>	
Failure is fatal	<p>Determines whether a failure in this startup class prevents a server from starting. If this check box is cleared and the startup class fails, the server continues its startup process.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.StartupClassMBean</code></p> <p><i>Attribute:</i> <code>FailureIsFatal</code></p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false

Table 522-1

Attribute Label	Description	Value Constraints
Run Before Application Deployments	<p>Causes a server to load and run a startup class before it activates JMS and JDBC services and before it starts deployment for applications and EJBs.</p> <p>Deployment for applications and EJBs consists of two phases: prepare and activate. If you select the Run Before Application Deployments option for a startup class, a server loads and runs the startup class before the prepare phase. At this point, JMS and JDBC services are not yet available.</p> <p>By default, a server instance loads startup classes after it activates JMS and JDBC services, EJBs, and applications.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.StartupClassMBean</code></p> <p><i>Attribute:</i> <code>LoadBeforeAppDeployments</code></p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false

Table 522-1

Attribute Label	Description	Value Constraints
Run Before Application Activations	<p>Causes a server to load and run a startup class after it activates JMS and JDBC services and before it activates applications and EJBs.</p> <p>Deployment for applications and EJBs consists of two phases: prepare and activate. If you select the Run Before Application Activations option for a startup class, a server loads and runs the startup class before the activate phase. At this point, JMS and JDBC services are available.</p> <p>Select this option if the startup class needs to be invoked after JDBC connection pools are available but before the applications are activated and ready to service client requests.</p> <p>By default, a server instance loads startup classes after it activates JMS and JDBC services, EJBs, and applications.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.StartupClassMBean</code></p> <p><i>Attribute:</i> <code>LoadBeforeAppActivation</code></p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ true ■ false



Startup Class --> Target and Deploy

Tasks

Overview

This tab shows all of the servers and clusters that are in the domain. Use it to assign or unassign this startup-class configuration for one or more of those servers or clusters.

To deploy to a cluster of servers, select the name of the cluster from the Clusters list. By default, the Administration Console deploys a startup class to all server instances in the cluster (the All servers in the cluster option). If you want to deploy to only a subset of the servers in a cluster, select Parts of the cluster, and then select the individual server instances to which you want to deploy the class.

Tasks

“Assign a Startup or Shutdown Configuration to Servers or Clusters” on page 525-5



Startup Class --> Notes

[Tasks](#) [Attributes](#)

Overview

This tab provides a space to maintain administrative notes on the current configuration of a startup class.

Tasks

“Configure a Startup or Shutdown Class” on page 525-2

“Modify an Existing Startup or Shutdown Configuration” on page 525-8

Attributes

Table 524-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.StartupClassMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes



1 Startup and Shutdown Classes

[“Attributes and Console Screen Reference for Startup and Shutdown” on page 526-1]

You can use startup and shutdown classes to configure a WebLogic Server to perform tasks when you start or gracefully shut down the server. A startup class is a Java program that is automatically loaded and executed when a WebLogic Server is started or restarted. For information about when WebLogic Server loads and executes startup classes, see "[Ordering Startup Class Execution and Deployment](#)."

A shutdown class is a Java program that is automatically loaded and executed when the WebLogic Server is shut down either from the Administration Console or the `weblogic.admin shutdown` command. For more information about when a server invokes startup and shutdown classes, refer to "[Server Lifecycle](#)."

To use startup or shutdown classes, you must configure and assign them to servers or clusters.

The following sections describe how to use startup and shutdown classes:

- “Configuring a Server to Use a Startup or Shutdown Class: Main Steps” on page 525-2
- “Modify an Existing Startup or Shutdown Configuration” on page 525-8
- “Delete a Startup or Shutdown Class Configuration” on page 525-8

Configuring a Server to Use a Startup or Shutdown Class: Main Steps

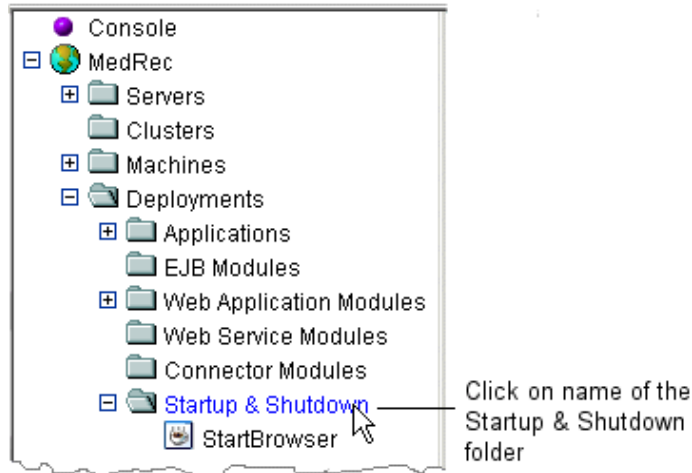
To configure a server instance to use a startup or shutdown class, complete the following tasks:

1. “Configure a Startup or Shutdown Class” on page 525-2 or “Clone a Configuration of a Startup or Shutdown Class” on page 525-4.
2. “Assign a Startup or Shutdown Configuration to Servers or Clusters” on page 525-5.
3. “Add the Class to the Server’s Classpath” on page 525-6.

Configure a Startup or Shutdown Class

1. In the left pane, expand the Deployments folder.
2. Click on the name of the Startup & Shutdown folder. (See Figure 525-1.)

Figure 525-1 Click on the Name of the Startup & Shutdown Folder



The Administration Console displays the Startup & Shutdown page. This page includes a list of the configurations that already exist in this domain. If any existing configuration is similar to the one that you want to create, consider cloning it. For more information, refer to “Clone a Configuration of a Startup or Shutdown Class” on page 525-4.

3. On the Startup & Shutdown page, do one of the following:
 - To configure a startup class, click the Configure a New Startup Class text link.
 - To configure a shutdown class, click the Configure a New Shutdown Class text link.

The Administration Console displays the set of tabs that you use to configure a startup or shutdown class.

4. Enter values in the Name, Class Name, Deployment Order, and Arguments attribute fields. For more information about each field, refer to “Attributes” on page 522-2 for startup classes or “Attributes” on page 519-2 for shutdown classes.

In the Arguments field, separate multiple arguments with a comma. For example,

```
firstname=MyFirst,lastname=MyLast
```

Do not use spaces to separate the arguments; use only a comma.

5. If you want a startup class to load and execute:
 - Before a server instance activates JMS and JDBC services, applications and EJBs, select the Run Before Application Activations checkbox.
 - After a server instance activates JMS and JDBC services, and before it starts deployment for EJBs and applications, select the Run Before Application Deployments checkbox.
 - After a server instance activates JMS and JDBC services, EJBs, and applications, clear both checkboxes.
6. Click Create.

The Administration Console adds your new configuration under the Startup & Shutdown node in the left pane.
7. If you want to add a note that describes your configuration, click the Notes tab. Enter a note and click Apply.

When you are ready to make the configured class available to one or more servers, refer to “Assign a Startup or Shutdown Configuration to Servers or Clusters” on page 525-5.

Clone a Configuration of a Startup or Shutdown Class

1. In the left pane, click the Startup & Shutdown node.

The Administration Console displays the Startup & Shutdown page. This page displays a list of the configurations that already exist in this domain.
2. On the Startup & Shutdown page, click the Clone icon in the row of the class you want to clone.

The Administration Console displays the set of tabs that you use to configure a startup or shutdown class.
3. Enter a value in the Name field.
4. You can modify any of the values in the Class Name, Deployment Order, and Arguments attribute fields. You can also change the setting for the Abort Startup on Failure checkbox.

5. Click Clone to create a configuration with the name that you specified in the Name field.

The Administration Console adds your new configuration under the Startup & Shutdown node in the left pane.

6. If you want to add or modify a note that describes your configuration, click the Notes tab. Enter a note and click Apply.

When you are ready to make the new configuration class available to one or more servers, refer to “Assign a Startup or Shutdown Configuration to Servers or Clusters” on page 525-5.

Assign a Startup or Shutdown Configuration to Servers or Clusters

1. In the left pane, expand the Startup & Shutdown folder and click a configuration.

The Administration Console displays the configuration in the right pane.

2. In the right pane, click the Target and Deploy tab.

If you have configured a cluster for the domain, the Target and Deploy tab displays a Clusters table, which lists all servers that are assigned to the cluster.

3. To assign a startup or shutdown class to servers that are in a cluster, do the following:
 - a. On the Target and Deploy tab, in the Clusters table, select the check box next to the cluster.
 - b. To assign the class to all servers in the cluster, click the All Servers in the Cluster button. (See Figure 525-2.)
 - c. To assign the class only to one or more specific servers, click the Part of the Cluster button. Then select the servers you want to target.

When you deploy to all or part of a cluster, the Administration Console initiates a two-phase deployment. In general, such a deployment ensures that if the deployment fails for one active server, it fails for all active servers. For more information, refer to "Introduction to Two-Phase Deployment" and "Guidelines for Deploying to a Cluster."

Figure 525-2 Deploy to an Entire Cluster

Configuration | **Target and Deploy** | Notes

Select the Servers and/or Clusters on which you would like to deploy this Startup Class.

Independent Servers	
<input type="checkbox"/>	MedRecServer

Targets:

Clusters	
<input checked="" type="checkbox"/>	MyCluster
<input checked="" type="radio"/>	All servers in the cluster
<input type="radio"/>	Part of the cluster
<input type="checkbox"/>	ManagedServer1
<input type="checkbox"/>	ManagedServer2
<input type="checkbox"/>	ManagedServer3

Apply

4. To assign the startup or shutdown class to servers that are not in a cluster, in the Independent Servers table, select one or more servers.

If the deployment fails for one server, the Administration Console reports the error but continues deploying on any remaining servers that you selected.

5. Click Apply.
6. If you have not already done so, “Add the Class to the Server’s Classpath” on page 525-6.

Note that clicking Apply sets the value of the Server (or Cluster)—Deployments—Startup/Shutdown list for each server or cluster that you specified.

Add the Class to the Server’s Classpath

The startup or shutdown class must be on the classpath of each server to which it is assigned. To add a class to a server’s classpath, do one of the following:

- If you use a script to start a server instance, open the script in a text editor. In the command that sets the classpath, add the pathname of the directory that contains your class's root package. Then restart the server.

For example, you create a startup class named `StartBrowser` in a package named `com.mycompany.startup`. You archive the class file in a JAR file named `c:\myDomain\src\myJAR.jar`. The start script for your server must add `c:\myDomain\src\myJAR.jar` to the server's classpath.

- If you use the Node Manager to start a server instance, do the following:
 - a. In the left pane of the Administration Console, expand the Servers folder.
 - b. Click on the name of the server.
 - c. In the right pane, click the Configuration tab. Then click the Remote Start tab.
 - d. In the Classpath field, enter the classes that WebLogic Server requires to be on the classpath.

For example, `weblogic.jar` must be on the classpath. For a complete list, refer to [Setting the Classpath](#). Use an absolute pathname or a pathname that is relative to the Node Manager's home directory.

Separate multiple classes with the type of separator that your operating system or shell requires. For example, on Windows, use `;` (semicolon) and in a BASH shell, use `:` (colon).

- e. In the Classpath field, add your class to the classpath.

For example, you create a startup class named `StartBrowser` in a package named `com.mycompany.startup`. You archive the class file in a JAR file named `c:\myDomain\src\myJAR.jar`. The start script for your server must add `c:\myDomain\src\myJAR.jar` to the server's classpath.

The following values in the Classpath field register the required classes and your `StartBrowser` startup class:

```
c:\bea\weblogic810\server\lib\weblogicsp.jar;c:\bea\weblogic810\server\lib\weblogic.jar;c:\myDomain\src\myJAR.jar
```

- f. Restart the server.

Modify an Existing Startup or Shutdown Configuration

1. In the left pane, expand the Startup & Shutdown node and click a configuration.
The Administration Console displays the configuration in the right pane.
2. On the Configuration tab, modify any of the values in the Class Name, Deployment Order, and Arguments attribute fields. You can also change the settings for the Failure is fatal or Run before application deployments checkboxes.
3. Click Apply to save any changes.
4. On the Notes tab, modify or add a note.
5. Click Apply to save any changes.
6. On the Targets tab reassign the configuration to different clusters or servers. To remove the configuration from a target, select the target in the Chosen column and click the mover control to move it to the Available column.
7. Click Apply to save any changes.

Delete a Startup or Shutdown Class Configuration

1. In the left pane, click the Startup & Shutdown node.
The Administration Console displays the Startup & Shutdown page. This page displays a list of the configurations that already exist in this domain.
2. On the Startup & Shutdown page, click the Delete icon in the row of a configuration.
A dialog displays in the right pane asking you to confirm your deletion request.

3. Click Yes to delete the configuration.

Attributes and Console Screen Reference for Startup and Shutdown

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“Startup or Shutdown Class” on page 518-1](#)

[“Startup Class --> Configuration” on page 522-1](#)

[“Startup Class --> Target and Deploy” on page 523-1](#)

[“Startup Class --> Notes” on page 524-1](#)

[“Shutdown Class --> Configuration” on page 519-1](#)

[“Shutdown Class --> Target and Deploy” on page 520-1](#)

[“Shutdown Class --> Notes” on page 521-1](#)



Tasks --> Status

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The Task Runtime --> Status tab displays the following information about the task:

- A description of the task
- The status of the task
- The time when the task execution began.
- The time when the task execution ended.
- Any exception that occurred while executing the task.
- A log of the task's activities.

Tasks

[View Details about a Task](#)

Related Topics

[“Deploying Applications and Modules” on page 62-1](#)

[“Starting and Stopping Servers” on page 497-1](#)

Attributes

Table 527-1

Attribute Label	Description	Value Constraints
Description	A description of this task <i>MBean:</i> weblogic.management.runtime.TaskRuntimeMBean <i>Attribute:</i> Description	
Status	The status of this Task. <i>MBean:</i> weblogic.management.runtime.TaskRuntimeMBean <i>Attribute:</i> Status	
Begin Time	The time that the task started. <i>MBean:</i> weblogic.management.runtime.TaskRuntimeMBean <i>Attribute:</i> BeginTime	
End Time	The time that the task completed. <i>MBean:</i> weblogic.management.runtime.TaskRuntimeMBean <i>Attribute:</i> EndTime	

Tasks --> Details

The Tasks --> Details Tab displays detailed information regarding this task. If the task failed, the Details tab displays the complete text of the exception, a stack trace or other diagnostic information.

Tasks

View Details about a Task

Related Topics

[“Deploying Applications and Modules” on page 62-1](#)

[“Starting and Stopping Servers” on page 497-1](#)



Migration Task Runtime --> migration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Displays the status of a migration task

Tasks

[Configure Migratable Targets for Pinned Services](#)

[Migrating a Pinned Service to a Target Server Instance](#)

Related Topics

[Using WebLogic Server Clusters](#)

Attributes

Table 529-1

Attribute Label	Description	Value Constraints
Description	A description of this task	
Status	The status of this Task.	

Table 529-1

Attribute Label	Description	Value Constraints
Begin Time	The time that the migration task started. <i>MBean:</i> weblogic.management.runtime.MigrationTaskRuntimeMBean <i>Attribute:</i> BeginTime	
End Time	The time that the migration task completed. <i>MBean:</i> weblogic.management.runtime.MigrationTaskRuntimeMBean <i>Attribute:</i> EndTime	

MigrationTaskRuntime --> Details

The Migration Tasks --> Details Tab displays a list of executed actions regarding this migration task.

- For more information, see Tasks Status



Tasks

This page provides an overview of administrative tasks such as deployments, service migration, and attempts to start or stop managed servers. Click on a task's description to get more detailed information about the task.

For more information, see “Tasks Status” on page 532-1.

To purge completed tasks from the list, click the Purge Tasks button.



1 Tasks Status

[“Attributes and Console Screen Reference for Tasks” on page 533-1]

The Tasks page provides an log of administrative tasks such as deployments, service migration, and attempts to start or stop managed servers. Because some tasks are completed immediately and others take varying amounts of time to complete, the tasks log allows you to monitor the completion status of all tasks.

View Details about a Task

1. Click on a task's description.
2. Click on the Status tab to get more information about the task.
3. Click on the Details tab to view details about the task, including the complete exception or stack trace if the task fails.

Attributes and Console Screen Reference for Tasks

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“Migration Task Runtime --> migration” on page 529-1](#)

[“MigrationTaskRuntime --> Details” on page 530-1](#)

[“Tasks” on page 531-1](#)

[“Tasks --> Details” on page 528-1](#)

[“Tasks --> Status” on page 527-1](#)



Virtual Host --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The Configuration tab is used to create or edit a VirtualHost associated with this instance of WebLogic Server.

Tasks

“Configuring a VirtualHost” on page 540-1

Related Topics

For additional information about VirtualHosts, please see the *Administration Guide*, [Overview of WebLogic Server HTTP Services](#).

Attributes

Table 534-1

Attribute Label	Description	Value Constraints
Name	The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration. <i>MBean:</i> weblogic.management.configuration.VirtualHostMBean <i>Attribute:</i> Name	
Virtual Host Names	The host name for which this web server will serve requests. <i>MBean:</i> weblogic.management.configuration.VirtualHostMBean <i>Attribute:</i> VirtualHostNames	<i>Default:</i> null <i>Configurable:</i> yes

Virtual Host --> Configuration --> HTTP

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab allows you to set configuration attributes for a Virtual Host.

Tasks

“Configuring a VirtualHost” on page 540-1

“Cloning a VirtualHost” on page 540-2

Related Topics

For additional information about VirtualHosts, please see the *Administration Guide*, [Overview of WebLogic Server HTTP Services](#).

Attributes

Table 535-1

Attribute Label	Description	Value Constraints
Default Server Name	The HTTP defaultServerName <i>MBean:</i> weblogic.management.configuration.VirtualHostMBean <i>Attribute:</i> DefaultServerName	<i>Default:</i> null <i>Configurable:</i> yes
Post Timeout Secs	Timeout (in seconds) for reading HTTP POST data in a servlet request. <i>MBean:</i> weblogic.management.configuration.VirtualHostMBean <i>Attribute:</i> PostTimeoutSecs	<i>Minimum:</i> 0 <i>Maximum:</i> 120 <i>Default:</i> 30
Max Post Time Secs	Max Post Time (in seconds) for reading HTTP POST data in a servlet request. MaxPostTime < 0 means unlimited <i>MBean:</i> weblogic.management.configuration.VirtualHostMBean <i>Attribute:</i> MaxPostTimeSecs	<i>Units:</i> seconds <i>Default:</i> -1
Max Post Size	Max Post Size (in bytes) for reading HTTP POST data in a servlet request. MaxPostSize < 0 means unlimited <i>MBean:</i> weblogic.management.configuration.VirtualHostMBean <i>Attribute:</i> MaxPostSize	<i>Units:</i> bytes <i>Default:</i> -1

Table 535-1

Attribute Label	Description	Value Constraints
Keep Alive Enabled	Returns whether or not HTTP keep-alive is enabled <i>MBean:</i> weblogic.management.configuration.VirtualHostMBean <i>Attribute:</i> KeepAliveEnabled	<i>Default:</i> true <i>Valid values:</i> ■ true ■ false
Keep Alive Secs	Number of seconds to maintain HTTP keep-alive before timing out the request. <i>MBean:</i> weblogic.management.configuration.VirtualHostMBean <i>Attribute:</i> KeepAliveSecs	<i>Units:</i> seconds <i>Minimum:</i> 5 <i>Maximum:</i> 120 <i>Default:</i> 30
Https Keep Alive Secs	Number of seconds to maintain HTTPS keep-alive before timing out the request. <i>MBean:</i> weblogic.management.configuration.VirtualHostMBean <i>Attribute:</i> HttpsKeepAliveSecs	<i>Units:</i> seconds <i>Minimum:</i> 30 <i>Maximum:</i> 360 <i>Default:</i> 60
Accept Context Path In Get Real Path	Beginning with this release inclusion of the contextPath in the virtualPath to the context.getRealPath() will not be allowed as it breaks the case when the subdirectories have the same name as contextPath. In order to support applications which might have been developed according to the old behaviour we are providing a compatibility switch. This switch will be deprecated in future releases. <i>MBean:</i> weblogic.management.configuration.VirtualHostMBean <i>Attribute:</i> AcceptContextPathInGetRealPath	<i>Default:</i> false <i>Valid values:</i> ■ true ■ false



Virtual Host --> Configuration --> Logging

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The Virtual Host—Configuration—Logging tab configures the logging of HTTP requests for a virtual host. If you enable HTTP logging, the virtual host saves its HTTP requests in a log file that is separate from the server's log files.

You can also use this tab to specify the longevity and format of the data in the virtual hosts's HTTP log file.

Tasks

“Specifying HTTP Log File Settings for a Virtual Host” on page 540-3

“Configuring a VirtualHost” on page 540-1

Related Topics

For more information about HTTP logs, see "Setting Up HTTP Access Logs" under [Configuring WebLogic Server Web Components](#) in the *Administration Guide*. This topic includes information about using the extended format in HTTP logs.

For more information about VirtualHosts, please see the *Administration Guide*, [Configuring WebLogic Server Web Components](#).

Attributes

Table 536-1

Attribute Label	Description	Value Constraints
Logging Enabled	Enables logging of HTTP requests. <i>MBean:</i> weblogic.management.configuration.VirtualHostMBean <i>Attribute:</i> LoggingEnabled	<i>Default:</i> true <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false <i>Configurable:</i> yes
Log File Name	The HTTP request log file The name of the file that stores HTTP requests. If the pathname is not absolute, the path is assumed to be relative to the root directory of the machine on which the server is running. This value is relevant only if HTTP logging is enabled. To include a time or date stamp in the file name when the log file is rotated, add <code>java.text.SimpleDateFormat</code> variables. Surround each variable with percentage (%) characters. For example, <code>access_%yyyy%_%MM%_%dd%_%hh%_%mm%.log</code> If you do not include a time and date stamp, the rotated log files are numbered in order of creation <code>filenamennnnn</code> , where <i>filename</i> is the name configured for the log file. <i>MBean:</i> weblogic.management.configuration.VirtualHostMBean <i>Attribute:</i> LogFileName	<i>Default:</i> DEFAULT_LOG_FILE_NAME <i>Configurable:</i> yes

Table 536-1

Attribute Label	Description	Value Constraints
Log File Format	<p>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.</p> <p><i>MBean:</i> weblogic.management.configuration.VirtualHostMBean</p> <p><i>Attribute:</i> LogFileFormat</p>	<p><i>Default:</i> "common"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ "common" ■ "extended" <p><i>Configurable:</i> yes</p> <p><i>Dynamic:</i> yes</p>
Log File BufferK Bytes	<p>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 LogFileFlushSecs property to determine the frequency with which the server checks the size of the buffer.</p> <p><i>MBean:</i> weblogic.management.configuration.VirtualHostMBean</p> <p><i>Attribute:</i> LogFileBufferKBytes</p>	<p><i>Units:</i> kilobytes</p> <p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 1024</p> <p><i>Default:</i> 8</p> <p><i>Configurable:</i> yes</p>
Max Log File Size KBytes	<p>The file size (1 - 65535 kilobytes) that triggers the server to move log messages to a separate file. After the log file reaches the specified size, the next time the server checks the file size, it will rename the current log file as <i>FileName.n</i> and create a new one to store subsequent messages.</p> <p>0 causes the file to grow indefinitely.</p> <p>This property is relevant only if you choose to rotate files by size.</p> <p><i>MBean:</i> weblogic.management.configuration.VirtualHostMBean</p> <p><i>Attribute:</i> MaxLogFileSizeKBytes</p>	<p><i>Units:</i> kilobytes</p> <p><i>Minimum:</i> 0</p> <p><i>Default:</i> 0</p> <p><i>Configurable:</i> yes</p>

Table 536-1

Attribute Label	Description	Value Constraints
Log Rotation Type	<p>Criteria for moving old HTTP requests to a separate log file:</p> <ul style="list-style-type: none">■ By Size. When the log file reaches the size that you specify in <code>MaxLogFileSizeKBytes</code>, the server renames the file as <code>LogFileName.n</code>.■ By Time. At each time interval that you specify in <code>LogRotationPeriodMin</code>, the server renames the file as <code>LogFileName.n</code>. <p>After the server renames a file, subsequent messages accumulate in a new file with the name that you specified in <code>LogFileName</code>.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.VirtualHostMBean</code></p> <p><i>Attribute:</i> <code>LogRotationType</code></p>	<p><i>Default:</i> "size"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "size"■ "date" <p><i>Configurable:</i> yes</p>
Log Rotation Period Mins	<p>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.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.VirtualHostMBean</code></p> <p><i>Attribute:</i> <code>LogRotationPeriodMins</code></p>	<p><i>Units:</i> minutes</p> <p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 1440</p> <p><i>Configurable:</i> yes</p>

Table 536-1

Attribute Label	Description	Value Constraints
Log File Flush Secs	<p>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 <code>LogFileBufferKBytes</code> property, the server writes the data in the buffer to the HTTP request log file.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.VirtualHostMBean</code></p> <p><i>Attribute:</i> <code>LogFileFlushSecs</code></p>	<p><i>Units:</i> seconds</p> <p><i>Minimum:</i> 1</p> <p><i>Maximum:</i> 360</p> <p><i>Default:</i> 60</p> <p><i>Configurable:</i> yes</p>
Log Rotation Time Begin	<p>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 <code>LogRotationPeriodMins</code>.</p> <p>Use the following <code>java.text.SimpleDateFormat</code> format to specify a date and time: <code>MM-dd-yyyy-k:mm:ss</code>. For information about this format, refer to the J2EE Javadoc.</p> <p>If the time that you specify has already past, then the server starts its file rotation immediately.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.VirtualHostMBean</code></p> <p><i>Attribute:</i> <code>LogRotationTimeBegin</code></p>	<p><i>Configurable:</i> yes</p>



Virtual Host --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab allows the user to make notes to describe the Virtual Host.

Tasks

Related Topics

For additional information about VirtualHosts, please see the *Administration Guide*, [Overview of WebLogic Server HTTP Services](#).

Attributes

Table 537-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.VirtualHostMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

Domain --> VirtualHost --> Target and Deploy

[Tasks](#) [Related Topics](#)

Overview

Virtual hosting allows you to define host names to which servers or clusters respond. When you use virtual hosting you use DNS to specify one or more host names that map to the IP address of a WebLogic Server or cluster and you specify which Web Applications are served by each virtual host.

Tasks

“Assigning a VirtualHost” on page 540-5

Related Topics

For additional information about VirtualHosts, please see the *Administration Guide*, [Overview of WebLogic Server HTTP Services](#).



Virtual Host

This table allows you to view and manage the Virtual Hosts you have created for this instance of WebLogic Server. It allows you to choose the attributes by which you want to sort the Virtual Hosts you have created and to clone or delete selected Virtual Hosts.

To create a new Virtual Host click the *Configure a new Virtual Host* link.

To select attributes to be shown in the table Click the *Customize this View* link.

- For more information, see
- Configuring a VirtualHost



1 Virtual Hosts

[“Attributes and Console Screen Reference for Virtual Hosts” on page 541-1]

Virtual hosting allows you to define host names to which servers or clusters respond. When you use virtual hosting you use DNS naming to specify one or more host names that map to the IP address of a WebLogic Server or cluster and you specify which Web Applications are served by each virtual host.

Virtual Hosts are instances of a web server that have been given specific DNS names in order to facilitate routing requests from clients to the correct web server. Virtual Hosts prevent the client from discovering the IP address of the web server inadvertently by placing the DNS name into the header of the request rather than the IP address of the machine on which the web server is deployed..

Virtual Hosting is used in order to direct client requests to specific web servers using DNS names to differentiate between servers which may be found at the same IP address.

Tasks

Configuring 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 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.
5. Click the Logging tab to set up logging of HTTP requests. For more information, refer to “Specifying HTTP Log File Settings for a Virtual Host” on page 540-3. Click Apply to save your changes.
6. Click the HTTP tab and change the attribute fields or accept the default values as assigned. Click Apply to save your changes.
7. Restart the server for the Virtual Host to take effect.

Cloning 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 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.
5. Click the Logging tab to set up logging of HTTP requests. For more information, refer to “Specifying HTTP Log File Settings for a Virtual Host” on page 540-3. Click Apply to save your changes.
6. Click the HTTP tab and change the attribute fields or accept the default values as assigned. Click Apply to save your changes.

Specifying HTTP Log File Settings for a Virtual Host

1. Click the VirtualHost node. The VirtualHosts table displays in the right pane showing all the VirtualHosts defined in your domain.
2. Click a VirtualHost.
3. On the Logging tab, click Enable Logging to activate or deactivate HTTP logging. (If Enable Logging is checked, the HTTP logging will be activated.)

HTTP requests for a virtual host are kept in a log file that is separate from the server's log files.
4. Determine the format of the HTTP log file by selecting Common or Extended from the Format list.
5. To determine the frequency with which the server empties the virtual host's HTTP-request buffer and writes the data to the HTTP log file, do the following:
 - a. In Log File Buffer KBytes, specify the maximum size (in kilobytes) of the HTTP-request buffer.
 - b. In the Log File Flush Secs parameter, specify the interval (in seconds) at which the server checks the size of the HTTP-request buffer. If the buffer has reached the maximum size, the server writes the data to the HTTP log file.
6. If you want the server to move old HTTP requests to another file when the current HTTP log file reaches a specific size, do the following:
 - a. In Log Rotation Type, choose *By Size*.
 - b. In Max Log File Size KBytes, specify the file size (1 - 65535 kilobytes) that triggers the server to move log messages to a separate file. After the log file reaches the specified size, the next time the server checks the file size, it will rename the current log file and create a new one to store subsequent messages.
 - c. Click Apply.
7. If you want the server to move old HTTP requests to another file at specific time intervals, do the following:
 - a. In Rotation Type, choose *By Time*.
 - b. In Log Rotation Time Begin, enter the start time.

At the time that you specify, the server rotates the current log file. If the time that you specify has already past, then the server starts its file rotation immediately. Thereafter, the server rotates the log file at an interval that you specify in Log Rotation Period Mins.

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](#).

- c. In Log Rotation Period Mins, enter the interval (in minutes) at which the server saves old messages to another file.
 - d. Click Apply.
8. To include a time or date stamp in the file name when the log file is rotated, in the File Name field, add `java.text.SimpleDateFormat` variables to the file name. Surround each variable with percentage (%) characters.

For example, if you enter the following value in the File Name field:

```
access_%yyyy%_%MM%_%dd%_%hh%_%mm%.log
```

the virtual host's HTTP log file will be named:

```
access_yyyy_MM_dd_hh_mm.log
```

When the server instance rotates the HTTP log file, the rotated file name contains the date stamp. For example, if the server instance rotates the log file on 2 April, 2003 at 10:05 AM, the log file that contains the old log messages will be named:

```
access_2003_04_02_10_05.log
```

If you do not include a time and date stamp, the rotated log files are numbered in order of creation `filenamennnnn`, where `filename` is the name configured for the log file. For example: `access.log00007`.

Deleting 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.

Assigning 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 Target and Deploy tab.
3. Complete the following steps:
 - a. Select one or more targets that you want to assign to the VirtualHost.
 - b. Click Apply to save your assignments.

Targeting 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.

Associating a Virtual Host with a Server

1. Expand the Servers Node in the Left pane.
2. Select the server with which you would like to associate one or more Virtual Hosts.

3. Click the Services tab. Click the Virtual Hosts tab.
4. Select the Virtual Host you wish to associate with the server. Use the arrow to move the selected Virtual Host to the Chosen section of the tab.
5. Click Apply to confirm your selection.

Removing an Associated Virtual Host

1. Expand the Servers Node in the Left pane.
2. Select the server from which you would like to remove one or more associated Virtual Hosts.
3. Click the Services tab. Click the Virtual Hosts tab.
4. Select the Virtual Host you wish to associate with the server. Use the arrow to move the selected Virtual Host From the Chosen section of the tab to the Available section.
Click Apply to confirm your selection.

Attributes and Console Screen Reference for Virtual Hosts

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

“Domain --> VirtualHost --> Target and Deploy” on page 538-1

“Virtual Host --> Configuration --> General” on page 534-1

“Virtual Host --> Configuration --> HTTP” on page 535-1

“Virtual Host --> Configuration --> Logging” on page 536-1

“Virtual Host --> Notes” on page 537-1

“Virtual Host” on page 539-1



Web Application --> Configuration --> Descriptor

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

In the Connector Component > Configuration > Descriptor tab, you define the configuration of the application deployment descriptor file that is associated with this Web application module.

This release of WebLogic Server has deprecated the Administration Console Deployment Descriptor Editor. A new Descriptor tab in the Administration Console has replaced it. Using the Descriptor tab, you can view, modify, and persist deployment descriptor elements to the descriptor file within the Web application in the same manner that they were persisted using the Deployment Descriptor Editor.

However, these descriptor elements takes place dynamically at runtime without requiring the Web application to be redeployed. The descriptor elements contained in the Descriptor tab are limited to only those descriptor elements that may be dynamically changed at runtime.

Note: These attributes are maintained in the `weblogic.xml` deployment descriptor file. (For more information, refer to "[weblogic.xml](#)" in *Developing Web Applications for WebLogic Server*.) If you alter the value of any of these attributes and select Apply, then the Web Application deployment descriptor file is updated and deployed to all of the appropriate server machines.

Tasks

“Designating a Default Web Application” on page 559-2

“Deploying a New Web Application” on page 559-2

-
- “Testing the Deployment” on page 559-3
 - “Viewing Deployed Web Applications” on page 559-4
 - “Stopping Deployed Web Applications” on page 559-4
 - “Adding or Editing Web Application Deployment Notes” on page 559-4
 - “Deleting a Web Application” on page 559-5
 - “Monitoring Web Applications and Servlets” on page 559-5
 - “Viewing and Updating Run-Time Descriptor Elements” on page 559-6

Related Topics

[Developing Web Applications for WebLogic Server](#)

Attributes

Table 542-1

Attribute Label	Description	Value Constraints
Session Cookie Max Age Secs	The life span of the session cookie, in seconds, after which it expires on the client. <i>MBean:</i> <code>weblogic.management.configuration.WebAppComponentMBean</code> <i>Attribute:</i> <code>SessionCookieMaxAgeSecs</code>	<i>Units:</i> seconds

Table 542-1

Attribute Label	Description	Value Constraints
Session Invalidation Interval Secs	<p>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.</p> <p><i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean</p> <p><i>Attribute:</i> SessionInvalidationIntervalSecs</p>	<i>Units:</i> seconds
Session Timeout Secs	<p>The amount of time (in seconds) that a session can remain inactive before it is invalidated.</p> <p><i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean</p> <p><i>Attribute:</i> SessionTimeoutSecs</p>	<i>Units:</i> seconds
Servlet Reload Check Secs	<p>The amount of time (in seconds) that WebLogic Server waits to check if a servlet was modified and needs to be reloaded.</p> <p><i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean</p> <p><i>Attribute:</i> ServletReloadCheckSecs</p>	<i>Units:</i> seconds
Single Threaded Servlet Pool Size	<p>The size of the pool used for single threaded mode instance pools.</p> <p><i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean</p> <p><i>Attribute:</i> SingleThreadedServletPoolSize</p>	

Table 542-1

Attribute Label	Description	Value Constraints
Index Directory Enabled	<p>Specifies whether the target should automatically generate an HTML directory listing if no suitable index file is found.</p> <p><i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean</p> <p><i>Attribute:</i> IndexDirectoryEnabled</p>	
Enable Session Monitoring	<p>Specifies whether runtime MBeans will be created for sessions.</p> <p><i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean</p> <p><i>Attribute:</i> SessionMonitoringEnabled</p>	
JSPCompile Command	<p>The full pathname of the Java compiler to use for compiling the generated JSP servlets.</p> <p><i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean</p> <p><i>Attribute:</i> JSPCompileCommand</p>	
JSPPage Check Secs	<p>The interval, in seconds, at which WebLogic Server checks to see if JSP files have changed and need recompiling.</p> <p><i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean</p> <p><i>Attribute:</i> JSPPageCheckSecs</p>	<i>Units:</i> seconds
JSPKeep Generated	<p>Specifies whether to keep the generated source files or delete them after compiling a JSP.</p> <p><i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean</p> <p><i>Attribute:</i> JSPKeepGenerated</p>	

Table 542-1

Attribute Label	Description	Value Constraints
JSPVerbose	Enables the JSP compiler's verbose output option. <i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean <i>Attribute:</i> JSPVerbose	<i>Default:</i> false
Enable JSP Line Numbers	Compiles JSPs with the Java compiler's debug option enabled. <i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean <i>Attribute:</i> JSPDebug	<i>Default:</i> false



Web Application --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to set the deployment order and staging mode of the application. The deployment with the lowest deployment order value will be deployed first.

Available staging modes are:

1. nostage: does not copy application files to another location.

A server in nostage mode will run applications deployed to it directly from their source directories. In this mode, the web application container detects changes to JSPs and servlets.

2. stage: copies application files to server targeted in deployment

The stage mode means that the application will automatically be copied to the staging directory on the server. The servers will initialize and run the application from this directory.

3. external_stage: the user, and not WebLogic Server, copies application files to the server's staging directory.

The deployment should be copied to a directory with the same name as the application name under each target server's staging directory.

The external stage mode means that the application will be run from a staging directory, to which an external entity is expected to distribute the files. This mode is useful in environments that are managed by third-party tools.

Tasks

- “Designating a Default Web Application” on page 559-2
- “Deploying a New Web Application” on page 559-2
- “Testing the Deployment” on page 559-3
- “Viewing Deployed Web Applications” on page 559-4
- “Stopping Deployed Web Applications” on page 559-4
- “Adding or Editing Web Application Deployment Notes” on page 559-4
- “Deleting a Web Application” on page 559-5
- “Monitoring Web Applications and Servlets” on page 559-5
- “Viewing and Updating Run-Time Descriptor Elements” on page 559-6

Related Topics

[Developing Web Applications for WebLogic Server](#)

Attributes

Table 543-1

Attribute Label	Description	Value Constraints
Name	The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration. <i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean <i>Attribute:</i> Name	

Table 543-1

Attribute Label	Description	Value Constraints
Context Root	<p>The URI, located on the Administration Server, of the original source files for this Web application module.</p> <p><i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean</p> <p><i>Attribute:</i> ContextRoot</p>	
Path	<p>D:\bea81b\weblogic81\samples\server\config\examples\applications\DefaultWebApp</p> <p><i>MBean:</i> weblogic.management.configuration.ApplicationMBean</p> <p><i>Attribute:</i> Path</p>	
Load Order	<p>A numerical value that indicates when this module or application is deployed, relative to other deployable modules and applications. Modules with lower Load Order values are deployed before those with higher values.</p> <p><i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean</p> <p><i>Attribute:</i> LoadOrder</p>	<p><i>Default:</i> 100</p>
Staging Mode	<p>Indicates whether this application is being staged. Staging involves distributing the application files from the admin server to the targeted managed servers staging directory. This attribute is used to override the managed server's StagingMode attribute.</p> <p><i>MBean:</i> weblogic.management.configuration.ApplicationMBean</p> <p><i>Attribute:</i> StagingMode</p>	<p><i>Default:</i> null</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ nostage■ stage■ external_stage



Web Applications --> Configuration --> Other

[Tasks](#) [Related Topics](#) [Attribute](#)

Overview

Use this tab to specify additional configuration information about the Web application, such as the size of the pool used for single threaded mode instance pools, the authentication realm name, and whether the classes located in the WEB-INF directory should be loaded before classes with the same name located in the application or system classloader.

Tasks

“Designating a Default Web Application” on page 559-2

“Deploying a New Web Application” on page 559-2

“Testing the Deployment” on page 559-3

“Viewing Deployed Web Applications” on page 559-4

“Stopping Deployed Web Applications” on page 559-4

“Adding or Editing Web Application Deployment Notes” on page 559-4

“Deleting a Web Application” on page 559-5

“Monitoring Web Applications and Servlets” on page 559-5

“Viewing and Updating Run-Time Descriptor Elements” on page 559-6

Related Topics

[Developing Web Applications for WebLogic Server](#)

Attribute

Attribute Label	Description	Value Constraints
Single Threaded Servlet Pool Size	The size of the pool used for SingleThreadedModel instance pools.	<i>Default:</i> 5 <i>Configurable:</i> no
Auth Realm Name	The Realm in the Basic Authentication HTTP dialog box which pops up on the browsers <i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean <i>Attribute:</i> AuthRealmName	<i>Default:</i> "weblogic"
Prefer Web Inf Classes	If true, classes located in the WEB-INF directory of a web-app will be loaded in preference to classes loaded in the application or system classloader. Deprecated the setting from console starting from WLS7.1. Need to set it in weblogic.xml instead. <i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean <i>Attribute:</i> PreferWebInfClasses	<i>Default:</i> false <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false

Web Applications --> Configuration --> Files

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to specify how often the target server checks to see whether a servlet in the Web application has been modified, and whether it should be reloaded. Also use this tab to specify whether a target server should automatically generate an HTML directory listing if no suitable index file is found.

Tasks

“Designating a Default Web Application” on page 559-2

“Deploying a New Web Application” on page 559-2

“Testing the Deployment” on page 559-3

“Viewing Deployed Web Applications” on page 559-4

“Stopping Deployed Web Applications” on page 559-4

“Adding or Editing Web Application Deployment Notes” on page 559-4

“Deleting a Web Application” on page 559-5

“Monitoring Web Applications and Servlets” on page 559-5

“Viewing and Updating Run-Time Descriptor Elements” on page 559-6

Related Topics

[Developing Web Applications for WebLogic Server](#)

Attributes

Attribute Label	Description	Value Constraints
Reload Period	How often WebLogic checks whether a servlet has been modified, and if so reloads it. -1 is never reload, 0 is always reload <i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean <i>Attribute:</i> ServletReloadCheckSecs	<i>Default:</i> 1 <i>Configurable:</i> no
Index Directories	Indicates whether or not to automatically generate an HTML directory listing if no suitable index file is found	<i>Default:</i> false <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false <i>Configurable:</i> no

Web Application --> Deploy

This tab displays the target servers and deployment status for the selected Web Application, and lets you deploy and stop the application on configured targets.

Use the Web Application > Deploy tab to view the deployment status of the Web application module, and stop, deploy, or redeploy the Web application module. You can also stop or deploy this Web application module from or to all targets using the buttons shown beneath the Deployment Status by Target table. (To configure additional deployment targets for this Web application module, click the Targets tab.)

Tasks

“Designating a Default Web Application” on page 559-2

“Deploying a New Web Application” on page 559-2

“Testing the Deployment” on page 559-3

“Viewing Deployed Web Applications” on page 559-4

“Stopping Deployed Web Applications” on page 559-4

“Adding or Editing Web Application Deployment Notes” on page 559-4

“Deleting a Web Application” on page 559-5

“Monitoring Web Applications and Servlets” on page 559-5

“Viewing and Updating Run-Time Descriptor Elements” on page 559-6

Related Topics

For more information, see:

- “Web Applications” on page 559-1
- [Developing Web Applications for WebLogic Server](#)

-
- [Deploying WebLogic Server Applications](#)

Web Application --> Monitor --> Web Application

[Tasks](#) [Related Topics](#)

Overview

Use the tabs under the Monitoring tab as follows:

General—Use this tab to monitor general information about the server, such as all active queues, all server connections, and all active sockets.

Performance—Use this tab to monitor performance information about the server, such as idle threads, oldest pending request, throughput, queue length, and more.

Security—Use this tab to monitor security information about the server, such as invalid login attempts, user lockouts, login attempts while locked, and more.

JMS—Use this tab to monitor JMS information about the server, such as current connections, total connections, and more.

JTA—Use this tab to monitor JTA information about the server, such as total transactions, rollback information, and more.

Tasks

“Designating a Default Web Application” on page 559-2

“Deploying a New Web Application” on page 559-2

“Testing the Deployment” on page 559-3

“Viewing Deployed Web Applications” on page 559-4

“Stopping Deployed Web Applications” on page 559-4

“Adding or Editing Web Application Deployment Notes” on page 559-4

“Deleting a Web Application” on page 559-5

“Monitoring Web Applications and Servlets” on page 559-5

“Viewing and Updating Run-Time Descriptor Elements” on page 559-6

Related Topics

[Developing Web Applications for WebLogic Server](#)

Web Application --> Monitor --> Sessions

[Tasks](#) [Related Topics](#)

Overview

This page allows you to view statistics about all of the Web application modules that are currently active. You can also customize the information that is presented by clicking the [Customize this view...](#) link.

If the Web application is deployed as an exploded archive rather than as a WAR or part of an EAR, the session monitoring enabled check box appears. Use this to specify whether you want session monitoring enabled for this Web application.

Tasks

“Designating a Default Web Application” on page 559-2

“Deploying a New Web Application” on page 559-2

“Testing the Deployment” on page 559-3

“Viewing Deployed Web Applications” on page 559-4

“Stopping Deployed Web Applications” on page 559-4

“Adding or Editing Web Application Deployment Notes” on page 559-4

“Deleting a Web Application” on page 559-5

“Monitoring Web Applications and Servlets” on page 559-5

“Viewing and Updating Run-Time Descriptor Elements” on page 559-6

Related Topics

[Developing Web Applications for WebLogic Server](#)

Web Application --> Monitor --> Servlets

[Tasks](#) [Related Topics](#)

Overview

This tab displays a table that contains the list of servlets associated with this Web Application along with the following information about each servlet:

- the WebLogic Server instance with which it is associated
- the total number of times the servlet has been invoked since WebLogic Server started
- the average time it took to execute the servlet

Tasks

“Designating a Default Web Application” on page 559-2

“Deploying a New Web Application” on page 559-2

“Testing the Deployment” on page 559-3

“Viewing Deployed Web Applications” on page 559-4

“Stopping Deployed Web Applications” on page 559-4

“Adding or Editing Web Application Deployment Notes” on page 559-4

“Deleting a Web Application” on page 559-5

“Monitoring Web Applications and Servlets” on page 559-5

“Viewing and Updating Run-Time Descriptor Elements” on page 559-6

Related Topics

[Developing Web Applications for WebLogic Server](#)

Web Application --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to record notes about a Web Application's deployment configuration.

Tasks

“Designating a Default Web Application” on page 559-2

“Deploying a New Web Application” on page 559-2

“Testing the Deployment” on page 559-3

“Viewing Deployed Web Applications” on page 559-4

“Stopping Deployed Web Applications” on page 559-4

“Adding or Editing Web Application Deployment Notes” on page 559-4

“Deleting a Web Application” on page 559-5

“Monitoring Web Applications and Servlets” on page 559-5

“Viewing and Updating Run-Time Descriptor Elements” on page 559-6b

Related Topics

[Developing Web Applications for WebLogic Server](#)

Attributes

Table 550-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

Web Application Deployment Assistant -->

Step 2 - Select Targets

[Tasks](#) [Related Topics](#)

Overview

This page displays the available servers and clusters to which you can deploy a Web Application or Web Service.

To deploy to individual servers, select one or more server instances from the Independent Servers list and click Continue.

To deploy to a cluster of servers, select the name of the cluster from the Clusters list. By default, the Administration Console deploys a Web Application or Web Service to all server instances in the cluster (the All servers in the cluster option). If you want to deploy to only a subset of the servers in a cluster, select Parts of the cluster, and then select the individual server instances to which you want to deploy the application.

Click Continue to review your choices and deploy the application.

Tasks

[“Deploying a New Web Application” on page 559-2](#)

[“Testing the Deployment” on page 559-3](#)

[“Viewing Deployed Web Applications” on page 559-4](#)

[“Stopping Deployed Web Applications” on page 559-4](#)

[“Adding or Editing Web Application Deployment Notes” on page 559-4](#)

[“Deleting a Web Application” on page 559-5](#)

Related Topics

[Developing Web Applications for WebLogic Server](#)

Web Application --> Targets

[Tasks](#) [Related Topics](#)

Overview

In the Web Application Component > Targets tab, you select the independent servers, clusters, or virtual hosts on which you would like to deploy this Web application module at server startup time. You can reconfigure deployment targets later if you wish. To deploy or undeploy this Web application module immediately without restarting your server(s), click the Deploy tab.

Tasks

“Designating a Default Web Application” on page 559-2

“Deploying a New Web Application” on page 559-2

“Testing the Deployment” on page 559-3

“Viewing Deployed Web Applications” on page 559-4

“Stopping Deployed Web Applications” on page 559-4

“Adding or Editing Web Application Deployment Notes” on page 559-4

“Deleting a Web Application” on page 559-5

“Monitoring Web Applications and Servlets” on page 559-5

“Viewing and Updating Run-Time Descriptor Elements” on page 559-6

Related Topics

[Developing Web Applications for WebLogic Server](#)

Web Application --> Configuration --> Testing

[Tasks](#) [Related Topics](#)

Overview

In the Web Application Component > Testing tab, you can test the deployment of the Web Application component (WAR file) to make sure it was a successful one.

Tasks

“Designating a Default Web Application” on page 559-2

“Deploying a New Web Application” on page 559-2

“Testing the Deployment” on page 559-3

“Viewing Deployed Web Applications” on page 559-4

“Stopping Deployed Web Applications” on page 559-4

“Adding or Editing Web Application Deployment Notes” on page 559-4

“Deleting a Web Application” on page 559-5

“Monitoring Web Applications and Servlets” on page 559-5

“Viewing and Updating Run-Time Descriptor Elements” on page 559-6

Related Topics

[Developing Web Applications for WebLogic Server](#)



Web Application Deployment Assistant -->

Step 1 - Select Archive

[Tasks](#) [Related Topics](#)

Overview

The Web Application Deployment Assistant helps you deploy a new Web Application or Web Service (in `.war` format) to one or more servers in the domain. You can select either an archived web application (`.war` file), or an application in exploded archive format.

Use the links in the Location field to browse directories on the Administration Server machine and locate the Web Application or Web Service to deploy. If the application does not reside on the Administration Server machine, first use the upload link to upload the application `.war` file. This places the application archive in the Administration Server's configured upload directory, and automatically opens that directory in the Location field.

When the assistant detects a `.war` file or exploded `.war` file in the current directory, it lists the archive or directory name as a selection beneath the Location field. Select the name of the archive or directory you want to configure for deployment.

If your domain contains multiple WebLogic Server instances, click Target Module to target the application to a single server, or to multiple server instances or clusters.

In a single server domain, click Continue to automatically target the application to the available server instance.

Tasks

[“Deploying a New Web Application” on page 559-2](#)

[“Testing the Deployment” on page 559-3](#)

[“Viewing Deployed Web Applications” on page 559-4](#)
[“Stopping Deployed Web Applications” on page 559-4](#)
[“Adding or Editing Web Application Deployment Notes” on page 559-4](#)
[“Deleting a Web Application” on page 559-5](#)

Related Topics

[Developing Web Applications for WebLogic Server](#)

Web Application Deployment Assistant -->

Step 3 - Review Choices and Deploy

[Tasks](#) [Related Topics](#)

Overview

This page displays a list of the targeted clusters and servers for the Web Application or Web Service. Review the entries under the Deployment Targets heading. If you need to change a target, click your browser's Back button.

The Source accessibility header displays the selected staging mode for deploying the Web Application or Web Service source files:

- **Copy this application onto every target for me**—This option is selected by default if you targeted the application to a cluster or to multiple server instances. This corresponds to “stage” mode where the Administration Server copies the application files to each targeted server; and the target servers then deploy the application using their copy of the files.
- **I will make the application accessible from the following location**—This option is selected by default if you targeted the application to a single server instance. This corresponds to the “nostage” mode where the server deploys an application from a single directory; all targeted servers must be able to access the directory to deploy the application. Select this option if you are deploying to a cluster that resides on a single physical machine.

In the Identity header, the Name field specifies a unique name to refer to this Web Application or Web Service in the Administration Console. Accept the default name or enter a new name to describe the application.

Click Deploy to accept the values on this page and deploy the application to all specified server instances.

Tasks

- “Designating a Default Web Application” on page 559-2
- “Deploying a New Web Application” on page 559-2
- “Testing the Deployment” on page 559-3
- “Viewing Deployed Web Applications” on page 559-4
- “Stopping Deployed Web Applications” on page 559-4
- “Adding or Editing Web Application Deployment Notes” on page 559-4
- “Deleting a Web Application” on page 559-5
- “Monitoring Web Applications and Servlets” on page 559-5
- “Viewing and Updating Run-Time Descriptor Elements” on page 559-6

Related Topics

[Developing Web Applications for WebLogic Server](#)

Web Applications

The Web Applications page displays a list of Web Applications and Web Services (configured as `.war` files or exploded `.war` directories) that have been deployed in this domain. After you have deployed a Web Application to one or more servers in the domain, you can later deploy, redeploy, or stop the application by selecting its name on this page.

To deploy a new Web Application or Web Service on servers in this domain, click the Deploy a New Web Application link.

- For more information about configuring a Web Application or Web Service for deployment, see [“Deploying New Applications and Modules” on page 62-6](#).
- To change the order of automatic deployment for Web Applications or Web Services, see [“Changing the Order of Deployment” on page 62-6](#).



ServletRuntime

This table allows you to view and manage the Servlets you have deployed on this instance of WebLogic Server. It allows you to choose the attributes by which you want to sort the Servlets and to clone or delete selected Servlets.

To select attributes to be shown in the table Click the *Customize this View* link.

- For more information, see
- “Configuring and Deploying a New Enterprise Application or Web Service” on page 26-2



Web App Component Runtime

Use this table to manage the Web Applications deployed on this instance of WebLogic Server. It allows you to choose the attributes by which you want to sort the Web Applications and to clone or delete Web Applications.

To select attributes to be shown in the table, click the *Customize this View* link.

- For more information, see
- “Configuring and Deploying a New Enterprise Application or Web Service” on page 26-2



1 Web Applications

[“Attributes and Console Screen Reference for Web Applications” on page 560-1]

Overview

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.

These procedures describe how to use the Administration Console to deploy Web Applications.

Note: A Web service is a special kind of Web Application. Web services are Web applications that have an additional deployment descriptor called `web-services.xml`. If you package a Web service as a `.war` file or an exploded `.war` directory, you can deploy the Web service in the same manner as you would a Web application.

For more information about configuring Web Applications, see [Developing Web Applications for WebLogic Server](#). For more information about J2EE Enterprise applications in the Administration Console, see [Applications](#).

Tasks

For more information on application deployment, refer to “Deploying Applications and Modules” on page 62-1.

Designating a Default Web Application

The default Web Application is presented to clients who do not specify a URI (or specify "/" as the URI). To deploy a Web Application as a default Web Application, set the value of the `context-root` element to "/" in its deployment descriptor.

You can specify the `context-root` element in the `weblogic.xml` deployment descriptor for Web Applications that are packaged as a `.war` archive or exploded `.war` directory. If you package the Web Application as part of an Enterprise Application (`.ear` archive or exploded `.ear`), specify the `context-root` in `application.xml`. Note that the `application.xml` `context-root` takes precedent over the `weblogic.xml` value.

Deploy the application using the instructions in “Deploying a New Web Application” on page 559-2.

To deploy a Web Application as part of an `.ear` archive or exploded `.ear`, see [Deploying WebLogic Server Applications](#).

Deploying a New Web Application

To deploy a new Web application packaged as a WAR archive or exploded WAR directory using the WebLogic Server Administration Console:

1. Start the WebLogic Server Administration Console for the domain in which you will be working.
2. In the left pane of the Console, expand the Deployments folder, right-click Web Application Modules, and select Deploy a New Web Application Module. This initiates the Deployment Assistant.

3. Using the Deployment Assistant, locate the WAR file you would like to configure for use with WebLogic Server. You can also configure an "exploded" Web Application or component directory. Note that WebLogic Server will deploy all components it finds in and below the specified directory.
4. When you have located the archive file to configure, click Target Application.
5. If you have more than one server or cluster in your domain, select the one on which you want to deploy your new Web Application and click Continue. If you have just one server in your domain, go to the next step.
6. Enter a name for the Web Application in the Name field.

If you have more than one server or cluster in your domain, click whether you want to copy the file to each server.
7. Click Deploy. The Console displays the Deploy panel, which lists deployment status and deployment activities for the Web Application.
8. Using the available tabs, enter the following information:
 - Configuration—Define the general configuration of this Web Application.
 - Targets—Define the servers or clusters on which you would like to deploy all of the modules in this Web Application.
 - Deploy—View the deployment status of each module in Web Application, and to undeploy or redeploy individual modules.
 - Notes—Include any additional information that describes the configuration of this Web Application.
9. Click Apply.

Testing the Deployment

To test the deployment of a Web Application in the Administration Console:

1. In the left pane of the Console, expand Deployments. Then expand Web Application Modules and click the name of the application you want to test.
2. On the Testing tab, click the Web Application Module to ensure that it has been properly deployed.

Viewing Deployed Web Applications

To view deployed Web Applications in the Administration Console:

1. In the left pane of the Console, expand Deployments and click the Web Applications Modules folder name.
2. View a list of deployed Web Applications in the table displayed in the right side of the Console.

Stopping Deployed Web Applications

Stopping a Web Application makes the application unavailable to WebLogic Server clients. Stopping an application does not remove the deployment files from the server; you can later redeploy a stopped application by clicking its redeploy button in the Administration Console.

To stop a deployed Web Application:

3. In the left pane of the Console, expand Deployments and then Web Applications Modules.
4. Click the name of the Web Application you wish to stop.
5. Select the Deploy tab, and click Stop or Stop All buttons to stop the application.

Adding or Editing Web Application Deployment Notes

To add or edit Web Application deployment notes:

1. In the left pane of the Console, expand Deployments and then click the Web Application Modules folder name.
2. Click the name of the Web Application.
3. Select the Notes tab.
4. Add or edit the optional information in the Notes scroll box.

5. Click Apply.

Deleting a Web Application

To delete a Web Application:

1. In the left pane of the Console, expand Deployments and then click the Web Application Modules folder name. A table is displayed in the right pane of the Console showing all the deployed Web Applications.
2. Select the Configuration tab if it is not displayed.
3. Click the Garbage Can icon to the right of the file you want to delete.
4. Click Yes to confirm your decision.
5. Click Continue to return to the previous screen.

Monitoring Web Applications and Servlets

To monitor active Web Applications and Servlets in the Console, proceed as follows:

1. In the left pane of the Console, expand Deployments, right-click Web Application Modules, and select Monitor All Web Application Modules.
2. To monitor Web Applications, select the Web Applications sub-tab.
3. To monitor Servlets, select the Servlets sub-tab.
4. To enable Session Monitoring, select the Session sub-tab and select the Session Monitoring Enabled checkbox. Then click Apply.
5. On the Web Applications or Servlets sub-tab, you can select Customize This View to customize the available monitoring features. You can choose which items to monitor, determine how to sort these options, and set this view as your default view. Click Apply to save your settings.

Viewing and Updating Run-Time Descriptor Elements

This release of WebLogic Server has deprecated the Administration Console Deployment Descriptor Editor. A new Descriptor tab in the Administration Console has replaced it. Using the Descriptor tab, you can view, modify, and persist deployment descriptor elements to the descriptor file within an exploded Web application in the same manner that they were persisted using the Deployment Descriptor Editor.

These descriptor elements are engaged dynamically at runtime; you do not have to redeploy the Web Application. The descriptor elements contained in the Descriptor tab are limited to only those descriptor elements that may be dynamically changed at runtime.

To view and edit descriptor information in the Console, proceed as follows:

1. In the left pane of the Console, expand Deployments and then click the Web Application Modules folder name.
2. Click the name of the Web Application whose descriptors you want to change.
3. In the right pane, select Configuration and then select Descriptor
4. Click the link for the deployment descriptor you want to change.
5. Define the configuration of the application deployment descriptor file that is associated with this Web application module by changing the provided attribute values as needed.
6. Click Apply to save your changes.

Attributes and Console Screen Reference for Web Applications

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“Web Application --> Configuration --> General” on page 543-1](#)

[“Web Applications --> Configuration --> Other” on page 544-1](#)

[“Web Applications --> Configuration --> Files” on page 545-1](#)

[“Web Application --> Deploy” on page 546-1](#)

[“Web Application --> Configuration --> Descriptor” on page 542-1](#)

[“Web Application --> Notes” on page 550-1](#)

[“Web Application --> Monitor --> Web Application” on page 547-1](#)

[“Web Application --> Monitor --> Sessions” on page 548-1](#)

[“Web Application --> Monitor --> Servlets” on page 549-1](#)

[“Web Application --> Targets” on page 552-1](#)

[“Web Application --> Configuration --> Testing” on page 553-1](#)

[“Web App Component Runtime” on page 558-1](#)

[“Web Applications” on page 556-1](#)

[“Web Application Deployment Assistant --> Step 1 - Select Archive” on page 554-1](#)

[“Web Application Deployment Assistant --> Step 2 - Select Targets” on page 551-1](#)

[“Web Application Deployment Assistant --> Step 3 - Review Choices and Deploy” on page 555-1](#)



Web Service Component Runtime

Use this table to manage the Web services and Web applications deployed on this instance of WebLogic Server. It allows you to choose the attributes by which you want to sort the Web service and Web applications and to clone or delete them.

To select attributes to be shown in the table, click the *Customize this View* link.

For more information, see:

- “Deploying a New Web Application” on page 559-2.
- “Configuring and Deploying a New Web Service” on page 574-2



Web Services

The Web Services page displays a list of Web applications and Web services that are configured for deployment in this domain. After you have configured a Web application or Web service for deployment, you can deploy, redeploy, undeploy, or delete it using the Administration Console.

For more information, see:

- “Deploying a New Web Application” on page 559-2.
- “Configuring and Deploying a New Web Service” on page 574-2



Web Service --> Testing

[Tasks](#) [Related Topics](#)

Overview

Use the Web Service > Testing tab to test your Web Service.

Click on Launch Test Page link to go to the Web Service's Home Page, from which you can:

- View the WSDL that describes the service.
- Test each operation to ensure that it is working correctly.
As part of testing a Web service, you can edit the XML in the SOAP request that describes non-built-in data types to debug interoperability conflicts.
- View the SOAP request and response messages from a successful execution of an operation
- Possibly download the Web service-specific client JAR file that contains the interfaces, classes, and stubs needed to invoke the Web service from a client application.

Click the View This Service's Formal Definition to view the WSDL that describes the service.

Invoking a Secure WebLogic Web Service From Its Home Page

To invoke a secure WebLogic Web Service from its Home Page, you must first follow these steps:

1. Configure SSL for WebLogic Server.

For more information, see [Configuring the SSL Protocol at {DOCR00T}/secmanage/ssl.html](#).

-
2. Add the following flags to the script that starts up this instance of WebLogic Server:

```
-Dweblogic.webservice.client.BaseWLSAdapter.strictCertChecking=false  
-Dweblogic.security.SSL.ignoreHostnameVerification=true
```

3. Create a certificate, key, and trusted CA and configure WebLogic Server to use them.

For more information, see [Configuring the SSL Protocol at {DOCR00T}/secmanage/ssl.html](#).

4. Restart WebLogic Server for the startup flags to take effect.
5. Invoke the secure WebLogic Web service's Home Page in your browser. The browser will return a message saying the certificate is not trusted.
6. Load the trusted certificate in your browser. You may need to restart your browser for it to take effect.
7. Invoke the secure WebLogic Web Service's Home Page again in your browser. You should now be able to test your secure Web Service.

Tasks

“Configuring and Deploying a New Web Service” on page 574-2

“Viewing Deployed Web Services” on page 574-4

“Undeploying Deployed Web Services” on page 574-5

“Deleting a Web Service” on page 574-5

“Viewing Web Service Deployment Descriptors” on page 574-6

“Configuring Web Service Reliable SOAP Messaging” on page 574-8

Related Topics

[The WebLogic Web Services Home Page and WSDL URLs](#)

Web Service --> Targets

[Tasks](#) [Related Topics](#)

Overview

In the Web Service > Targets tab, you select the independent servers, clusters, or virtual hosts on which you would like to deploy this Web service at server startup time. You can reconfigure deployment targets later if you wish. To deploy or undeploy this Web service immediately without restarting your server(s), click the Deploy tab.

Tasks

“Configuring and Deploying a New Web Service” on page 574-2

“Viewing Deployed Web Services” on page 574-4

“Undeploying Deployed Web Services” on page 574-5

“Deleting a Web Service” on page 574-5

“Viewing Web Service Deployment Descriptors” on page 574-6

“Configuring Web Service Reliable SOAP Messaging” on page 574-8

Related Topics

- [Developing Web Applications for WebLogic Server](#)
- [Programming WebLogic Web Services](#)



Web Service --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to record notes about a Web service's deployment configuration.

Tasks

“Configuring and Deploying a New Web Service” on page 574-2

“Viewing Deployed Web Services” on page 574-4

“Undeploying Deployed Web Services” on page 574-5

“Deleting a Web Service” on page 574-5

“Viewing Web Service Deployment Descriptors” on page 574-6

“Configuring Web Service Reliable SOAP Messaging” on page 574-8

Related Topics

- [Developing Web Applications for WebLogic Server](#)
- [Programming WebLogic Web Services](#)

Attributes

Table 565-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

Web Service --> Monitoring --> Web Services

[Tasks](#) [Related Topics](#)

Overview

This tab displays a table that contains the list of active Web services and the following information about each Web service:

- the context root of the Web service.
- the number of servlets that are associated with the Web service
- the current total number of open sessions for this Web service
- the largest number of sessions that were opened all at once since this server was started
- the total number of sessions since the server was started.

Tasks

“Configuring and Deploying a New Web Service” on page 574-2

“Viewing Deployed Web Services” on page 574-4

“Undeploying Deployed Web Services” on page 574-5

“Deleting a Web Service” on page 574-5

“Viewing Web Service Deployment Descriptors” on page 574-6

“Configuring Web Service Reliable SOAP Messaging” on page 574-8

Related Topics

- [Developing Web Applications for WebLogic Server](#)
- [Programming WebLogic Web Services](#)

Web Service --> Monitoring --> Sessions

[Tasks](#) [Related Topics](#)

Overview

Use this tab to specify whether you want session monitoring enabled for this Web service.

Tasks

“Configuring and Deploying a New Web Service” on page 574-2

“Viewing Deployed Web Services” on page 574-4

“Undeploying Deployed Web Services” on page 574-5

“Deleting a Web Service” on page 574-5

“Viewing Web Service Deployment Descriptors” on page 574-6

“Configuring Web Service Reliable SOAP Messaging” on page 574-8

Related Topics

- [Developing Web Applications for WebLogic Server](#)
- [Programming WebLogic Web Services](#)



Web Service --> Monitoring --> Servlets

[Tasks](#) [Related Topics](#)

Overview

This tab displays a table that contains the list of servlets associated with this Web service along with the following information about each servlet:

- the WebLogic Server instance with which it is associated
- the total number of times the servlet has been invoked since WebLogic Server started
- the average time it took to execute the servlet

Tasks

“Configuring and Deploying a New Web Service” on page 574-2

“Viewing Deployed Web Services” on page 574-4

“Undeploying Deployed Web Services” on page 574-5

“Deleting a Web Service” on page 574-5

“Viewing Web Service Deployment Descriptors” on page 574-6

“Configuring Web Service Reliable SOAP Messaging” on page 574-8

Related Topics

- [Developing Web Applications for WebLogic Server](#)
- [Programming WebLogic Web Services](#)



Web Service --> Deploy

Use the Web Service > Deploy tab to view the deployment status of the Web service module, and stop, deploy, or redeploy the Web service. You can also stop or deploy this Web service from or to all targets using the buttons shown beneath the Deployment Status by Target table. (To configure additional deployment targets for this Web service, click the Targets tab.)

Tasks

“Configuring and Deploying a New Web Service” on page 574-2

“Viewing Deployed Web Services” on page 574-4

“Undeploying Deployed Web Services” on page 574-5

“Deleting a Web Service” on page 574-5

“Viewing Web Service Deployment Descriptors” on page 574-6

“Configuring Web Service Reliable SOAP Messaging” on page 574-8

Related Topics

For more information, see:

- “Web Applications” on page 559-1
- “Configuring and Deploying a New Web Service” on page 574-2
- [Developing Web Applications for WebLogic Server](#)
- Deployment Procedures



Web Service --> Configuration --> Other

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to specify additional configuration information about the Web service, such as the size of the pool used for single threaded mode instance pools, the authentication realm name, and whether the classes located in the WEB-INF directory should be loaded before classes with the same name located in the application or system classloader.

Tasks

“Configuring and Deploying a New Web Service” on page 574-2

“Viewing Deployed Web Services” on page 574-4

“Undeploying Deployed Web Services” on page 574-5

“Deleting a Web Service” on page 574-5

“Viewing Web Service Deployment Descriptors” on page 574-6

“Configuring Web Service Reliable SOAP Messaging” on page 574-8

Related Topics

[Configuring Web Application Components](#)

Attributes

Table 570-1

Attribute Label	Description	Value Constraints
Single Threaded Servlet Pool Size	The size of the pool used for SingleThreadedModel instance pools. <i>MBean:</i> weblogic.management.configuration.WebServiceComponentMBean <i>Attribute:</i> SingleThreadedServletPoolSize	<i>Default:</i> 5
Auth Realm Name	The Realm in the Basic Authentication HTTP dialog box, which pops up on the browsers. <i>MBean:</i> weblogic.management.configuration.WebServiceComponentMBean <i>Attribute:</i> AuthRealmName	<i>Default:</i> "weblogic"
Prefer Web Inf Classes	If true, classes located in the WEB-INF directory of a web-app will be loaded in preference to classes loaded in the application or system classloader. Deprecated the setting from console starting from WLS7.1. Need to set it in weblogic.xml instead. <i>MBean:</i> weblogic.management.configuration.WebServiceComponentMBean <i>Attribute:</i> PreferWebInfClasses	<i>Default:</i> false <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false

Web Service --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab describes the general configuration information for the deployed Web service. In particular, it describes:

- The name of the Web application WAR file which contains the `web-services.xml` file.
- The context root of the Web service, used when invoking the Web service.
- The full pathname of the Web application WAR file.

Tasks

“Configuring and Deploying a New Web Service” on page 574-2

“Viewing Deployed Web Services” on page 574-4

“Undeploying Deployed Web Services” on page 574-5

“Deleting a Web Service” on page 574-5

“Viewing Web Service Deployment Descriptors” on page 574-6

“Configuring Web Service Reliable SOAP Messaging” on page 574-8

Related Topics

[Developing Web Applications for WebLogic Server](#)

Attributes

Table 571-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean</p> <p><i>Attribute:</i> Name</p>	
URI	<p>Return a URI pointing to the application, usually on the Admin Server.</p> <p><i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean</p> <p><i>Attribute:</i> URI</p>	
Load Order	<p>Specifies the order applications are loaded at server startup. Applications with the lowest values are loaded first.</p> <p>Application ordering is only supported for applications deployed with the 2 phase protocol.</p> <p><i>MBean:</i> weblogic.management.configuration.ApplicationMBean</p> <p><i>Attribute:</i> LoadOrder</p>	<i>Default:</i> 100

Table 571-1

Attribute Label	Description	Value Constraints
Deployment Order	<p>A priority that the server uses to determine when it deploys an item. The priority is relative to other deployable items of the same type. For example, the server prioritizes and deploys all EJBs before it prioritizes and deploys startup classes.</p> <p>Items with the lowest Deployment Order value are deployed first. There is no guarantee on the order of deployments with equal Deployment Order values. There is no guarantee of ordering across clusters.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.WebAppComponentMBean</code></p> <p><i>Attribute:</i> <code>DeploymentOrder</code></p>	<p><i>Minimum:</i> <code>MIN_ORDER</code></p> <p><i>Maximum:</i> <code>MAX_ORDER</code></p> <p><i>Default:</i> <code>DEFAULT_ORDER</code></p> <p><i>Dynamic:</i> yes</p>
Staging Mode	<p>Indicates whether this application is being staged. Staging involves distributing the application files from the admin server to the targeted managed servers staging directory. This attribute is used to override the managed server's <code>StagingMode</code> attribute.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.ApplicationMBean</code></p> <p><i>Attribute:</i> <code>StagingMode</code></p>	<p><i>Default:</i> null</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ <code>nostage</code>■ <code>stage</code>■ <code>external_stage</code>



Web Service --> Configuration --> Files

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to specify how often the target server checks to see whether a servlet in the Web service has been modified, and whether it should be reloaded. Also use this tab to specify whether a target server should automatically generate an HTML directory listing if no suitable index file is found.

Tasks

“Configuring and Deploying a New Web Service” on page 574-2

“Viewing Deployed Web Services” on page 574-4

“Undeploying Deployed Web Services” on page 574-5

“Deleting a Web Service” on page 574-5

“Viewing Web Service Deployment Descriptors” on page 574-6

“Configuring Web Service Reliable SOAP Messaging” on page 574-8

Related Topics

[Configuring Web Application Components](#)

Attributes

Table 572-1

Attribute Label	Description	Value Constraints
Reload Period	How often WebLogic checks whether a servlet has been modified, and if so reloads it. -1 is never reload, 0 is always reload. <i>MBean:</i> weblogic.management.configuration.WebServiceComponentMBean <i>Attribute:</i> ServletReloadCheckSecs	<i>Default:</i> 1
Index Directories	Indicates whether or not to automatically generate an HTML directory listing if no suitable index file is found. <i>MBean:</i> weblogic.management.configuration.WebServiceComponentMBean <i>Attribute:</i> IndexDirectoryEnabled	<i>Default:</i> false <i>Valid values:</i> <ul style="list-style-type: none">■ true■ false

Web Service --> Configuration --> Descriptor

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

In the Web Services > Configuration > Descriptor tab, you can configure some of the elements and attributes of the Web application deployment descriptors (`web.xml` and `weblogic.xml`) that are associated with this Web service.

Note: A Web service is a special kind of Web Application. Web services are Web applications that have an additional deployment descriptor called `web-services.xml`. You install, configure, and deploy Web services the same as Web applications.

Additionally, using the links at the bottom of the tab, you can view in read-only mode the full deployment descriptor files associated with this Web service:

- `web.xml` (J2EE Web application deployment descriptor)
- `weblogic.xml` (Weblogic-specific Web application deployment descriptor)
- `web-services.xml` (WebLogic-specific Web services deployment descriptor)

Tasks

“Configuring and Deploying a New Web Service” on page 574-2

“Viewing Deployed Web Services” on page 574-4

“Undeploying Deployed Web Services” on page 574-5

“Deleting a Web Service” on page 574-5

“Viewing Web Service Deployment Descriptors” on page 574-6

“Configuring Web Service Reliable SOAP Messaging” on page 574-8

Related Topics

- [web.xml Deployment Descriptor Elements](#)
- [weblogic.xml Deployment Descriptor Elements](#)
- [WebLogic Web Service Deployment Descriptor Elements](#)
- [Configuring Web Application Components](#)

Attributes

Table 573-1

Attribute Label	Description	Value Constraints
Session Cookie Max Age Secs	The maximum age allowed for a session cookie. NEED MORE INFO. <i>MBean:</i> weblogic.management. configuration. WebAppComponentMBean <i>Attribute:</i> SessionCookieMaxAgeSecs	<i>Units:</i> seconds
Session Invalidation Interval Secs	The amount of time (in seconds) that is allowed between sweeps for invalid sessions. <i>MBean:</i> weblogic.management. configuration. WebAppComponentMBean <i>Attribute:</i> SessionInvalidationIntervalSe cs	<i>Units:</i> seconds

Table 573-1

Attribute Label	Description	Value Constraints
Session Timeout Secs	<p>The amount of time (in seconds) that a session can go unused before it is invalidated.</p> <p><i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean</p> <p><i>Attribute:</i> SessionTimeoutSecs</p>	<i>Units:</i> seconds
Reload Period	<p>The amount of time (in seconds) representing how often the target checks whether a servlet has been modified, and if so, reloads it. -1 means never reload and 0 means always reload.</p> <p><i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean</p> <p><i>Attribute:</i> ServletReloadCheckSecs</p>	<i>Units:</i> seconds
Single Threaded Servlet Pool Size	<p>The size of the pool used for single threaded mode instance pools.</p> <p><i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean</p> <p><i>Attribute:</i> SingleThreadedServletPoolSize</p>	
Index Directories	<p>Specifies whether the target should automatically generate an HTML directory listing if no suitable index file is found.</p> <p><i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean</p> <p><i>Attribute:</i> IndexDirectoryEnabled</p>	

Table 573-1

Attribute Label	Description	Value Constraints
Enable Session Monitoring	Specifies whether runtime MBeans will be created for sessions. <i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean <i>Attribute:</i> SessionMonitoringEnabled	
Compile Command	<i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean <i>Attribute:</i> JSPCompileCommand	
Reload Period	The amount of time (in seconds) representing how often the target checks whether a servlet or JSP has been modified, and if so, reloads it. -1 means never reload and 0 means always reload. <i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean <i>Attribute:</i> JSPPageCheckSecs	<i>Units:</i> seconds
Keep Generated Source Files	Specifies whether to keep the generated source files or delete them after compiling a JSP. <i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean <i>Attribute:</i> JSPKeepGenerated	
Verbose	Enables the JSP compiler's verbose output option. <i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean <i>Attribute:</i> JSPVerbose	<i>Default:</i> false

Table 573-1

Attribute Label	Description	Value Constraints
Enable JSP Line Numbers	Compiles JSPs with the Java compiler's debug option enabled. <i>MBean:</i> weblogic.management.configuration.WebAppComponentMBean <i>Attribute:</i> JSPDebug	<i>Default:</i> false



1 Web Services


[“Attributes and Console Screen Reference for Web Services” on page 575-1]

Overview

Web services are a type of service that can be shared by and used as components of distributed Web-based applications. They commonly interact with existing back-end applications, such as customer relationship management systems, order-processing systems, and so on.

Typically, WebLogic Web services are assembled, packaged, and deployed as Enterprise Application files. The *.ear file contains a special Web application file (which contains an additional Web service-specific deployment descriptor file, `web-services.xml`) along with an optional EJB file. The *.jar contains the stateless session EJBs that implement the Web service. The `servicegen` Ant task, the standard way of assembling a WebLogic Web service, assembles all Web services as EAR files.

You can, however, also package a Web service as just a Web application WAR file if your Web service is implemented with a Java class.

Note: The Administration Console uses the  icon to indicate the special Web application WAR file that contains the additional Web service-specific deployment descriptor file `web-services.xml`.

The following procedures show how to deploy and configure Web services that are packaged either as *.ear files or as *.war files.

For additional information about Web services, see [Programming WebLogic Web Services](#).

Tasks

Configuring and Deploying a New Web Service

Web services can be packaged as Enterprise Application *.ear files (most typically) or as Web application *.war files. The following two procedures describe how to configure and deploy a Web service, depending on how you have packaged it.

Configuring and Deploy a Web Service Packaged as an Enterprise Application File (EAR)

1. In the left pane of the Administration Console, expand the Deployments folder.
2. Right-click Applications, and select Deploy a New Application.
3. Use the location field to locate the Web service *.ear file you want to configure for use with WebLogic Server. You can also configure an "exploded" application directory. WebLogic Server deploys all components it finds in and below the specified directory.

If the *.ear does not reside on the Administration Server, use the upload link to upload it.

Note: The Administration Console upload functionality helps you upload a single archive file to the Administration Server machine for deployment. If you need to upload an exploded archive directory, see [Weblogic.Deployer Tasks at {DOCROOT}/deployment/scenarios.html](#).

4. When you have located the archive file to configure, click Target Application.
5. If you have more than one server or cluster in your domain, select the one on which you want to deploy your new Web service and click Continue. If you have just one server in your domain, go to the next step.
6. Enter a name for the Web service in the Name field.

If you have more than one server or cluster in your domain, click in the Source Accessibility section whether you want to copy the file to each server.

7. Click Deploy. The Deploy panel displays deployment status and deployment activities for the Web service.
8. Use the available tabs as follows:
 - Configuration—Define the general configuration of this Web service.
 - Targets—Define additional servers or clusters on which you would like to deploy all of the modules in this Web service.
 - Deploy—View the deployment status of each module in Web service, and to undeploy or redeploy individual modules.
 - Notes—Include any additional information that describes the configuration of this Web service.
9. Click Apply.

Configuring and Deploying a Web Service Packaged as a Web Application File (WAR)

1. In the left pane of the Administration Console, expand the Deployments folder.
2. Right-click Web Application Modules, and select Deploy a New Web Application Module.
3. Use the location field to locate the Web service *.war file you want to configure for use with WebLogic Server. You can also configure an "exploded" Web application directory. WebLogic Server deploys all components it finds in and below the specified directory.

If the *.war does not reside on the Administration Server, use the upload link to upload it.

Note: The Administration Console upload functionality helps you upload a single archive file to the Administration Server machine for deployment. If you need to upload an exploded archive directory, see [Weblogic.Deployer Tasks at {DOCROOT}/deployment/scenarios.html](#).


4. When you have located the archive file to configure, click Target Module.
5. If you have more than one server or cluster in your domain, select the one on which you want to deploy your new Web service and click Continue. If you have just one server in your domain, go to the next step.

6. Enter a name for the Web service in the Name field.
If you have more than one server or cluster in your domain, click in the Source Accessibility section whether you want to copy the file to each server.
7. Click Deploy. The Deploy panel displays deployment status and deployment activities for the Web Service.
8. Use the available tabs as follows:
 - Configuration—Define the general configuration of this Web service.
 - Targets—Select additional independent servers, clusters, or virtual hosts on which you would like to deploy the Web service at server startup time.
 - Deploy—View the deployment status of this Web service module, and undeploy, deploy, or redeploy the Web service.
 - Monitoring—Monitor the Web services that are currently active.
 - Testing—Invoke the Web Service Home Page and test it.
 - Notes—Include any additional information that describes the configuration of this Web service.
9. Click Apply to apply all relevant changes.

Viewing Deployed Web Services


Web services can be packaged as Enterprise Application *.ear files (most typically) or as Web application *.war files. The following procedure describes how to view deployed Web services, depending on how you have packaged them.

Viewing a Deployed Web Service Packaged as an Enterprise Application File (EAR)

1. In the left pane of the Administration Console, expand the Deployments folder.
2. Expand the Applications folder.
3. Click the name of the deployed applications until you find one that includes a Web service *.war file component, indicated by the  icon.

4. Click the name of the Web Service. Information about the Web Service appears in the right frame.

Viewing a Deployed Web Service Packaged as a Web Application File (WAR)

1. In the left pane of the Administration Console, expand the Deployments folder.
2. Expand the Web Application Modules folder. The deployed Web services are indicated with the  icon.
3. Click the name of the Web Service. Information about the Web Service appears in the right frame.

Undeploying Deployed Web Services

Web services can be packaged as Enterprise Application *.ear files (most typically) or as Web application *.war files. The following procedure describes how to undeploy a deployed Web service, depending on how you have packaged it.

1. In the left pane of the Administration Console, expand the Deployments folder.
2. Click on the Applications folder name if the Web Service is packaged as an *.ear file or on the Web Application Modules folder name if the Web Service is packaged as a *.war file.
3. In the displayed table, click the name of the Web Service you want to undeploy.
4. Select the Deploy tab.
5. Click Stop Application if your Web Service is packaged as an *.ear file or Stop if your Web Service is packaged as a *.war file.

Deleting a Web Service

Web services can be packaged as Enterprise Application *.ear files (most typically) or as Web application *.war files. The following procedure describes how to delete a Web service, depending on how you have packaged it.


1. In the left pane of the Administration Console, expand the Deployments folder.
2. Click on the Applications folder name if the Web Service is packaged as an *.ear file or on the Web Application Modules folder name if the Web Service is packaged as a *.war file. A table displays all deployed Enterprise or Web applications.
3. In this table, locate the Web service you want to delete and click the Garbage Can icon to the right of the file.
4. Click Yes to confirm your decision.
5. Click Continue to return to the previous screen.

Viewing Web Service Deployment Descriptors

Web services can be packaged as Enterprise Application *.ear files (most typically) or as Web application *.war files. The following two procedures describe how to view (in read-only mode) the web-services.xml deployment descriptor file of a Web service, depending on how you have packaged it.


Note: You cannot use the Administration Console to update the web-services.xml file.

Viewing Deployment Descriptors for a Web Service Packaged as an Enterprise Application File (EAR)

1. In the left pane of the Administration Console, expand the Deployments—Applications folders.
2. Expand the folder for the Web service for which you want to view the web-services.xml deployment descriptor file.
3. Under the Web service *.ear file, click the name of the *.war file that contains the web-services.xml file, indicated with the  icon.
4. In the right pane, select the Configuration—Descriptor tabs.

5. Under the Deployment Descriptors section at the bottom of the right pane, click the `web-services.xml` link. You can also view the `web.xml` and `weblogic.xml` Web application deployment descriptors by clicking on the corresponding link.

Viewing Deployment Descriptors for a Web Service Packaged as a Web Application File (WAR)

1. In the left pane of the Administration Console, expand the Deployments→Web Applications Modules folders.
2. Click the name of the Web service for which you want to view the `web-services.xml` deployment descriptor file. Web services are indicated with the  icon.
3. In the right pane, select the Configuration→Descriptor tabs.
4. Under the Deployment Descriptors section at the bottom of the right pane, click the `web-services.xml` link. You can also view the `web.xml` and `weblogic.xml` Web application deployment descriptors by clicking on the corresponding link.

Testing a Web Service From Its Home Page

Every Web Service deployed on WebLogic Server has a Home Page. From the Home page you can:

- View the WSDL that describes the service.
- Test each operation with sample parameter values to ensure that it is working correctly.
- View the SOAP request and response messages from a successful execution of an operation.

The following procedure describes how to view the Home Page of a Web Service and use it to test its operations.

1. Follow the instructions in “Viewing Deployed Web Services” on page 574-4 to view information about a particular Web Service.
2. In the right frame, select the Testing tab.
3. Click the Launch Test Page link. The Web Service Home Page displays in a new browser window.
4. To test a particular Web Service operation:
 - a. Click on the operation link.
 - b. Enter sample values for the parameters in the table. The first two columns of the table list the name and Java data type of the operation.
 - c. Click Invoke.

The SOAP request and response messages and the value returned by the operation are displayed in a new browser window.

The main Web Service Home Page also displays an example of the Java code to invoke one of the operations and a sample `build.xml` file for executing the `clientgen` Ant task to generate the Web Service-specific client JAR file.

Configuring Web Service Reliable SOAP Messaging

Reliable SOAP messaging is a framework whereby an application running in one WebLogic Server instance can asynchronously and reliably invoke a Web Service running on another WebLogic Server instance.

Note: Reliable SOAP messaging also works between two Web Services deployed on a single WebLogic Server instance. Typically this setup is used for development. However, in real-life, reliable SOAP messaging is meant to be used between *two* WebLogic Server instances, both of which must be configured to use reliable SOAP messaging.

There is a *sender* WebLogic Server instance and a *receiver*. The sender has an application that asynchronously invokes a reliable Web Service operation running on the receiver. The sender sends the receiver a SOAP message that has reliable SOAP messaging information in its header. The Web Service operation being invoked has been configured for reliable SOAP messaging. Due to the asynchronous nature of the

invocation, the sender does not immediately know whether the relevant operation has been invoked, but it has the guarantee that it will get one of two possible notifications:

- The message has been received by the receiver.

Note: This does not mean that the Web Service operation on the receiver WebLogic Server was *successfully* invoked; the operation might fail due to an application exception. The exception will be included in the notification to the sender.

- The sender was unable to deliver the message.

You can configure the current WebLogic Server as both a sender and a receiver of reliable SOAP messages, as described in:

- “Configuring the Sender WebLogic Server Instance” on page 574-9
- “Configuring the Receiver WebLogic Server Instance” on page 574-10

For additional information about reliable SOAP messaging, see [Using Reliable SOAP Messaging at {DOCROOT}/webserv/reliable.html](#).

Configuring the Sender WebLogic Server Instance

This section describes how to configure default reliable SOAP messaging attributes for a WebLogic Server instance in its role as a sender of a reliable SOAP message.

1. In the left pane of the Administration Console, expand the Servers node.
2. Click the name of the WebLogic Server instance for which you want to configure reliable SOAP messaging in its role as a sender.
3. In the right pane, select the Services—Web Services tabs.
4. Select the JMS store from the Store drop-down list that contains WebLogic Server’s reliable SOAP messages when acting as a sender.

You must first create a JMS store before it appears in the drop-down list. For details, see “JMS Store Tasks” on page 232-22 and “JMS JDBC Store Tasks” on page 232-25.

5. In the Default Retry Count field, enter the default maximum number of times the sender WebLogic Server should attempt to resend a message.

6. In the Default Retry Interval field, enter the default minimum number of seconds that the sender WebLogic Server should wait between retries.
7. In the Default Time to Live field, enter the default minimum number of seconds that the receiver of the reliable SOAP message should persist its message in the receiver's persistent JMS store.

Warning: Do not set this value larger than the corresponding value of any Web service operation being invoked reliably. This value is configured in the Web service's `web-services.xml` file, in particular the `persist-duration` attribute of the `<reliable-delivery>` subelement of the invoked `<operation>`.

8. Click Apply.

Configuring the Receiver WebLogic Server Instance

This section describes how to configure default reliable SOAP messaging attributes for a WebLogic Server instance in its role as a receiver of a reliable SOAP message.

1. In the left pane of the Administration Console, expand the Servers node.
2. Click the name of the WebLogic Server instance for which you want to configure reliable SOAP messaging in its role as a receiver.
3. In the right pane, select the Services—Web Services tabs.
4. Select the JMS store from the Store drop-down list that will contain WebLogic Server's reliable SOAP messages when acting as a receiver.

You must first create a JMS store before it appears in the drop-down list. For details, see “JMS Store Tasks” on page 232-22 and “JMS JDBC Store Tasks” on page 232-25.

5. Enter the default minimum number of seconds that the receiver of the reliable SOAP message should persist its message in the receiver's persistent JMS store in the Default Time to Live field.

Note: Each Web service operation can override this default value by setting the `persist-duration` of the `<reliable-delivery>` subelement of the corresponding `<operation>` element.

6. Click Apply.

Attributes and Console Screen Reference for Web Services

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“Web Service --> Configuration --> Descriptor” on page 573-1](#)

[“Web Service --> Configuration --> Files” on page 572-1](#)

[“Web Service --> Configuration --> General” on page 571-1](#)

[“Web Service --> Configuration --> Other” on page 570-1](#)

[“Web Service --> Deploy” on page 569-1](#)

[“Web Service --> Monitoring --> Servlets” on page 568-1](#)

[“Web Service --> Monitoring --> Sessions” on page 567-1](#)

[“Web Service --> Monitoring --> Web Services” on page 566-1](#)

[“Web Service --> Notes” on page 565-1](#)

[“Web Service --> Targets” on page 564-1](#)

[“Web Service --> Testing” on page 563-1](#)

[“Web Service Component Runtime” on page 561-1](#)

[“Web Services” on page 562-1](#)



WLEC Connection Pool --> Configuration --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to configure a WLEC connection pool.

Tasks

“Configure a New WLEC Connection Pool” on page 582-2

Related Topics

Using WebLogic Enterprise Connectivity at [{DOCROOT}/wlec/index.html](#)

Attributes

Table 576-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> weblogic.management.configuration.WLECCConnectionPoolMBean</p> <p><i>Attribute:</i> Name</p>	
Primary Addresses	<p>The list of addresses for IIOP Listener/Handlers used to establish a connection between the WLEC connection pool and the Tuxedo domain.</p> <ul style="list-style-type: none">■ The format of each address is //hostname:port.■ The addresses must match the ISL addresses defined in the UBBCONFIG file. Multiple addresses are separated by commas. 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. Example: corbalocs://hostname:port. <p><i>MBean:</i> weblogic.management.configuration.WLECCConnectionPoolMBean</p> <p><i>Attribute:</i> PrimaryAddresses</p>	

Table 576-1

Attribute Label	Description	Value Constraints
Failover Addresses	<p>The list of addresses for IIOP Listener/Handlers used if connections defined in the primary addresses cannot be established or fail. Multiple addresses are separated by commas.</p> <p><i>MBean:</i> weblogic.management.configuration.WLECCConnectionPoolMBean</p> <p><i>Attribute:</i> FailoverAddresses</p>	
Domain	<p>The name of the WLEC domain to which the pool is connected.</p> <ul style="list-style-type: none">■ You can have only one WLEC connection pool per Tuxedo domain.■ The domain name must match the domainid parameter in the RESOURCES section of the UBBCONFIG file for the Tuxedo domain. <p><i>MBean:</i> weblogic.management.configuration.WLECCConnectionPoolMBean</p> <p><i>Attribute:</i> WLEDomain</p>	
Minimum Pool Size	<p>The number of IIOP connections to be added to the WLEC connection pool when WebLogic Server starts.</p> <p><i>MBean:</i> weblogic.management.configuration.WLECCConnectionPoolMBean</p> <p><i>Attribute:</i> MinimumPoolSize</p>	<i>Default:</i> 1

Table 576-1

Attribute Label	Description	Value Constraints
Maximum Pool Size	The maximum number of IIOP connections that can be made from the WLEC connection pool. <i>MBean:</i> weblogic.management.configuration.WLECCConnectionPoolMBean <i>Attribute:</i> MaximumPoolSize	<i>Default:</i> 1

WLEC Connection Pool --> Configuration --> Security

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Using WebLogic Enterprise Connectivity, a security context established in WebLogic Server can be used to establish a security identify in a BEA Tuxedo domain. In a WLEC connection pool, each network connection has been authenticated through a User identity that is defined by the system administrator of WebLogic Server. You can use either password or certificate authentication to establish a WLEC connection pool.

Tasks

“Configuring User Security” on page 582-4

Related Topics

[Using WebLogic Enterprise Connectivity at {DOCROOT}/wlec/index.html](#)

Attributes

Table 577-1

Attribute Label	Description	Value Constraints
User Name	<p>The name of a qualified user. This field is required only when the security level in the Tuxedo domain is USER_AUTH, ACL or MANDATORY_ACL.</p> <p><i>MBean:</i> weblogic.management.configuration.WLECCConnectionPoolMBean</p> <p><i>Attribute:</i> UserName</p>	
User Password	<p>The password of the qualified user specified in the User Name field. This field is required only when you define the User Name field.</p> <p><i>MBean:</i> weblogic.management.configuration.WLECCConnectionPoolMBean</p> <p><i>Attribute:</i> UserPassword</p>	<i>Encrypted:</i> yes
User Role	<p>The user role for this connection pool. This field is required when the security level in the Tuxedo domain is APP_PW, USER_AUTH, ACL, or MANDATORY_ACL.</p> <p><i>MBean:</i> weblogic.management.configuration.WLECCConnectionPoolMBean</p> <p><i>Attribute:</i> UserRole</p>	

Table 577-1

Attribute Label	Description	Value Constraints
Application Password	<p>The password for the application. This field is required when the security level in the Tuxedo domain is APP_PW, USER_AUTH, ACL, or MANDATORY_ACL.</p> <p><i>MBean:</i> weblogic.management.configuration.WLECCConnectionPoolMBean</p> <p><i>Attribute:</i> ApplicationPassword</p>	<i>Encrypted:</i> yes
Minimum Encryption Level	<p>The minimum SSL encryption level used between the Tuxedo domain and WebLogic Server.</p> <ul style="list-style-type: none">■ Zero (0) indicates that the data is signed but not sealed.■ 40, 56, and 128 specify the length, in bits, of the encryption key.■ Default value is 40.■ If this minimum level of encryption is not met, the SSL connection between Tuxedo and WebLogic Server fails. <p><i>MBean:</i> weblogic.management.configuration.WLECCConnectionPoolMBean</p> <p><i>Attribute:</i> MinimumEncryptionLevel</p>	

Table 577-1

Attribute Label	Description	Value Constraints
Maximum Encryption Level	<p>The maximum SSL encryption level used between the Tuxedo domain and WebLogic Server.</p> <ul style="list-style-type: none">■ Zero (0) indicates that the data is signed but not sealed.■ 40, 56, and 128 specify the length, in bits, of the encryption key.■ The default is the maximum level allowed by the Encryption Package kit license.■ If this minimum level of encryption is not met, the SSL connection between Tuxedo and WebLogic Server fails. <p><i>MBean:</i> <code>weblogic.management.configuration.WLECCConnectionPoolMBean</code></p> <p><i>Attribute:</i> <code>MaximumEncryptionLevel</code></p>	

Table 577-1

Attribute Label	Description	Value Constraints
Enable Certificate Authentication	<p>The state of certificate authentication.</p> <ul style="list-style-type: none">■ 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 Tuxedo domain.■ If password authentication is required, WLEC uses the values for the User Name and User Password fields to authenticate. <p><i>MBean:</i> weblogic.management.configuration.WLECCConnectionPoolMBean</p> <p><i>Attribute:</i> CertificateAuthenticationEnabled</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false
Enable Security Context	<p>The state of the security context the WebLogic Server User passed to the Tuxedo domain. If selected, security context is enabled.</p> <p><i>MBean:</i> weblogic.management.configuration.WLECCConnectionPoolMBean</p> <p><i>Attribute:</i> SecurityContextEnabled</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false



WLEC Connection Pool --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this section to supply optional information about your configuration.

Tasks

Enter free form text notes to describe your WLEC pool or configuration.

Related Topics

[Using WebLogic Enterprise Connectivity at {DOCROOT}/wlec/index.html](#)

Attributes

Table 578-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.WLECConnectionPoolMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes



WLEC Connection Pool --> Target and Deploy

[Tasks](#) [Related Topics](#)

Overview

Use this tab to assign connectivity information contained in WLEC connection pools to selected servers and clusters.

Tasks

“Assign WLEC Connection Pools to a Server” on page 582-3

“Assign WLEC Connection Pools to a Cluster” on page 582-3

Related Topics

[Using WebLogic Enterprise Connectivity at {DOCROOT}/wlec/index.html](#)



WLEC Connection Pool

The WebLogic Enterprise Connectivity uses connection pools to enable WebLogic Server clients to connect to BEA Tuxedo domains. A WLEC connection pool is a set of IIOP connections to a BEA Tuxedo domain. WebLogic Server creates the WLEC connection pools at startup and assigns connections to WebLogic Server clients as needed.

- For information on how to configure a WLEC connection pool, see [Configure a New WLEC Connection Pool](#).
- For information on how to monitor active WLEC connection pools, see [Monitor Active Instances of a WLEC Connection Pool](#).
- For information on how to target a server, see [Assign WLEC Connection Pools to a Server](#).
- For information on how to target a cluster, see [Assign WLEC Connection Pools to a Cluster](#).
- For information configure WLEC security, see [Configuring User Security](#).



WLEC Connection Pool --> Monitoring

[Tasks](#) [Related Topics](#)

Overview

From the WLEC Connection Pool—Monitoring tab, you can click the Monitor all Active Pools text link to access a table that lists information about database connections in the connection pool.

Tasks

“Configure a New WLEC Connection Pool” on page 582-2

“Monitor Active Instances of a WLEC Connection Pool” on page 582-2

Related Topics

[Using WebLogic Enterprise Connectivity at {DOCROOT}/wlec/index.html](#)



1 WLEC

[“Attributes and Console Screen Reference for WLEC” on page 583-1]

WebLogic Enterprise Connectivity (WLEC) was deprecated in WebLogic Server 7.0. Tuxedo Corba applications using WLEC should migrate to WebLogic Tuxedo Connector. For more information, see [WebLogic Tuxedo Connector at {DOCR00T}/wtc.html](#).

WebLogic Enterprise Connectivity uses WebLogic Server connection pools to call BEA Tuxedo CORBA objects from WebLogic Server clients (servlets, EJBs, JSPs, and RMI objects).

Configuring WLEC for WebLogic Server

This section provides information on how to configure WebLogic Server to interoperate with Tuxedo using WLEC. For more information see, [Using WebLogic Enterprise Connectivity at {DOCR00T}/wlec/implementing.html](#).

You must update your CLASSPATH environment variable used by your WebLogic Server and command-line shells to include the following files from your WebLogic Server installation:

- `wleorb.jar`
- `wlepool.jar`
- `wleclient.jar`

Configuring WLEC Connection Pools

The following sections provide information on how to configure WLEC Connections Pools for you Tuxedo Corba applications.

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 Create.
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.

Monitor Active 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 Name of the WLEC connection pool you want to monitor.
3. Click the Monitoring tab.

4. Click the Monitor all Active Pools text link.

Assign WLEC Connection Pools to a Server

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 Target and Deploy tab.
3. To assign a connection pool to a selected server, select one or more servers in the list of Independent Servers that has an empty check box. A check mark is displayed next to each server assigned to a WLEC connection pool.
4. To remove a connection pool from a server, select one or more servers in the list of Independent Servers that has a check mark. An empty check box is displayed next to each server that is not assigned to a WLEC connection pool.
5. Click Apply to save your assignments.

Assign WLEC Connection Pools to a Cluster

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 Target and Deploy tab.
3. To assign a connection pool to the selected cluster, select one or more clusters in the list of clusters that has an empty check box. A check mark is displayed next to each server assigned to a WLEC connection pool.
 - a. To assign a connection pool to the all of the servers in a cluster, click All servers in the cluster.
 - b. To assign a connection pool to selected servers in a cluster, click Part of the cluster. Select one or more clusters of the available servers.

4. To remove a connection pool from a cluster, select one or more servers in the list of clusters that has a check mark. An empty check box is displayed next to each cluster that is not assigned to a WLEC connection pool.
5. Click Apply to save your assignments.

Configuring User Security

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 Security tab.
3. In User Name, assign a user identity.
4. In User Password, click change to assign a user password.
5. In User Role, assign a user role.
6. In Application Password, click change to assign a user password.
7. In Minimum Encryption Level, set the minimum level of encryption.
8. In Maximum Encryption Level, set the maximum level of encryption.
9. To enable Certificate Authentication, click the checkbox.
10. To enable Security Context, click the checkbox.
11. Click Apply to save your assignments.

Attributes and Console Screen Reference for WLEC

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“WLEC Connection Pool --> Configuration --> General” on page 576-1](#)

[“WLEC Connection Pool --> Configuration --> Security” on page 577-1](#)

[“WLEC Connection Pool --> Monitoring” on page 581-1](#)

[“WLEC Connection Pool --> Notes” on page 578-1](#)

[“WLEC Connection Pool --> Target and Deploy” on page 579-1](#)

[“WLEC Connection Pool” on page 580-1](#)



Exported Services --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use the tab to provide information on services exported by a local Tuxedo access point. If not specified, all local access points accept requests to all of the services according to the default JNDI lookup rules. If specified, this tab restricts the set of local services requested from a remote domain.

Tasks

“Creating an Exported Service” on page 619-7

Related Topics

[Configuring WebLogic Tuxedo Connector at {DOCROOT}/wtc_admin/Install.html](#)

Attributes

Table 584-1

Attribute Label	Description	Value Constraints
Resource Name	<p>The name used to identify an exported service. This name must be unique within defined Exports. This allows you to define unique configurations having the same Remote Name.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCEXportMBean</p> <p><i>Attribute:</i> ResourceName</p>	
Local Access Point	<p>The name of the local access point that exports the service.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCEXportMBean</p> <p><i>Attribute:</i> LocalAccessPoint</p>	
EJB Name	<p>The complete name of the EJB home interface to use when invoking a service. If not specified, the default interface used is <code>tuxedo.services.servicenameHome</code>.</p> <p>For example: If the service being invoked is TOUPPER and EJBName attribute is not specified, the home interface looked up in JNDI would be <code>tuxedo.services.TOUPPERHome</code>.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCEXportMBean</p> <p><i>Attribute:</i> EJBName</p>	

Table 584-1

Attribute Label	Description	Value Constraints
Remote Name	<p>The remote name of the service. If not specified, the ResourceName attribute is used.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCEXportMBean</p> <p><i>Attribute:</i> RemoteName</p>	



Exported Services

Configure Exported Services to provide information on services exported by a local Tuxedo access point.

- For information on how to create an exported service, see [Creating an Exported Service](#).
- For information on how to delete an exported service, see [Deleting an Exported Service](#).



Imported Services --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to configure services imported and available on remote Tuxedo access points.

Tasks

“Creating Imported Services” on page 619-8

Related Topics

[Configuring WebLogic Tuxedo Connector](#)

Attributes

Table 586-1

Attribute Label	Description	Value Constraints
Resource Name	<p>The name used to identify an imported service. This name must be unique within defined Imports. This allows you to define unique configurations having the same Remote Name.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCImportMBean</p> <p><i>Attribute:</i> ResourceName</p>	
Local Access Point	<p>The name of the local access point that imports the service.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCImportMBean</p> <p><i>Attribute:</i> LocalAccessPoint</p>	
Remote Access Point List	<p>Defines a comma-separated failover list that identifies the remote Tuxedo access points through which resources are imported.</p> <p>Example: TDOM1,TDOM2,TDOM3</p> <p><i>MBean:</i> weblogic.management.configuration.WTCImportMBean</p> <p><i>Attribute:</i> RemoteAccessPointList</p>	
Remote Name	<p>The remote name of the service. If not specified, the ResourceName attribute is used.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCImportMBean</p> <p><i>Attribute:</i> RemoteName</p>	





Imported Services

Configure Imported Services to provide information on services imported and available on remote Tuxedo access points.

- For information on how to create imported services, see [Creating Imported Services](#).
- For information on how to delete imported services, see [Deleting an Imported Service](#).



Local Tuxedo Access Points --> Connections

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to specify the conditions a local Tuxedo access point uses to create a connection with a remote Tuxedo domain.

Tasks

“Configuring Connection Attributes for Local Tuxedo Access Points” on page 619-3

Related Topics

[Configuring WebLogic Tuxedo Connector at {DOCROOT}/wtc_admin/BDCONFIG.html](#)

Attributes

Table 588-1

Attribute Label	Description	Value Constraints
Connection Policy	<p>The conditions under which a local Tuxedo access point tries to establish a connection to a remote Tuxedo access point.</p> <ul style="list-style-type: none">■ ON_DEMAND: A connection is attempted only when requested by either a client request to a remote service or an administrative connect command.■ ON_STARTUP: A domain gateway attempts to establish a connection with its remote Tuxedo access points at gateway server initialization time. Remote services (services advertised in JNDI by the domain gateway for this local access point) are advertised only if a connection is successfully established to that remote Tuxedo access point. If there is no active connection to a remote Tuxedo access point, then the remote services are suspended. By default, this connection policy retries failed connections every 60 seconds. Use the <code>MaxRetry</code> and <code>RetryInterval</code> attributes to specify application specific values.■ INCOMING_ONLY: A domain gateway does not attempt an initial connection to remote Tuxedo access points at startup and remote services are initially suspended. The domain gateway is available for incoming connections from remote Tuxedo access points and remote services are advertised when the domain gateway for this local Tuxedo access point receives an incoming connection. Connection retry processing is not allowed. <p><i>MBean</i>: <code>weblogic.management.configuration.WTCLocalTuxDomMBean</code></p> <p><i>Attribute</i>: <code>ConnectionPolicy</code></p>	<p><i>Default</i>: "ON_DEMAND"</p> <p><i>Valid values</i>:</p> <ul style="list-style-type: none">■ "ON_DEMAND"■ "ON_STARTUP"■ "INCOMING_ONLY"

Table 588-1

Attribute Label	Description	Value Constraints
Connection Principal Name	<p>The principal name used to verify the identity of this local Tuxedo access point when it establishes a session connection with a remote Tuxedo access point.</p> <ul style="list-style-type: none">■ This parameter only applies to domains of type TDOMAIN that are running BEA Tuxedo 7.1 or later software.■ If not specified, the connection principal name defaults to the AccessPointID for this local Tuxedo access point. <p>Note: ConnectionPrincipalName is not supported in this release.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCLocalTuxDomMBean</p> <p><i>Attribute:</i> ConnPrincipalName</p>	
Retry Interval	<p>The number of seconds between automatic attempts to establish a session connection to remote Tuxedo access points. Use only when Connection Policy is set to ON_STARTUP.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCLocalTuxDomMBean</p> <p><i>Attribute:</i> RetryInterval</p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 60</p>
Max Retries	<p>The number of times that a domain gateway tries to establish a session connection to remote Tuxedo access points. Use only when Connection Policy is set to ON_STARTUP.</p> <ul style="list-style-type: none">■ Use the minimum value to disable the retry mechanism.■ Use the maximum value to try until a connection is established. <p><i>MBean:</i> weblogic.management.configuration.WTCLocalTuxDomMBean</p> <p><i>Attribute:</i> MaxRetries</p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 9223372036854775807</p> <p><i>Default:</i> 9223372036854775807</p>

Table 588-1

Attribute Label	Description	Value Constraints
Blocking Time Out	The maximum wait time (seconds) allowed for a blocking call. <i>MBean:</i> weblogic.management.configuration.WTCLocalTuxDomMBean <i>Attribute:</i> BlockTime	<i>Minimum:</i> 0 <i>Maximum:</i> 2147483647 <i>Default:</i> 60
Interoperate	Specifies whether the local Tuxedo access point interoperates with remote Tuxedo access points that are based upon Tuxedo release 6.5. If Yes, the local Tuxedo access point interoperates with a Tuxedo 6.5 domain. <i>MBean:</i> weblogic.management.configuration.WTCLocalTuxDomMBean <i>Attribute:</i> Interoperate	<i>Default:</i> "No"
Compression Limit	The compression threshold used when sending data to a remote Tuxedo access point. Application buffers larger than this size are compressed. <i>MBean:</i> weblogic.management.configuration.WTCLocalTuxDomMBean <i>Attribute:</i> CmpLimit	<i>Minimum:</i> 0 <i>Maximum:</i> 2147483647 <i>Default:</i> 2147483647



Local Tuxedo Access Points --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Local Tuxedo Access Points provide configuration information to connect available remote Tuxedo domains to a WTC Service. You must have at least one Local Tuxedo Access Point configured to create a WTC Service.

Tasks

“Creating a Local Tuxedo Access Point” on page 619-3

Related Topics

[Configuring WebLogic Tuxedo Connector at {DOCROOT}/wtc_admin/Install.html](#)

Attributes

Table 589-1

Attribute Label	Description	Value Constraints
Access Point	<p>Defines a name used to identify a local Tuxedo access point in a WTC Service. This name must be unique within defined Local Tuxedo Access Points. This allows you to define unique configurations having the same Access Point ID.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCLocalTuxDomMBean</p> <p><i>Attribute:</i> AccessPoint</p>	
Access Point Id	<p>The connection principal name used to identify this local Tuxedo access point when attempting to establish a session connection with a remote Tuxedo access point.</p> <p>The AccessPointId must match the corresponding DOMAINID in the *DM_REMOTE_DOMAINS section of your Tuxedo DMCONFIG file.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCLocalTuxDomMBean</p> <p><i>Attribute:</i> AccessPointId</p>	

Table 589-1

Attribute Label	Description	Value Constraints
Network Address	<p>The network address and port number of this local Tuxedo access point. Specify the TCP/IP address in one of the following formats:</p> <ul style="list-style-type: none">■ //hostname:port_number■ //#. #. #. #:port_number <p>If the hostname is used, the access point finds an address for hostname using the local name resolution facilities (usually DNS). If dotted decimal format is used, each # should be a number from 0 to 255. This dotted decimal number represents the IP address of the local machine. The port_number is the TCP port number at which the access point listens for incoming requests.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCLocalTuxDomMBean</p> <p><i>Attribute:</i> NWAddr</p>	



Local Tuxedo Access Points --> Security

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Domain gateways can authenticate incoming connections requested by remote Tuxedo access points and outgoing connections requested by local Tuxedo access points. You define when security should be enforced for incoming connections from remote Tuxedo access points. Specify the level of security used by a particular local Tuxedo access point by setting the `SECURITY` attribute.

Data encryption can be used to prevent network-based eavesdroppers from accessing the content of messages or application-generated messages flowing from local Tuxedo access points to remote Tuxedo access points. Configure this security mechanism by setting the `MINENCRYPTBITS` and `MAXENCRYPTBITS` attributes.

Tasks

“Configuring Security Attributes for Local Tuxedo Access Points” on page 619-4

Related Topics

[Configuring WebLogic Tuxedo Connector at {DOCROOT}/wtc_admin/BDCONFIG.html](#)

Attributes

Table 590-1

Attribute Label	Description	Value Constraints
Security	<p>The type of application security enforced.</p> <ul style="list-style-type: none">■ NONE: No security is used.■ APP_PW: Password security is enforced when a connection is established from a remote Tuxedo access point. The application password is defined in WTC Resources.■ DM_PW: Domain password security is enforced when a connection is established from a remote Tuxedo access point. The domain password is defined in WTC Passwords. <p><i>MBean:</i> weblogic.management.configuration.WTCLocalTuxDomMBean</p> <p><i>Attribute:</i> Security</p>	<p><i>Default:</i> "NONE"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "NONE"■ "APP_PW"■ "DM_PW"

Table 590-1

Attribute Label	Description	Value Constraints
Min Encryption Level	<p>The minimum encryption key length (in bits) used when establishing a network connection for a local domain.</p> <ul style="list-style-type: none">■ A value of 0 indicates no encryption is used.■ The value of the MinEncrypBits attribute must be less than or equal to the value of the MaxEncrypBits attribute.■ A MinEncrypBits of 40 can be used only with domains running Tuxedo 7.1 or higher. <p><i>MBean:</i> weblogic.management.configuration.WTCLocalTuxDomMBean <i>Attribute:</i> MinEncryptBits</p>	<p><i>Default:</i> "0"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "0"■ "40"■ "56"■ "128"
Max Encryption Level	<p>The maximum encryption key length (in bits) used when establishing a network connection for a local domain.</p> <ul style="list-style-type: none">■ A value of 0 indicates no encryption is used.■ The value of the MaxEncryptBits attribute must be greater than or equal to the value of the MinEncrypBits attribute.■ A MaxEncryptBits of 40 can be used only with domains running Tuxedo 7.1 or higher. <p><i>MBean:</i> weblogic.management.configuration.WTCLocalTuxDomMBean <i>Attribute:</i> MaxEncryptBits</p>	<p><i>Default:</i> "128"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "0"■ "40"■ "56"■ "128"



Local Tuxedo Access Points

Local Tuxedo Access Points provide configuration information to connect available remote Tuxedo domains to a WTC Service. You must have at least one Local Tuxedo Access Point configured to create a WTC Service.

- For information on how to create a Local Tuxedo Access Points, see [Creating a Local Tuxedo Access Point](#).
- For information on how to configure network connections, see [Configuring Connection Attributes for Local Tuxedo Access Points](#).
- For information on how to configure security attributes, see [Configuring Security Attributes for Local Tuxedo Access Points](#).
- For information on how to delete a Local Tuxedo Access Points, see [Deleting a Local Tuxedo Access Point](#).



Passwords --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to configure passwords for inter-domain authentication. Password configurations are identified in a WTC service using the combination of the Local Access Point and the Remote Access Point.

Tasks

“Creating a Password Configuration” on page 619-9

Related Topics

[Configuring WebLogic Tuxedo Connector at {DOCROOT}/wtc_admin/Install.html](#)

Attributes

Table 592-1

Attribute Label	Description	Value Constraints
Local Access Point	The name of the local Tuxedo access point to which the password applies. <i>MBean:</i> weblogic.management.configuration.WTCTPasswordMBean <i>Attribute:</i> LocalAccessPoint	
Remote Access Point	The name of the remote Tuxedo access point to which the password applies. <i>MBean:</i> weblogic.management.configuration.WTCTPasswordMBean <i>Attribute:</i> RemoteAccessPoint	
Local Password	Returned from the genpasswd utility. This password is used to authenticate connections between the local Tuxedo access point identified by LocalAccessPoint and the remote Tuxedo access point identified by RemoteAccessPoint. <i>MBean:</i> weblogic.management.configuration.WTCTPasswordMBean <i>Attribute:</i> LocalPassword	
Local Password IV	The initialization vector used to encrypt the local password. <i>MBean:</i> weblogic.management.configuration.WTCTPasswordMBean <i>Attribute:</i> LocalPasswordIV	

Table 592-1

Attribute Label	Description	Value Constraints
Remote Password	<p>Returned from the <code>genpasswd</code> utility. This password is used to authenticate connections between the local Tuxedo access point and the remote Tuxedo access point.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.WTCTPasswordMBean</code></p> <p><i>Attribute:</i> <code>RemotePassword</code></p>	
Remote Password IV	<p>The initialization vector used to encrypt the remote password.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.WTCTPasswordMBean</code></p> <p><i>Attribute:</i> <code>RemotePasswordIV</code></p>	



Passwords

Create a Password configuration to provide passwords for inter-domain authentication.

- For information on how to create a Password configuration, see [Creating a Password Configuration](#).
- For information on how to delete a Password configuration, see [Deleting a Password Configuration](#).



Remote Tuxedo Access Points --> Connections

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

This tab specifies the conditions under which a remote Tuxedo access point tries to establish a session connection with a local Tuxedo access point.

Tasks

“Configuring Connection Attributes for Remote Tuxedo Access Points” on page 619-5

Related Topics

[Configuring WebLogic Tuxedo Connector at {DOCR00T}/wtc_admin/BDCONFIG.html](#)

Attributes

Table 594-1

Attribute Label	Description	Value Constraints
Connection Policy	<p>The conditions under which a remote Tuxedo access point establishes a session connection to a local Tuxedo access point.</p> <ul style="list-style-type: none"> ■ ON_DEMAND: A connection is attempted only when requested by either a client request to a remote service or an administrative connect command. ■ ON_STARTUP: A domain gateway attempts to establish a connection with its remote Tuxedo access points at gateway server initialization time. Remote services (services advertised in JNDI by the domain gateway for this local access point) are advertised only if a connection is successfully established to that remote Tuxedo access point. If there is no active connection to a remote Tuxedo access point, then the remote services are suspended. By default, this connection policy retries failed connections every 60 seconds. Use the MaxRetry and RetryInterval attributes to specify application specific values. ■ INCOMING_ONLY: A domain gateway does not attempt an initial connection to remote Tuxedo access points at startup and remote services are initially suspended. The domain gateway is available for incoming connections from remote Tuxedo access points and remote services are advertised when the domain gateway for this local Tuxedo access point receives an incoming connection. Connection retry processing is not allowed. <p><i>MBean:</i> weblogic.management.configuration.WTCRemoteTuxDomMBean</p> <p><i>Attribute:</i> ConnectionPolicy</p>	<p><i>Default:</i> "ON_DEMAND"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ "ON_DEMAND" ■ "ON_STARTUP" ■ "INCOMING_ONLY" ■ "LOCAL"

Table 594-1

Attribute Label	Description	Value Constraints
Connection Principal Name	<p>The principal name used to verify the identity of this remote Tuxedo access point when it establishes a session connection with a local Tuxedo access point.</p> <ul style="list-style-type: none">■ This parameter only applies to remote Tuxedo access points of type TDOMAIN that are running BEA Tuxedo 7.1 or later software.■ If not specified, the connection principal name defaults to the AccessPointID for this remote Tuxedo access point. <p>Note: ConnectionPrincipalName is not supported in this release.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCRemoteTuxDomMBean</p> <p><i>Attribute:</i> ConnPrincipalName</p>	
Retry Interval	<p>The number of seconds between automatic attempts to establish a connection to remote Tuxedo access points.</p> <ul style="list-style-type: none">■ Use only when Connection Policy is set to ON_STARTUP.■ Use -1 to default to the Retry Interval attribute specified in the Local Tuxedo Access Point configuration named in Local Access Point. <p><i>MBean:</i> weblogic.management.configuration.WTCRemoteTuxDomMBean</p> <p><i>Attribute:</i> RetryInterval</p>	<p><i>Minimum:</i> -1</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> -1</p>

Table 594-1

Attribute Label	Description	Value Constraints
Max Retries	<p>The number of times that a remote Tuxedo access point tries to establish a session connection to local Tuxedo access point. Use only when Connection Policy is set to ON_STARTUP.</p> <ul style="list-style-type: none">■ Use -1 to default to the MaxRetries attribute specified in the Local Tuxedo Access Point configuration named in Local Access Point.■ Use 0 to disable the retry mechanism.■ Use the maximum value to try until a connection is established. <p><i>MBean:</i> weblogic.management.configuration.WTCRemoteTuxDomMBean <i>Attribute:</i> MaxRetries</p>	<p><i>Minimum:</i> -1 <i>Maximum:</i> 9223372036854775807 <i>Default:</i> -1</p>
Cmp Limit	<p>The compression threshold used when sending data to a local Tuxedo access point. Application buffers larger than this size are compressed.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCRemoteTuxDomMBean <i>Attribute:</i> CmpLimit</p>	<p><i>Minimum:</i> 0 <i>Maximum:</i> 2147483647 <i>Default:</i> 2147483647</p>



Remote Tuxedo Access Points --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Remote Tuxedo Access Points provide configuration information to connect a WTC Service to available remote Tuxedo domains.

Tasks

“Creating a Remote Tuxedo Access Point” on page 619-5

Related Topics

[Configuring WebLogic Tuxedo Connector at {DOCROOT}/wtc_admin/Install.html](#)

Attributes

Table 595-1

Attribute Label	Description	Value Constraints
Access Point	<p>Defines a name used to identify a remote Tuxedo access point in a WTC Service. This name must be unique within defined Remote Tuxedo Access Points. This allows you to define unique configurations having the same Access Point ID.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCRemoteTuxDomMBean</p> <p><i>Attribute:</i> AccessPoint</p>	
Access Point ID	<p>The connection principal name used to identify this remote Tuxedo access point when attempting to establish a session connection to local Tuxedo access point.</p> <p>The Access Point ID must match the corresponding DOMAINID in the *DM_LOCAL_DOMAINS section of your Tuxedo DMCONFIG file.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCRemoteTuxDomMBean</p> <p><i>Attribute:</i> AccessPointId</p>	
Local Access Point	<p>The local Tuxedo access point name from which a remote domain is reached.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCRemoteTuxDomMBean</p> <p><i>Attribute:</i> LocalAccessPoint</p>	

Table 595-1

Attribute Label	Description	Value Constraints
Network Address	<p>The network address and port number of this remote Tuxedo access point. Specify the TCP/IP address in one of the following formats:</p> <ul style="list-style-type: none">■ //hostname:port_number■ //#. #. #. #:port_number <p>If the hostname is used, the access point finds an address for hostname using the local name resolution facilities (usually DNS). If dotted decimal format is used, each # should be a number from 0 to 255. This dotted decimal number represents the IP address of the local machine. The port_number is the TCP port number at which the access point listens for incoming requests.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCRemoteTuxDomMBean</p> <p><i>Attribute:</i> NWAddr</p>	
Federation URL	<p>The URL for a foreign name service that is federated into JNDI. If omitted:</p> <ul style="list-style-type: none">■ WebLogic Tuxedo Connector assumes there is a CosNaming server in the foreign domain.■ WebLogic Tuxedo Connector federates to the CosNaming server using TGIOP. <p>WebLogic Tuxedo Connector can to federate to non-CORBA service providers.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCRemoteTuxDomMBean</p> <p><i>Attribute:</i> FederationURL</p>	

Table 595-1

Attribute Label	Description	Value Constraints
Federation Name	The context at which to federate to a foreign name service. If omitted then the federation point is <code>tuxedo.domains</code> . <i>MBean:</i> <code>weblogic.management.configuration.WTCRemoteTuxDomMBean</code> <i>Attribute:</i> <code>FederationName</code>	

Remote Tuxedo Access Points --> Security

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Access Control Lists (ACLs) limit the access to local services within a local Tuxedo access point by restricting the remote Tuxedo access points that can execute these services. Inbound policy from a remote Tuxedo access point is specified using the `AclPolicy` element. Outbound policy towards a remote Tuxedo access point is specified using the `CredentialPolicy` element. This allows WebLogic Server and Tuxedo applications to share the same set of users and the users are able to propagate their credentials from one system to the other.

This release of WebLogic Tuxedo Connector provides the following AppKey Generator plug-ins to provide user security information to Tuxedo:

- **TpUsrFile**—Provides traditional Tuxedo TpUserFile functionality for users who do not need single point security administration or custom security authentication.
- **LDAP**— Provides single point security administration that allows you to maintain user security information in a WebLogic Server embedded LDAP server and use the WebLogic Server Console to administer the security information from a single system. Requires Tuxedo 8.1 and higher.
- **Custom.**—Provides the ability for you to create customized security authentication.

Tasks

“Configuring Security Attributes for Remote Tuxedo Access Points” on page 619-6

Related Topics

[Configuring WebLogic Tuxedo Connector at {DOCROOT}/wtc_admin/BDCONFIG.html](#)

Attributes

Table 596-1

Attribute Label	Description	Value Constraints
Acl Policy	<p>The inbound access control list (ACL) policy toward requests from a remote access point.</p> <ul style="list-style-type: none">■ If Interoperate is set to Yes, AclPolicy is ignored.■ LOCAL: The local access point modifies the identity of service requests received from a given remote access point to the principal name specified in the local principal name for a given remote access point.■ GLOBAL: The local access point passes the service request with no change in identity. <p><i>MBean:</i> weblogic.management.configuration.WTCRemoteTuxDomMBean</p> <p><i>Attribute:</i> AclPolicy</p>	<p><i>Default:</i> "LOCAL"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "GLOBAL"■ "LOCAL"

Table 596-1

Attribute Label	Description	Value Constraints
Credential Policy	<p>The outbound access control list (ACL) policy toward requests to a remote access point.</p> <ul style="list-style-type: none">■ If Interoperate is set to Yes, CredentialPolicy is ignored.■ LOCAL: The remote access point controls the identity of service requests received from the local access point to the principal name specified in the local principal name for this remote access point.■ GLOBAL: The remote access point passes the service request with no change. <p><i>MBean:</i> weblogic.management.configuration.WTCRemoteTuxDomMBean <i>Attribute:</i> CredentialPolicy</p>	<p><i>Default:</i> "LOCAL"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "GLOBAL"■ "LOCAL"
Min Encryption Level	<p>The minimum encryption key length (in bits) used when establishing a network connection for a local access point.</p> <ul style="list-style-type: none">■ A value of 0 indicates no encryption is used.■ The value of the MinEncrypBits attribute must be less than or equal to the value of the MaxEncrypBits attribute.■ A MinEncrypBits of 40 can be used only with access points running Tuxedo 7.1 or higher. <p><i>MBean:</i> weblogic.management.configuration.WTCRemoteTuxDomMBean <i>Attribute:</i> MinEncryptBits</p>	<p><i>Default:</i> "0"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "0"■ "40"■ "56"■ "128"

Table 596-1

Attribute Label	Description	Value Constraints
Max Encryption Level	<p>The maximum encryption key length (in bits) used when establishing a network connection for a local access point.</p> <ul style="list-style-type: none">■ A value of 0 indicates no encryption is used.■ The value of the MaxEncryptBits attribute must be greater than or equal to the value of the MinEncrypBits attribute.■ A MaxEncryptBits of 40 can be used only with access points running Tuxedo 7.1 or higher. <p><i>MBean:</i> weblogic.management.configuration.WTCRemoteTuxDomMBean <i>Attribute:</i> MaxEncryptBits</p>	<p><i>Default:</i> "128"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "0"■ "40"■ "56"■ "128"
Allow Anonymous	<p>Specifies whether the anonymous user is allowed to access Tuxedo. If the anonymous user is allowed to access Tuxedo, the default AppKey will be used for TpUsrFile and LDAP AppKey plug-ins. Interaction with the Custom AppKey plug-in depends on the design of the Custom AppKey generator.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCRemoteTuxDomMBean <i>Attribute:</i> AllowAnonymous</p>	<p><i>Default:</i> false</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ true■ false

Table 596-1

Attribute Label	Description	Value Constraints
Default AppKey	<p>The default AppKey value to be used by the anonymous user and other users who are not defined in the user database if the AppKey plug-in allows them to access Tuxedo. The TpUsrFile and LDAP plug-ins do not allow users that are not defined in user database to access Tuxedo unless Allow Anonymous is enabled.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCRemoteTuxDomMBean</p> <p><i>Attribute:</i> DefaultAppKey</p>	<i>Default:</i> "-1"

Table 596-1

Attribute Label	Description	Value Constraints
AppKey Generator	<p>Specifies the type of AppKey plug-in used. You can choose from the following:</p> <ul style="list-style-type: none"> ■ TpUsrFile. ■ LDAP. ■ Custom. <p>The <code>TpUsrFile</code> is the default plug-in. It uses an imported Tuxedo TPUSR file to provide user security information. Previous releases of WebLogic Tuxedo Connector support this option.</p> <p>The <code>LDAP</code> plug-in utilizes an embedded LDAP server to provide user security information. The user record must define the Tuxedo UID and GID information in the <code>description</code> field. This functionality is not supported in previous releases of WebLogic Tuxedo Connector.</p> <p>A <code>Custom</code> plug-in is provided by users who write their own AppKey generator class to provide the security information required by Tuxedo. This functionality is not supported in previous releases of WebLogic Tuxedo Connector.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.WTCRemoteTuxDomMBean</code></p> <p><i>Attribute:</i> <code>AppKey</code></p>	<p><i>Default:</i> "TpUsrFile"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none"> ■ "TpUsrFile" ■ "LDAP" ■ "Custom"

Table 596-1

Attribute Label	Description	Value Constraints
Tp User File	<p>The full path to the user password file containing UID/GID information. This file is generated by the Tuxedo <code>tpusradd</code> utility on the remote Tuxedo domain specified by the remote Tuxedo access point. A copy of this file must be available in your WebLogic Tuxedo Connector environment to provide correct authorization, authentication, and auditing.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.WTCRemoteTuxDomMBean</code></p> <p><i>Attribute:</i> <code>TpUsrFile</code></p>	
Tuxedo UID Keyword	<p>The keyword for Tuxedo UID (user id) when using the Tuxedo migration utility <code>tpmigldap</code>. This keyword is used to find the Tuxedo UID in the user record of the embedded LDAP database.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.WTCRemoteTuxDomMBean</code></p> <p><i>Attribute:</i> <code>TuxedoUidKw</code></p>	<i>Default:</i> "TUXEDO_UID"
Tuxedo GID Keyword	<p>The keyword for Tuxedo GID (group id) used when using the Tuxedo migration utility <code>tpmigldap</code>. The keyword is used to find Tuxedo GID in the user record of the embedded LDAP database.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.WTCRemoteTuxDomMBean</code></p> <p><i>Attribute:</i> <code>TuxedoGidKw</code></p>	<i>Default:</i> "TUXEDO_GID"

Table 596-1

Attribute Label	Description	Value Constraints
Custom AppKey Class	<p>The full pathname to the custom AppKey generator class. The class at this location is loaded at runtime if the Custom AppKey plug-in is selected.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCRemoteTuxDomMBean</p> <p><i>Attribute:</i> CustomAppKeyClass</p>	
Custom AppKey Param	<p>The optional parameters to be used by the custom AppKey class at the class initialization time.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCRemoteTuxDomMBean</p> <p><i>Attribute:</i> CustomAppKeyClassParam</p>	

Remote Tuxedo Access Points

Remote Tuxedo Access Points provide configuration information to connect a WTC Service to available remote Tuxedo domains.

- For information on how to configure a Remote Tuxedo Access Point, see [Creating a Remote Tuxedo Access Point](#).
- For information on how to configure network connections, see [Configuring Connection Attributes for Remote Tuxedo Access Points](#).
- For information on how to configure security attributes, see [Configuring Security Attributes for Remote Tuxedo Access Points](#).
- For information on how to delete a Remote Tuxedo Access Point, see [Deleting a Remote Tuxedo Access Point](#).



Resources --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use to specify field table classes, reference view buffer structures, provide application passwords, and specify the path for the TPUSER file.

Tasks

“Creating a Resource” on page 619-10

Related Topics

[Configuring WebLogic Tuxedo Connector at {DOCROOT}/wtc_admin/Install.html](#)

Attributes

Table 598-1

Attribute Label	Description	Value Constraints
FldTbl classes	<p>The names of FldTbl16Classes which are loaded via a class loader and added to a FldTbl array.</p> <ul style="list-style-type: none">■ Used fully qualified names of the desired classes.■ Use a comma-separated list to enter multiple classes. <p><i>MBean:</i> weblogic.management.configuration.WTCResourcesMBean</p> <p><i>Attribute:</i> FldTbl16Classes</p>	
FldTbl32 classes	<p>The names of FldTbl32Classes which are loaded via a class loader and added to a FldTbl array.</p> <ul style="list-style-type: none">■ Used fully qualified names of the desired classes.■ Use a comma-separated list to enter multiple classes. <p><i>MBean:</i> weblogic.management.configuration.WTCResourcesMBean</p> <p><i>Attribute:</i> FldTbl32Classes</p>	

Table 598-1

Attribute Label	Description	Value Constraints
ViewTbl classes	<p>The names of ViewTbl16Classes which are loaded via a class loader and added to a ViewTbl array.</p> <ul style="list-style-type: none">■ Used fully qualified names of the desired classes.■ Use a comma-separated list to enter multiple classes. <p><i>MBean:</i> weblogic.management.configuration.WTCResourcesMBean</p> <p><i>Attribute:</i> ViewTbl16Classes</p>	
ViewTbl32 classes	<p>The names of ViewTbl32Classes which are loaded via a class loader and added to a ViewTbl array.</p> <ul style="list-style-type: none">■ Used fully qualified names of the desired classes.■ Use a comma-separated list to enter multiple classes. <p><i>MBean:</i> weblogic.management.configuration.WTCResourcesMBean</p> <p><i>Attribute:</i> ViewTbl32Classes</p>	
App Password	<p>The application password as returned from the genpasswd utility. This Tuxedo application password is the encrypted password used to authenticate connections.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCResourcesMBean</p> <p><i>Attribute:</i> AppPassword</p>	

Table 598-1

Attribute Label	Description	Value Constraints
App Password IV	<p>The initialization vector used to encrypt the AppPassword. It is returned from the <code>genpasswd</code> utility with the AppPassword.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.WTCResourcesMBean</code></p> <p><i>Attribute:</i> <code>AppPasswordIV</code></p>	
TpUsr File Path	<p>The full path to TPUSR file containing Tuxedo User ID (UID) and Group ID (GID) information. This file is generated by the Tuxedo <code>tpusradd</code> utility on the remote Tuxedo domain.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.WTCResourcesMBean</code></p> <p><i>Attribute:</i> <code>TpUsrFile</code></p>	

Resources

Use Resources to specify field table classes, reference view buffer structures, provide application passwords, and specify the path for the TPUSER file.

- For information on how to create resources, see [Creating a Resource](#).
- For information on how to remove resources, see [Removing a Resource](#).



WTC Service --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

You must configure a WTC Service to describe the Tuxedo /T DOMAINS connections used to link WebLogic Server to Tuxedo. You may define more than one WTC Service in your configuration.

Tasks

“Creating a WTC Service” on page 619-1

Related Topics

(Requires an Internet connection.)

[Configuring WebLogic Tuxedo Connector at {DOCROOT}/wtc_admin/Install.html](#)

Attributes

Table 600-1

Attribute Label	Description	Value Constraints
Name	<p>The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.WTCTServerMBean</code></p> <p><i>Attribute:</i> <code>Name</code></p>	
Deployment Order	<p>A priority that the server uses to determine when it deploys an item. The priority is relative to other deployable items of the same type. For example, the server prioritizes and deploys all EJBs before it prioritizes and deploys startup classes.</p> <p>Items with the lowest Deployment Order value are deployed first. There is no guarantee on the order of deployments with equal Deployment Order values. There is no guarantee of ordering across clusters.</p> <p><i>MBean:</i> <code>weblogic.management.configuration.WTCTServerMBean</code></p> <p><i>Attribute:</i> <code>DeploymentOrder</code></p>	<p><i>Minimum:</i> <code>MIN_ORDER</code></p> <p><i>Maximum:</i> <code>MAX_ORDER</code></p> <p><i>Default:</i> <code>DEFAULT_ORDER</code></p> <p><i>Dynamic:</i> <code>yes</code></p>

WTC Service --> Contents --> Exported

Configure Exported Services to provide information on services exported by a local Tuxedo access point.

- For information on how to create an Exported Service, see [Creating an Exported Service](#).
- For information on how to delete an Exported Service, see [Deleting an Exported Service](#).



WTC Service --> Contents --> Imported

Configure Imported Services to provide information on services imported and available on remote Tuxedo domains.

- For information on how to create Imported Services, see [Creating Imported Services](#).
- For information on how to delete Imported Services, see [Deleting an Imported Service](#).



WTC Service --> Contents --> Local Access Points

Local Access Points (Local APs) provide configuration information to connect available remote Tuxedo domains to a WTC Service. You must have at least one Local Tuxedo Access Point configured to create a WTC Service.

- For information on how to create a Local Tuxedo Access Point, see [Creating a Local Tuxedo Access Point](#).
- For information on how to configure network connections, see [Configuring Connection Attributes for Local Tuxedo Access Points](#).
- For information on how to configure security attributes, see [Configuring Security Attributes for Local Tuxedo Access Points](#).
- For information on how to delete a Local Tuxedo Access Point, see [Deleting a Local Tuxedo Access Point](#).



WTC Service --> Contents --> Passwords

Create a Password configuration to provide passwords for inter-domain authentication.

- For information on how to create a Password configuration, see [Creating a Password Configuration](#).
- For information on how to delete a Password configuration, see [Deleting a Password Configuration](#).



WTC Service --> Contents --> Remote Access Points

Remote Access Points (Remote APs) provide configuration information to connect a WTC Service to available remote Tuxedo domains.

- For information on how to configure a Remote Tuxedo Access Point, see [Creating a Remote Tuxedo Access Point](#).
- For information on how to configure network connections, see [Configuring Connection Attributes for Remote Tuxedo Access Points](#).
- For information on how to configure security attributes, see [Configuring Security Attributes for Remote Tuxedo Access Points](#).
- For information on how to delete a Remote Tuxedo Access Point, see [Deleting a Remote Tuxedo Access Point](#).



WTC Service --> Contents --> Queuing Bridge

Create a Tuxedo Queuing Bridge to provide bi-directional JMS interface for your WebLogic Server applications to communicate with Tuxedo application environments.

- For information on how to create a Tuxedo Queuing Bridge connection, see “Creating a Tuxedo Queuing Bridge Connection” on page 619-11.
- For information on how to delete a Tuxedo Queuing Bridge connection, see “Remove a Tuxedo Queuing Bridge Connection” on page 619-11.
- For information on how to assign connection attributes, see “Configuring Connection Attributes for a Tuxedo Queuing Bridge Connection” on page 619-12.
- For information on how to assign connection factories, see “Configuring Connection Factories for a Tuxedo Queuing Bridge Connection” on page 619-12.
- For information on how to create a priority map, see “Configuring Priority Mapping for a Tuxedo Queuing Bridge Connection” on page 619-12.



WTC Service --> Contents --> Redirections

Create a Redirections configuration to provide one-to-one connections between the JMS interface and Tuxedo application environments.

You must create a Queuing Bridge configuration before you can create a Redirections configuration.

- For information on how to create a Tuxedo Queuing Bridge Redirect, see “Creating a Tuxedo Queuing Bridge Redirection” on page 619-13.
- For information on how to delete a Tuxedo Queuing Bridge Redirect, see “Deleting a Tuxedo Queuing Bridge Redirection” on page 619-13.



WTC Service --> Contents --> Resources

Use Resources to specify field table classes, reference view buffer structures, provide application passwords, and specify the path for the TPUSER file.

- For information on how to create resources, see [Creating a Resource](#).
- For information on how to remove resources, see [Removing a Resource](#).



WTC Service --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this section to supply optional information about your configuration.

Tasks

Enter free form text notes to describe your WTC Service configuration.

Related Topics

[Configuring WebLogic Tuxedo Connector at {DOCROOT}/wtc_admin/Install.html](#)

Attributes

Table 609-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.WTCTServerMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes



WTC Service --> Target and Deploy

[Tasks](#) [Related Topics](#)

Overview

Targeting deploys the connectivity information contained in a WTC Service configuration to a selected server. Although you may have many WTC Service configurations in your WebLogic Tuxedo Connector node, only one WTC Service configuration can be targeted to any given server.

Deciding when to target a WTC Service is very important. Once a WTC Service is targeted, the configuration deployed to the selected server is static. Although you can continue to make changes in your WTC Service configuration, any changes made after the configuration is deployed will not be reflected in the selected server. You will need to remove the service from the server and target the service to the server for these changes to take effect.

Tasks

“Assign a WTC Service to a Server” on page 619-2

Related Topics

Configuring WebLogic Tuxedo Connector at [{DOCROOT}/wtc_admin/Install.html](#)



WTC Service

Configure a WTC Service to describe the Tuxedo /T DOMAINS connections used to link WebLogic Server to Tuxedo.

- For information on how to create a WTC Service, see [Creating a WTC Service](#).
- For information on how to delete a WTC Service, see [Deleting a WTC Service](#).
- For information on how to Target a WTC Service, see [Assign a WTC Service to a Server](#).
- For information on how to configure Local Tuxedo Access Points, see [Local Tuxedo Access Point Tasks](#).
- For information on how to configure Remote Tuxedo Access Points, see [Remote Tuxedo Access Point Tasks](#).
- For information on how to configure Exported Services, see [Exported Services Tasks](#).
- For information on how to configure Imported Services, see [Imported Services Tasks](#).
- For information on how to configure Passwords, see [Password Tasks](#).
- For information on how to configure Resources, see [Resource Tasks](#).
- For information on how to configure the Tuxedo Queuing bridge, see [Tuxedo Queuing Bridge Tasks](#).



TUXEDO Queuing Bridge --> Connections

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to configure connectivity attributes for your Tuxedo Queuing Bridge.

Tasks

“Creating a Tuxedo Queuing Bridge Connection” on page 619-11

Related Topics

[Configuring Tuxedo Queuing Bridge at {DOCROOT}/wtc_admin/tBridge.html](#)

Attributes

Table 612-1

Attribute Label	Description	Value Constraints
Timeout	<p>The effective length of a timeout for an entire redirection (seconds) when placing a message on the target location. 0 indicates an infinite wait.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCtBridgeGlobalMBean</p> <p><i>Attribute:</i> Timeout</p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 60</p>
Retries	<p>The number of attempts to redirect a message before putting the message in the specified error location and logging an error.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCtBridgeGlobalMBean</p> <p><i>Attribute:</i> Retries</p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 0</p>
Retry Delay	<p>The minimum amount of time (milliseconds) to wait before redirecting a message after a failure. During this time, no other messages are redirected from the thread. Other threads may continue to redirect messages.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCtBridgeGlobalMBean</p> <p><i>Attribute:</i> RetryDelay</p>	<p><i>Minimum:</i> 0</p> <p><i>Maximum:</i> 2147483647</p> <p><i>Default:</i> 10</p>

TUXEDO Queuing Bridge --> Factories

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to specify connection factories for your JMS, WebLogic Server, and Tuxedo environments.

Tasks

“Configuring Connection Factories for a Tuxedo Queuing Bridge Connection” on page 619-12

Related Topics

[Configuring Tuxedo Queuing Bridge at {DOCROOT}/wtc_admin/tBridge.html](#)

Attributes

Table 613-1

Attribute Label	Description	Value Constraints
JNDI Factory	The name of the JNDI lookup factory. Example: weblogic.jndi.WLInitialContextFactory <i>MBean:</i> weblogic.management.configuration.WTCTBridgeGlobalMBean <i>Attribute:</i> JndiFactory	
JMS Factory	The name of the JMS connection factory. Example: weblogic.jms.ConnectionFactory <i>MBean:</i> weblogic.management.configuration.WTCTBridgeGlobalMBean <i>Attribute:</i> JmsFactory	
Tuxedo Factory	The name of the Tuxedo connection factory. Example: tuxedo.services.TuxedoConnection <i>MBean:</i> weblogic.management.configuration.WTCTBridgeGlobalMBean <i>Attribute:</i> TuxFactory	

TUXEDO Queuing Bridge --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use this tab to configure global attributes for your Tuxedo Queuing Bridge.

Tasks

“Creating a Tuxedo Queuing Bridge Connection” on page 619-11

Related Topics

[Configuring Tuxedo Queuing Bridge at {DOCROOT}/wtc_admin/tBridge.html](#)

Attributes

Table 614-1

Attribute Label	Description	Value Constraints
Transactional	<p>Defines a flag that specifies the use of transactions when retrieving messages from a source location and when placing messages on a target location.</p> <ul style="list-style-type: none">■ If YES, transactions are used for both operations.■ If NO, transactions are not used for either operation. <p>Note: Transactional is not supported in this release.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCTBridgeGlobalMBean</p> <p><i>Attribute:</i> Transactional</p>	<p><i>Default:</i> "NO"</p>
WLS Error Destination	<p>The name of the location used to store WebLogic Server JMS messages when a message cannot be redirected.</p> <ul style="list-style-type: none">■ If not specified, all messages not redirected are lost.■ If the message cannot be placed into <code>WlsErrorDestination</code> for any reason, an error is logged and the message is lost. <p><i>MBean:</i> weblogic.management.configuration.WTCTBridgeGlobalMBean</p> <p><i>Attribute:</i> WlsErrorDestination</p>	

Table 614-1

Attribute Label	Description	Value Constraints
Tuxedo Error Queue	<p>The name of the Tuxedo queue used to store a message that cannot be redirected to a Tuxedo/Q source queue. This queue is in the same queue space as the source queue.</p> <ul style="list-style-type: none">■ If not specified, all messages not redirected are lost.■ If the message cannot be placed into the <code>TuxErrorQueue</code>, an error is logged and the message is lost. <p><i>MBean:</i> <code>weblogic.management.configuration.WTCTBridgeGlobalMBean</code></p> <p><i>Attribute:</i> <code>TuxErrorQueue</code></p>	
Delivery Mode Override	<p>The delivery mode to use when placing messages onto the target location.</p> <ul style="list-style-type: none">■ Overrides any delivery mode associated with a message.■ If not specified, the message is placed on the target location with the same delivery mode specified from the source location. <p><i>MBean:</i> <code>weblogic.management.configuration.WTCTBridgeGlobalMBean</code></p> <p><i>Attribute:</i> <code>DeliveryModeOverride</code></p>	<p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "PERSIST"■ "NONPERSIST"

Table 614-1

Attribute Label	Description	Value Constraints
Default Reply Delivery Mode	<p>The reply delivery mode to associate with a message when placing messages onto the target location.</p> <ul style="list-style-type: none">■ Use when messages are being redirected to Tuxedo/Q from JMS and the JMS_BEA_TuxGtway_Tuxedo_ReplyDeliveryMode property is not set for a message.■ If the defaultReplyDeliveryMode and JMS_BEA_TuxGtway_Tuxedo_ReplyDeliveryMode are not set, the default semantics defined for Tuxedo are enforced by the Tuxedo/Q subsystem. <p><i>MBean:</i> weblogic.management.configuration.WTCTBridgeGlobalMBean</p> <p><i>Attribute:</i> DefaultReplyDeliveryMode</p>	<p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "PERSIST"■ "NONPERSIST"■ "DEFAULT"

Table 614-1

Attribute Label	Description	Value Constraints
User Id	<p>Defines a user identity for all messages handled by the Tuxedo Queuing Bridge for ACL checks when security is configured.</p> <ul style="list-style-type: none">■ All messages assume this identity until the security/authentication contexts are passed between the subsystems. Until the security contexts are passed, there is no secure method to identify who generated a message received from the source location.■ The argument user may be specified as either a user name or a user identification number (uid). <p><i>MBean:</i> weblogic.management.configuration.WTCTBridgeGlobalMBean</p> <p><i>Attribute:</i> UserId</p>	
Allow Non Standard Types	<p>Defines a flag used to specify if non-standard data types are allowed to pass through the Tuxedo Queuing Bridge. Standard types are: ASCII text (TextMessage, STRING), or BLOB (BytesMessage, CARRAY).</p> <ul style="list-style-type: none">■ NO: Non-standard types are rejected and placed onto a specified error location.■ YES: Non-standard types are placed on the target location as BLOBs with a tag indicating the original type. <p><i>MBean:</i> weblogic.management.configuration.WTCTBridgeGlobalMBean</p> <p><i>Attribute:</i> AllowNonStandardTypes</p>	<i>Default:</i> "NO"



TUXEDO Queuing Bridge --> Priority Mapping

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

Use `priorityMapping` to map priorities between the JMS and Tuxedo. JMS has ten priorities (0 - 9); Tuxedo/Q has 100 priorities (1 - 100). There are two mapping directions: **JmstoTux** and **TuxtoJms**. Defaults are provided for all values, shown below in pairs of `value:range`.

`JmstoTux-` 0:1 | 1:12 | 2:23 | 3:34 | 4:45 | 5:56 | 6:67 | 7:78 | 8:89 | 9:100

`TuxtoJms-` 1-10:0 | 11-20:1 | 21-30:2 | 31-40:3 | 41-50:4 | 51-60:5 | 61-70:6 | 71-80:7 | 81-90:8 | 91-100:9

Tasks

“Configuring Priority Mapping for a Tuxedo Queuing Bridge Connection” on page 619-12

Related Topics

[Configuring Tuxedo Queuing Bridge at {DOCROOT}/wtc_admin/tBridge.html](#)

Attributes

Table 615-1

Attribute Label	Description	Value Constraints
Jms To Tux Priority Map	<p>The mapping of priorities from JMS to Tuxedo. The are 10 possible JMS priorities(0=>9) which can be paired to 100 possible Tuxedo priorities(1=>100). A mapping consists of a " " separated list of value-to-range pairs (jmsvalue:tuxrange) where pairs are separated by ":" and ranges are separated by "-".</p> <p>Examples</p> <p>0:1 1:12 2:23 3:34 4:45 5:56 6:67 7:78 8:89 9:100</p> <p>OR</p> <p>0:1-10 1:11-20 2:21-30 3:31-40 4:41-50 5:51-60 6:61-70 7:71-80 8:81-90 9:91-100</p> <p><i>MBean:</i> weblogic.management.configuration.WTctBridgeGlobalMBean</p> <p><i>Attribute:</i> JmsToTuxPriorityMap</p>	

Table 615-1

Attribute Label	Description	Value Constraints
Tux To Jms Priority Map	<p>The mapping of priorities to map from Tuxedo to JMS. There are 100 possible Tuxedo priorities(1=>100) which can be paired to 10 possible JMS priorities(0=>9). A mapping consists of a " " separated list of value-to-range pairs (tuxvalue:jmsrange) where pairs are separated by ":" and ranges are separated by "-".</p> <p>Examples:</p> <p>1:0 12:1 23:2 34:3 45:4 56:5 67:6 78:7 89:8 100:9</p> <p>OR</p> <p>20:0-1 40:2-3 60:4-5 80:6-7 100:8-9</p> <p><i>MBean:</i> weblogic.management.configuration.WTCTBridgeGlobalMBean</p> <p><i>Attribute:</i> TuxToJmsPriorityMap</p>	



TUXEDO Queuing Bridge

The Tuxedo Queuing Bridge provides a bi-directional JMS interface for your WebLogic Server applications to communicate with Tuxedo application environments.

- For information on how to create a Tuxedo Queuing Bridge connection, see “Creating a Tuxedo Queuing Bridge Connection” on page 619-11.
- For information on how to delete a Tuxedo Queuing Bridge connection, see “Remove a Tuxedo Queuing Bridge Connection” on page 619-11.
- For information on how to assign connection attributes, see “Configuring Connection Attributes for a Tuxedo Queuing Bridge Connection” on page 619-12.
- For information on how to assign connection factories, see “Configuring Connection Factories for a Tuxedo Queuing Bridge Connection” on page 619-12.
- For information on how to create a priority map, see “Configuring Priority Mapping for a Tuxedo Queuing Bridge Connection” on page 619-12.
- For information on how to create a Tuxedo Queuing Bridge Redirect, see “Creating a Tuxedo Queuing Bridge Redirection” on page 619-13.
- For information on how to delete a Tuxedo Queuing Bridge Redirect, see “Deleting a Tuxedo Queuing Bridge Redirection” on page 619-13.



Redirection --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The Tuxedo Queuing Bridge establishes a one-way data connection between instances of a JMS queue and a Tuxedo /Q or a JMS queue and a Tuxedo service. Each data connection provides a one-to-one connection between the identified points.

Tasks

“Creating a Tuxedo Queuing Bridge Redirection” on page 619-13

Related Topics

[Configuring Tuxedo Queuing Bridge at {DOCROOT}/wtc_admin/tBridge.html](#)

Attributes

Table 617-1

Attribute Label	Description	Value Constraints
Name	The name used to identify this Redirection in a WTC Service.	

Table 617-1

Attribute Label	Description	Value Constraints
Direction	<p>The direction of data flow. Each defined direction is handled by starting a new thread. At least one redirection must be specified in the Tuxedo Queuing Bridge configuration or the Tuxedo Queuing Bridge will fail to start and an error will be logged.</p> <p>Redirection keywords:</p> <ul style="list-style-type: none">■ JmsQ2TuxQ - From JMS to TUXEDO /Q■ TuxQ2JmsQ - From TUXEDO /Q to JMS■ JmsQ2TuxS - From JMS to TUXEDO Service reply to JMS■ JmsQ2JmsQ - From JMS to JMS <p><i>MBean:</i> weblogic.management.configuration.WTCTBridgeRedirectMBean</p> <p><i>Attribute:</i> Direction</p>	<p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "JmsQ2TuxQ"■ "TuxQ2JmsQ"■ "JmsQ2TuxS"■ "JmsQ2JmsQ"

Table 617-1

Attribute Label	Description	Value Constraints
TranslateFML	<p>The type of XML/FML translation.</p> <ul style="list-style-type: none">■ NO: No data translation is performed. TextMessage maps into STRING and vice versa depending on the directionoftransfer. BytesMessage maps into CARRAY and vice versa. All other data types cause the redirection to fail.■ FLAT: The message payload is transformed using the WebLogic Tuxedo Connector translator.■ WLXT: Translation performed by the XML-to-non-XML WebLogic XML Translator (WLXT). <p>Note: WLXT is not supported for this release.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCTBridgeRedirectMBean</p> <p><i>Attribute:</i> TranslateFML</p>	<p><i>Default:</i> "NO"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "NO"■ "FLAT"■ "WLXT"
Meta Data File	<p>The name of the metadataFile URL used to pass the call to the WLXT.</p> <p>Note: Not supported for this release.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCTBridgeRedirectMBean</p> <p><i>Attribute:</i> MetaDataFile</p>	
Reply Q	<p>The name of the JMS queue used specifically for synchronous calls to a TUXEDO service. The response is returned to the JMS ReplyQ.</p> <p><i>MBean:</i> weblogic.management.configuration.WTCTBridgeRedirectMBean</p> <p><i>Attribute:</i> ReplyQ</p>	

Table 617-1

Attribute Label	Description	Value Constraints
Source Access Point	The name of the local or remote access point where the source is located. <i>MBean:</i> weblogic.management.configuration.WTCtBridgeRedirectMBean <i>Attribute:</i> SourceAccessPoint	
Source Qspace	The name of the Qspace for a source location. <i>MBean:</i> weblogic.management.configuration.WTCtBridgeRedirectMBean <i>Attribute:</i> SourceQspace	
Source Name	The name of a source queue or service. Specify a JMS queue name, a TUXEDO queue name, or the name of a TUXEDO service. <i>MBean:</i> weblogic.management.configuration.WTCtBridgeRedirectMBean <i>Attribute:</i> SourceName	
Target Access Point	The name of the local or remote access point where the target is located. <i>MBean:</i> weblogic.management.configuration.WTCtBridgeRedirectMBean <i>Attribute:</i> TargetAccessPoint	
Target Qspace	The name of the Qspace for a target location. <i>MBean:</i> weblogic.management.configuration.WTCtBridgeRedirectMBean <i>Attribute:</i> TargetQspace	

Table 617-1

Attribute Label	Description	Value Constraints
Target Name	Defines a target queue or service. Specify a JMS queue name, a TUXEDO queue name, or the name of a TUXEDO service. <i>MBean:</i> weblogic.management.configuration.WTCTBridgeRedirectMBean <i>Attribute:</i> TargetName	



Redirection

The Tuxedo Queuing Bridge establishes a one-way data connection between instances of a JMS queue and a Tuxedo /Q or a JMS queue and a Tuxedo service. Each connection provides a one-to-one connection between the identified points.

- For information on how to create a Redirect, see “Creating a Tuxedo Queuing Bridge Redirection” on page 619-13.
- For information on how to delete a Redirect, see “Deleting a Tuxedo Queuing Bridge Redirection” on page 619-13.



1 WebLogic Tuxedo Connector (WTC)

[“Attributes and Console Screen Reference for WebLogic Tuxedo Connector” on page 620-1]

WebLogic Tuxedo Connector (WTC) provides interoperability between WebLogic Server applications and Tuxedo services. The connector allows WebLogic Server clients to invoke Tuxedo services and Tuxedo clients to invoke WebLogic Server Enterprise Java Beans (EJBs) in response to a service request.

WebLogic Tuxedo Connector provides:

- Java Application-to-Transaction Monitor Interface (JATMI) similar to the Tuxedo ATMI
- Tuxedo /Q and JMS advanced messaging service
- Interoperability between CORBA Java and CORBA C++ server applications.

WTC Service Tasks

Creating a WTC Service

1. Select WTC in the navigation tree.
2. Click Configure a new WTC Service.

3. Enter the name of the WTC Service in the name field.
4. Enter a value in the Deployment Order field.
5. Click Create. The new WTC Service appears under WTC in the navigation tree. You are now ready to configure this service.

Deleting a WTC Service

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and right-click.
3. Click Delete.
4. Click Yes to delete the WTC Service. The icon under the WTC in the navigation tree is deleted.
5. Click Continue.

Assign a WTC Service to a Server

1. Select a WTC Service instance in the navigation tree.
2. Click the Target and Deploy tab.
3. To assign a WTC Service to a selected server, select the server from the list of Independent Servers. A WTC Service can be assigned to one server. A check mark is displayed next to the server assigned to the WTC Service.
4. To remove a WTC Service from a server, select the server displaying a check mark. An empty check box is displayed next to each server that is not assigned to a WTC Service.
5. Click Apply to save your assignments.

Local Tuxedo Access Point Tasks

Creating a Local Tuxedo Access Point

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.
3. Click the Local Tuxedo Access Points node.
4. Click Configure a new Local Tuxedo Access Point.
5. In Access Point, enter the unique name used to identify this local Tuxedo access point. This name should be unique for all local and remote Tuxedo access points defined within a WTC Service.
6. In Access Point Id, enter the connection name used to identify this local Tuxedo access point when attempting to establish a session connection with remote Tuxedo access points.
7. In Network Address, enter the network address and port number of this local Tuxedo access point. Specify the TCP/IP address in the format:
`//hostname:port_number` or `//#. #. #. #:port_number`.
8. Click Create.

Configuring Connection Attributes for Local Tuxedo Access Points

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.
3. Click the Local Tuxedo Access Points node.
4. Click to select the Local Tuxedo Access Points instance.
5. Click the Connections tab.

6. Modify the connection attributes for your environment.
7. Click Apply.

Configuring Security Attributes for Local Tuxedo Access Points

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.
3. Click the Local Tuxedo Access Points node.
4. Click to select the Local Tuxedo Access Points instance.
5. Click the Security tab.
6. Modify the security attributes for your environment.
7. Click Apply.

Deleting a Local Tuxedo Access Point

1. Select WTC in the navigation tree.
2. Select a WTC Server instance and click to expand the node.
3. Click the Local Tuxedo Access Points node.
4. Click the Delete icon in the row of Local Tuxedo Access Points you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.
5. Click Yes to delete the Local Tuxedo Access Point. The icon under the Local Tuxedo Access Points node is deleted.
6. Click Continue.

Remote Tuxedo Access Point Tasks

Creating a Remote Tuxedo Access Point

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.
3. Click the Remote Tuxedo Access Points node.
4. Click Configure a new Remote Tuxedo Access Point.
5. In Access Point, enter the unique name used to identify this remote Tuxedo access point. This name should be unique for all local and remote Tuxedo access points defined within a WTC Service.
6. In Access Point Id, enter the connection principal name used to identify this remote Tuxedo access point when attempting to establish a session connection to local Tuxedo access points.
7. In Local Access Point, enter the local domain name from which this remote Tuxedo domain is reached.
8. In Network Address, enter the network address and port number of this remote Tuxedo access point. Specify the TCP/IP address in the format:
`//hostname:port_number` or `//#. #. #. #:port_number`.
9. If your remote domain connection uses RMI/IIOP, configure Federation URL and Federation Name attributes for your environment.
10. Click Create.

Configuring Connection Attributes for Remote Tuxedo Access Points

1. Select WTC in the navigation tree.

2. Select the WTC Server instance and click to expand the node.
3. Click the Remote Tuxedo Access Points node.
4. Click to select the Remote Tuxedo Access Points instance.
5. Click the Connections tab.
6. Modify the connection attributes for your environment.
7. Click Apply.

Configuring Security Attributes for Remote Tuxedo Access Points

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.
3. Click the Remote Tuxedo Access Points node.
4. Click to select the Remote Tuxedo Access Points instance.
5. Click the Security tab.
6. Modify the security attributes for your environment.
7. Click Apply.

Deleting a Remote Tuxedo Access Point

1. Select WTC in the navigation tree.
2. Select a WTC Server instance and click to expand the node.
3. Click the Remote Tuxedo Access Points node.
4. Click the Delete icon in the row of Remote Tuxedo Access Point you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.

5. Click Yes to delete the Remote Tuxedo Access Point. The icon under the Remote Tuxedo Access Points node is deleted.
6. Click Continue.

Exported Services Tasks

Creating an Exported Service

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.
3. Click the Exported Services node.
4. Click Configure a new Exported Service.
5. In Resource Name, enter a name for your exported service.
6. In Local Access Point, enter the name of the Local Access Point that exports this service.
7. In EJB Name, enter the complete name of the EJB home interface to use when invoking this service.
8. In Remote Name, enter the remote name of the service.
9. Click Create.

Deleting an Exported Service

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.
3. Click the Exported Services node.

4. Click the Delete icon in the row of exported services you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.
5. Click Yes to delete the exported service. The icon under the Exported Services node is deleted.
6. Click Continue.

Imported Services Tasks

Creating Imported Services

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.
3. Click the Imported Services node.
4. Click Configure a new Imported Service.
5. In Resource Name, enter the name used to identify this imported service.
6. In Local Access Point, enter the name of the local access point that offers this service.
7. In Remote Access Point List, enter a comma-separated failover list that identifies the remote domain access points through which resources are imported.
8. In Remote Name, enter the remote name of the service.
9. Click Create.

Deleting an Imported Service

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.

3. Click the Imported Services node.
4. Click the Delete icon in the row of imported services you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.
5. Click Yes to delete the imported service. The icon under the Imported Services node is deleted.
6. Click Continue.

Password Tasks

Creating a Password Configuration

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.
3. Click the Passwords node.
4. Click Configure a new Password.
5. In Local Access Point, enter the name of the local access point to which this password applies.
6. In Remote Access Point, enter the name of the remote access point to which this password applies.
7. In Local Password, enter the local password for this configuration.
8. In Local Password IV, enter the initialization vector used to encrypt the local password.
9. In Remote Password, enter the remote password for this configuration.
10. In Remote Password IV, enter the initialization vector used to encrypt the remote password.
11. Click Create.

Deleting a Password Configuration

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.
3. Click the Passwords node.
4. Click the Delete icon in the row of Password configuration you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.
5. Click Yes to delete the imported service. The icon under the Passwords node is deleted.
6. Click Continue.

Resource Tasks

Creating a Resource

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.
3. Click the Resources node.
4. Click Configure a new Resource.
5. Modify the resource attributes for your environment.
6. Click Create.

Removing a Resource

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.

3. Right-click the Resources node.
4. Click Yes to delete the resource.
5. Click Continue.

Tuxedo Queuing Bridge Tasks

Creating a Tuxedo Queuing Bridge Connection

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.
3. Click Configure a new TUXEDO Queuing Bridge.
4. Modify Tuxedo Queuing Bridge attributes for your environment.
5. Click Apply.

Remove a Tuxedo Queuing Bridge Connection

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.
3. Right-click the Tuxedo Queuing Bridge node.
4. Click Remove TUXEDO Queuing Bridge Configuration from the list box.
5. Click Yes to delete the resource.
6. Click Continue.

Configuring Connection Attributes for a Tuxedo Queuing Bridge Connection

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.
3. Click the Tuxedo Queuing Bridge node.
4. Click the Connections tab.
5. Modify the connection attributes for your environment.
6. Click Apply.

Configuring Connection Factories for a Tuxedo Queuing Bridge Connection

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.
3. Click the Tuxedo Queuing Bridge node.
4. Click the Factories tab.
5. Modify the factory attributes for your environment.
6. Click Apply.

Configuring Priority Mapping for a Tuxedo Queuing Bridge Connection

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.

3. Click the Tuxedo Queuing Bridge node.
4. Click the Priority Mapping tab.
5. Modify the mapping attributes for your environment.
6. Click Apply.

Creating a Tuxedo Queuing Bridge Redirection

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.
3. Click the Tuxedo Queuing Bridge node.
4. Click the Redirections node.
5. Click Configure a new Redirection.
6. Modify the redirect attributes for your environment.
7. Click Continue.

Deleting a Tuxedo Queuing Bridge Redirection

1. Select WTC in the navigation tree.
2. Select the WTC Server instance and click to expand the node.
3. Click the Tuxedo Queuing Bridge node.
4. Click the Redirections node.
5. Click the Delete icon in the row of the Redirection you want to delete. A dialog displays in the right pane prompting you to confirm your deletion request.
6. Click Yes to delete the resource. The icon under the Redirect node is deleted.
7. Click Continue.

Attributes and Console Screen Reference for WebLogic Tuxedo Connector

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“Exported Services --> General” on page 584-1](#)

[“Exported Services” on page 585-1](#)

[“Imported Services --> General” on page 586-1](#)

[“Imported Services” on page 587-1](#)

[“Local Tuxedo Access Points --> Connections” on page 588-1](#)

[“Local Tuxedo Access Points --> General” on page 589-1](#)

[“Local Tuxedo Access Points --> Security” on page 590-1](#)

[“Local Tuxedo Access Points” on page 591-1](#)

[“Passwords --> Configuration” on page 592-1](#)

[“Passwords” on page 593-1](#)

[“Remote Tuxedo Access Points --> Connections” on page 594-1](#)

[“Remote Tuxedo Access Points --> General” on page 595-1](#)

[“Remote Tuxedo Access Points --> Security” on page 596-1](#)

[“Remote Tuxedo Access Points” on page 597-1](#)

[“Resources --> Configuration” on page 598-1](#)

[“Resources” on page 599-1](#)

[“WTC Service --> Contents --> Exported” on page 601-1](#)

[“WTC Service --> Contents --> Imported” on page 602-1](#)

[“WTC Service --> Contents --> Local Access Points” on page 603-1](#)

“WTC Service --> Contents --> Passwords” on page 604-1

“WTC Service --> Contents --> Queuing Bridge” on page 606-1

“WTC Service --> Contents --> Redirections” on page 607-1

“WTC Service --> Contents --> Remote Access Points” on page 605-1

“WTC Service --> Contents --> Resources” on page 608-1

“WTC Service --> General” on page 600-1

“WTC Service --> Notes” on page 609-3

“WTC Service --> Target and Deploy” on page 610-1

“WTC Service” on page 611-1

“TUXEDO Queuing Bridge --> Connections” on page 612-1

“TUXEDO Queuing Bridge --> Factories” on page 613-1

“TUXEDO Queuing Bridge --> General” on page 614-1

“TUXEDO Queuing Bridge --> Priority Mapping” on page 615-1

“TUXEDO Queuing Bridge” on page 616-1

“Redirection --> General” on page 617-1

“Redirection” on page 618-1

XML Entity Spec Registry Entry --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

When you configure external entity resolution with WebLogic Server, you physically copy the entity files to a directory accessible by WebLogic Administration Server and specify that the Administration Server use the local copy whenever the external entity is referenced in an XML document.

You can specify that either the Public ID or System ID be used when the entity is referenced in an XML document. You can also specify when the entity should be cached and when it should be refreshed.

Tasks

“Configuring External Entity Resolution” on page 640-5

Related Topics

External Entity Configuration Tasks at
[{DOCR00T}/xml/xml_admin.html#admin009](#)

Attributes

Table 621-1

Attribute Label	Description	Value Constraints
Public Id	The public id of the document type represented by this registry entry. <i>MBean:</i> weblogic.management.configuration.XMLElementSpecRegistryEntryMBean <i>Attribute:</i> PublicId	<i>Dynamic:</i> yes
System Id	The system id of the document type represented by this registry entry. <i>MBean:</i> weblogic.management.configuration.XMLElementSpecRegistryEntryMBean <i>Attribute:</i> SystemId	<i>Dynamic:</i> yes
EntityURI	The location of a local copy of an external entity (e.g., a DTD) that is associated with this registry entry. The location is either a pathname relative to one of the XML registry directories of the installation, or is a URI of the entity location in some local repository (e.g. dbms). <i>MBean:</i> weblogic.management.configuration.XMLElementSpecRegistryEntryMBean <i>Attribute:</i> EntityURI	<i>Dynamic:</i> yes

Table 621-1

Attribute Label	Description	Value Constraints
When To Cache	<p>Set whether to cache this item as soon as possible or wait until it is referenced.</p> <p><i>MBean:</i> weblogic.management.configuration.XMLEntitySpecRegistryEntryMBean</p> <p><i>Attribute:</i> WhenToCache</p>	<p><i>Default:</i> "defer-to-registry-setting"</p> <p><i>Valid values:</i></p> <ul style="list-style-type: none">■ "cache-on-reference"■ "cache-at-initialization"■ "cache-never"■ "defer-to-registry-setting" <p><i>Dynamic:</i> yes</p>
Cache Timeout Interval	<p>The default timeout interval in seconds for the cache. A value of -1 causes this value to be delegated from the cache MBean.</p> <p><i>MBean:</i> weblogic.management.configuration.XMLEntitySpecRegistryEntryMBean</p> <p><i>Attribute:</i> CacheTimeoutInterval</p>	<p><i>Minimum:</i> -1</p> <p><i>Default:</i> -1</p> <p><i>Dynamic:</i> yes</p>



XML Entity Spec Registry Entry --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

When you configure external entity resolution with WebLogic Server, you physically copy the entity files to a directory accessible by WebLogic Administration Server and specify that the Administration Server use the local copy whenever the external entity is referenced in an XML document.

You can specify that either the Public ID or System ID be used when the entity is referenced in an XML document. You can also specify when the entity should be cached and when it should be refreshed.

Use the Notes tab to record information about this external entity entry for your XML Registry.

Tasks

“Configuring External Entity Resolution” on page 640-5

Related Topics

[External Entity Configuration Tasks at {DOCR00T}/xml/xml_admin.html#admin009](#)

Attributes

Table 622-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.XMLEntitySpecRegistryEntryMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

XML Entity Spec Registry Entry

When one or more XML entity spec registry entries have been configured, the table shows the list of external entity resolution entries for a particular XML Registry. Each entry specifies where a particular external entity is locally stored and how an XML document identifies the entity, either using a system ID or a public ID. The entry also specifies how often the cached entity should be refreshed.

To create a new external entity resolution entry for this XML Registry, click on *Configure a new XML Entity Spec Registry Entry*.

- For more information, see “Configuring External Entity Resolution” on page 640-5



XML Parser Select Registry Entry --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

By default, WebLogic Server is configured to use the built-in parser and transformer to parse and transform XML documents. In release 8.1, the built-in XML parser is Apache Xerces and the built-in transformer is Apache Xalan. As long as you use the default, you do not have to perform any configuration tasks for your XML applications. If you want to use a parser or transformer other than the built-in, you must use the XML Registry to configure them.

Use the Configuration tab to specify a parser for a particular document type based on either its System ID, its Public ID, or its root element. You can configure the parser used for DOM style parsing, SAX style parsing, and transformations by entering the appropriate factory class.

Tasks

“Configuring a Parser or Transformer Other Than the Built-In” on page 640-2

“Configuring a Parser for a Particular Document Type” on page 640-3

Related Topics

[XML Parser and Transformer Configuration Tasks at {DOCR00T}/xml/xml_admin.html#admin005](#)

Attributes

Table 624-1

Attribute Label	Description	Value Constraints
Public Id	The public id of the document type represented by this registry entry. <i>MBean:</i> weblogic.management.configuration.XMLParserSelectRegistryEntryMBean <i>Attribute:</i> PublicId	<i>Dynamic:</i> yes
System Id	The system id of the document type represented by this registry entry. <i>MBean:</i> weblogic.management.configuration.XMLParserSelectRegistryEntryMBean <i>Attribute:</i> SystemId	<i>Dynamic:</i> yes
Root Element Tag	The tag name of the document root element of the document type represented by this registry entry. <i>MBean:</i> weblogic.management.configuration.XMLParserSelectRegistryEntryMBean <i>Attribute:</i> RootElementTag	<i>Dynamic:</i> yes
Document Builder Factory	The class name of the DocumentBuilderFactory that is associated with the registry entry. <i>MBean:</i> weblogic.management.configuration.XMLParserSelectRegistryEntryMBean <i>Attribute:</i> DocumentBuilderFactory	<i>Dynamic:</i> yes

Table 624-1

Attribute Label	Description	Value Constraints
Parser Class Name	<p>Return class name of any custom XML parser that is associated with the registry entry.</p> <p><i>MBean:</i> weblogic.management.configuration.XMLParserSelectRegistryEntryMBean</p> <p><i>Attribute:</i> ParserClassName</p>	<i>Dynamic:</i> yes
SAXParser Factory	<p>The class name of the SAXParserFactory that is associated with the registry entry.</p> <p><i>MBean:</i> weblogic.management.configuration.XMLParserSelectRegistryEntryMBean</p> <p><i>Attribute:</i> SAXParserFactory</p>	<i>Dynamic:</i> yes



XML Parser Select Registry Entry --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

By default, WebLogic Server is configured to use the built-in parser and transformer to parse and transform XML documents. In release 8.1, the built-in XML parser is Apache Xerces and the built-in transformer is Apache Xalan. As long as you use the default, you do not have to perform any configuration tasks for your XML applications. If you want to use a parser or transformer other than the built-in, you must use the XML Registry to configure them.

Use the Notes tab to record information about this parser entry for your XML Registry.

Tasks

“Configuring a Parser or Transformer Other Than the Built-In” on page 640-2

“Configuring a Parser for a Particular Document Type” on page 640-3

Related Topics

[XML Parser and Transformer Configuration Tasks at {DOCR00T}/xml/xml_admin.html#admin005](#)

Attributes

Table 625-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.XMLParserSelectRegistryEntryMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

XML Parser Select Registry Entry

When one or more parsers have been registered for, the table shows the list of parser configuration entries for a particular XML Registry. Each entry specifies whether a particular document type uses a parser or transformer other than the built-in parser or transformer. A document type can be identified using a system ID, a public ID, or the root element tag.

To create a new parser configuration entry for this XML Registry, click on *Configure a new Parser Select Registry Entry*.

- For more information, see “Configuring a Parser for a Particular Document Type” on page 640-3



XML Registry --> Configuration

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The XML Registry is a facility for configuring and administering the XML resources of an instance of WebLogic Server. XML resources in WebLogic Server include the parser used by an application to parse XML data, the transformer used by an application to transform XML data, external entity resolution, and caching of external entities.

You can configure the following two types of entries for a given XML registry:

- to configure parsers and transformers.
- to configure external entity resolution.

Use the Configuration tab to set the `DocumentBuilderFactory`, `SaxParserFactory`, and `TransformerFactory` classes for a particular XML Registry. You can also override the default value of when you want WebLogic Server to cache external entities for this XML Registry. Click on the first link to configure external entity resolution and the second link to configure parsers for a particular document type.

Tasks

“Configuring a Parser or Transformer Other Than the Built-In” on page 640-2

“Configuring a Parser for a Particular Document Type” on page 640-3

“Configuring External Entity Resolution” on page 640-5

Related Topics

[Administering WebLogic Server XML at {DOCROOT}/xml/xml_admin.html](#)

Attributes

Table 627-1

Attribute Label	Description	Value Constraints
Name	The name of this configuration. WebLogic Server uses an MBean to implement and persist the configuration. <i>MBean:</i> weblogic.management.configuration.XMLRegistryMBean <i>Attribute:</i> Name	
DocumentBuilderFactory	The class name of the default DocumentBuilderFactory <i>MBean:</i> weblogic.management.configuration.XMLRegistryMBean <i>Attribute:</i> DocumentBuilderFactory	<i>Default:</i> "weblogic.apache.xerces.jaxp.DocumentBuilderFactoryImpl" <i>Dynamic:</i> yes
SAXParserFactory	The class name of the default SAXParserFactory <i>MBean:</i> weblogic.management.configuration.XMLRegistryMBean <i>Attribute:</i> SAXParserFactory	<i>Default:</i> "weblogic.apache.xerces.jaxp.SAXParserFactoryImpl" <i>Dynamic:</i> yes

Table 627-1

Attribute Label	Description	Value Constraints
Transformer Factory	The class name of the default TransformerFactory <i>MBean:</i> weblogic.management.configuration.XMLRegistryMBean <i>Attribute:</i> TransformerFactory	<i>Default:</i> "weblogic.apache.xalan.processor.TransformersFactoryImpl" <i>Dynamic:</i> yes
When To Cache	Set whether to cache items as soon as possible or wait until referenced. <i>MBean:</i> weblogic.management.configuration.XMLRegistryMBean <i>Attribute:</i> WhenToCache	<i>Default:</i> "cache-on-reference" <i>Valid values:</i> <ul style="list-style-type: none">■ "cache-on-reference"■ "cache-at-initialization"■ "cache-never" <i>Dynamic:</i> yes



XML Registry --> Notes

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The XML Registry is a facility for configuring and administering the XML resources of an instance of WebLogic Server. XML resources in WebLogic Server include the parser used by an application to parse XML data, the transformer used by an application to transform XML data, external entity resolution, and caching of external entities.

You can configure the following two types of entries for a given XML registry:

- to configure parsers and transformers.
- to configure external entity resolution.

Use the Notes tab to record notes about this XML Registry.

Tasks

“Configuring a Parser or Transformer Other Than the Built-In” on page 640-2

“Configuring a Parser for a Particular Document Type” on page 640-3

“Configuring External Entity Resolution” on page 640-5

Related Topics

[Administering WebLogic Server XML at {DOCROOT}/xml/xml_admin.html](#)

Attributes

Table 628-1

Attribute Label	Description	Value Constraints
Notes	Optional information that you can include to describe this configuration. <i>MBean:</i> weblogic.management.configuration.XMLRegistryMBean <i>Attribute:</i> Notes	<i>Dynamic:</i> yes

XML Registry --> Target and Deploy

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

The XML Registry is a facility for configuring and administering the XML resources of an instance of WebLogic Server. XML resources in WebLogic Server include the parser used by an application to parse XML data, the transformer used by an application to transform XML data, external entity resolution, and caching of external entities.

You can configure the following two types of entries for a given XML registry:

- to configure parsers and transformers.
- to configure external entity resolution.

Use the Target and Deploy tab to specify the WebLogic Server for which this XML Registry should be targetted and deployed.

Tasks

“Configuring a Parser or Transformer Other Than the Built-In” on page 640-2

“Configuring a Parser for a Particular Document Type” on page 640-3

“Configuring External Entity Resolution” on page 640-5

Related Topics

[Administering WebLogic Server XML at {DOCROOT}/xml/xml_admin.html](#)

Attributes

Attribute Label	Description	Value Constraints
XMLRegistry	The XML Registry for the server. <i>MBean:</i> weblogic.management.configuration.ServerMBean <i>Attribute:</i> XMLRegistry	<i>Configurable:</i> yes <i>Readable:</i> yes <i>Writable:</i> yes

XML Registry

If one or more XML Registries have been configured, this page displays a table that lists the XML Registries configured for WebLogic Server. The XML Registry is a facility for configuring and administering the XML resources of an instance of WebLogic Server. XML resources in WebLogic Server include the parser used by an application to parse XML data, the transformer used by an application to transform XML data, external entity resolution, and caching of external entities.

To create a new XML Registry, click on *Configure a new XML Registry*.

- For more information, see “Configuring a Parser or Transformer Other Than the Built-In” on page 640-2



XMLEntity Cache --> Session --> Rejections

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

You can specify that a managed WebLogic Server cache external entities that are referenced with a URL or a pathname relative to the Administration server, either at server-startup or when the entity is first referenced.

Caching the external entity in the managed WebLogic Server saves the remote access time and provides a local backup in the event that the Administration server cannot be accessed while an XML document is being parsed, due to the network or the Administration server being down.

A set of statistics that describes the external entity cache is available for you to use to monitor the effectiveness of the cache. These statistics describe:

- The current state of the cache.
- The cumulative activity for the current session.
- The cumulative activity since the cache was created, typically when WebLogic Server started.

This tab provides cumulative information about the external entities that have been rejected for inclusion in the external entity cache for the current session.

Tasks

“Monitoring the External Entity Cache” on page 640-8

Related Topics

[Monitoring the External Entity Cache at {DOCROOT}/xml/xml_admin.html#admin012](#)

Attributes

Table 631-1

Attribute Label	Description	Value Constraints
Total Number Of Rejections	<p>The cumulative total number of rejections from the entity cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> TotalNumberOfRejections</p>	
Total Size Of Rejections	<p>The cumulative total size of the rejections from the entity cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> TotalSizeOfRejections</p>	
Percent Rejected	<p>The cumulative percent of the potential entries to the entity cache that have been rejected.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> PercentRejected</p>	

Table 631-1

Attribute Label	Description	Value Constraints
Total Number Of Renewals	<p>The cumulative number of entries that have been refreshed in the entity cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> TotalNumberOfRenewals</p>	



XMLEntity Cache --> Current --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

You can specify that a managed WebLogic Server cache external entities that are referenced with a URL or a pathname relative to the Administration server, either at server-startup or when the entity is first referenced.

Caching the external entity in the managed WebLogic Server saves the remote access time and provides a local backup in the event that the Administration server cannot be accessed while an XML document is being parsed, due to the network or the Administration server being down.

A set of statistics that describes the external entity cache is available for you to use to monitor the effectiveness of the cache. These statistics describe:

- The current state of the cache.
- The cumulative activity for the current session.
- The cumulative activity since the cache was created, typically when WebLogic Server started.

This tab provides general information about the state of the external entity cache for the current session.

Tasks

“Monitoring the External Entity Cache” on page 640-8

Related Topics

[Monitoring the External Entity Cache at {DOCROOT}/xml/xml_admin.html#admin012](#)

Attributes

Table 632-1

Attribute Label	Description	Value Constraints
Total Current Entries	<p>The total current number of entries in the entity cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCurrentStateRuntimeMBean</p> <p><i>Attribute:</i> TotalCurrentEntries</p>	
Total Persistent Current Entries	<p>The total current number of entries in the cache that have been persisted to disk.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCurrentStateRuntimeMBean</p> <p><i>Attribute:</i> TotalPersistentCurrentEntries</p>	
Avg Percent Persistent	<p>Returns current average percentage of entries in the entity cache that have been persisted to the disk cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCurrentStateRuntimeMBean</p> <p><i>Attribute:</i> AvgPercentPersistent</p>	

Table 632-1

Attribute Label	Description	Value Constraints
Total Transient Current Entries	<p>The total current number of transient entries in the entity cache.</p> <p><i>MBean:</i> <code>weblogic.management.runtime.EntityCacheCurrentStateRuntimeMBean</code></p> <p><i>Attribute:</i> <code>TotalTransientCurrentEntries</code></p>	
Avg Percent Transient	<p>Returns current average percentage of entries in the entity cache that are transient, or have not been persisted.</p> <p><i>MBean:</i> <code>weblogic.management.runtime.EntityCacheCurrentStateRuntimeMBean</code></p> <p><i>Attribute:</i> <code>AvgPercentTransient</code></p>	
Min Entry Timeout	<p>The smallest timeout value for any current entry in the entity cache.</p> <p><i>MBean:</i> <code>weblogic.management.runtime.EntityCacheCurrentStateRuntimeMBean</code></p> <p><i>Attribute:</i> <code>MinEntryTimeout</code></p>	
Max Entry Timeout	<p>The largest timeout value for any current entry in the entity cache.</p> <p><i>MBean:</i> <code>weblogic.management.runtime.EntityCacheCurrentStateRuntimeMBean</code></p> <p><i>Attribute:</i> <code>MaxEntryTimeout</code></p>	

Table 632-1

Attribute Label	Description	Value Constraints
Avg Timeout	<p>The average amount of time that the entity cache has timed out when trying to retrieve an entity.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCurrentStateRuntimeMBean</p> <p><i>Attribute:</i> AvgTimeout</p>	

XMLEntity Cache --> Current --> Entry Resource Usage

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

You can specify that a managed WebLogic Server cache external entities that are referenced with a URL or a pathname relative to the Administration server, either at server-startup or when the entity is first referenced.

Caching the external entity in the managed WebLogic Server saves the remote access time and provides a local backup in the event that the Administration server cannot be accessed while an XML document is being parsed, due to the network or the Administration server being down.

A set of statistics that describes the external entity cache is available for you to use to monitor the effectiveness of the cache. These statistics describe:

- The current state of the cache.
- The cumulative activity for the current session.
- The cumulative activity since the cache was created, typically when WebLogic Server started.

This tab provides information about the resource usage of the external entity cache for the current session.

Tasks

“Monitoring the External Entity Cache” on page 640-8

Related Topics

[Monitoring the External Entity Cache at {DOCROOT}/xml/xml_admin.html#admin012](#)

Attributes

Table 633-1

Attribute Label	Description	Value Constraints
Max Entry Memory Size	<p>The current maximum size of the entries in the entity memory cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCurrentStateRuntimeMBean</p> <p><i>Attribute:</i> MaxEntryMemorySize</p>	
Min Entry Memory Size	<p>The current minimum size of the entries in the entity memory cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCurrentStateRuntimeMBean</p> <p><i>Attribute:</i> MinEntryMemorySize</p>	
Avg Per Entry Memory Size	<p>The current average size of the entries in the entity memory cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCurrentStateRuntimeMBean</p> <p><i>Attribute:</i> AvgPerEntryMemorySize</p>	

Table 633-1

Attribute Label	Description	Value Constraints
Avg Per Entry Disk Size	<p>The current average size of the entries in the entity disk cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCurrentStateRuntimeMBean</p> <p><i>Attribute:</i> AvgPerEntryDiskSize</p>	



XMLEntity Cache --> Session --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

You can specify that a managed WebLogic Server cache external entities that are referenced with a URL or a pathname relative to the Administration server, either at server-startup or when the entity is first referenced.

Caching the external entity in the managed WebLogic Server saves the remote access time and provides a local backup in the event that the Administration server cannot be accessed while an XML document is being parsed, due to the network or the Administration server being down.

A set of statistics that describes the external entity cache is available for you to use to monitor the effectiveness of the cache. These statistics describe:

- The current state of the cache.
- The cumulative activity for the current session.
- The cumulative activity since the cache was created, typically when WebLogic Server started.

This tab provides general cumulative information about the external entity cache for the current session.

Tasks

“Monitoring the External Entity Cache” on page 640-8

Related Topics

[Monitoring the External Entity Cache at {DOCROOT}/xml/xml_admin.html#admin012](#)

Attributes

Table 634-1

Attribute Label	Description	Value Constraints
Total Current Entries	<p>The total current number of entries in the entity cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> TotalCurrentEntries</p>	
Total Persistent Current Entries	<p>The total current number of entries in the cache that have been persisted to disk.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> TotalPersistentCurrentEntries</p>	
Avg Percent Persistent	<p>Returns current average percentage of entries in the entity cache that have been persisted to the disk cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> AvgPercentPersistent</p>	

Table 634-1

Attribute Label	Description	Value Constraints
Total Transient Current Entries	<p>The total current number of transient entries in the entity cache.</p> <p><i>MBean:</i> <code>weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</code></p> <p><i>Attribute:</i> <code>TotalTransientCurrentEntries</code></p>	
Avg Percent Transient	<p>Returns current average percentage of entries in the entity cache that are transient, or have not been persisted.</p> <p><i>MBean:</i> <code>weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</code></p> <p><i>Attribute:</i> <code>AvgPercentTransient</code></p>	
Min Entry Timeout	<p>The smallest timeout value for any current entry in the entity cache.</p> <p><i>MBean:</i> <code>weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</code></p> <p><i>Attribute:</i> <code>MinEntryTimeout</code></p>	
Max Entry Timeout	<p>The largest timeout value for any current entry in the entity cache.</p> <p><i>MBean:</i> <code>weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</code></p> <p><i>Attribute:</i> <code>MaxEntryTimeout</code></p>	

Table 634-1

Attribute Label	Description	Value Constraints
Avg Timeout	<p>The average amount of time that the entity cache has timed out when trying to retrieve an entity.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> AvgTimeout</p>	

XMLEntity Cache --> Session --> Entry Resource Usage

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

You can specify that a managed WebLogic Server cache external entities that are referenced with a URL or a pathname relative to the Administration server, either at server-startup or when the entity is first referenced.

Caching the external entity in the managed WebLogic Server saves the remote access time and provides a local backup in the event that the Administration server cannot be accessed while an XML document is being parsed, due to the network or the Administration server being down.

A set of statistics that describes the external entity cache is available for you to use to monitor the effectiveness of the cache. These statistics describe:

- The current state of the cache.
- The cumulative activity for the current session.
- The cumulative activity since the cache was created, typically when WebLogic Server started.

This tab provides cumulative information about the resource usage of the external entity cache for the current session.

Tasks

“Monitoring the External Entity Cache” on page 640-8

Related Topics

[Monitoring the External Entity Cache at {DOCROOT}/xml/xml_admin.html#admin012](#)

Attributes

Table 635-1

Attribute Label	Description	Value Constraints
Max Entry Memory Size	<p>The current maximum size of the entries in the entity memory cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> MaxEntryMemorySize</p>	
Min Entry Memory Size	<p>The current minimum size of the entries in the entity memory cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> MinEntryMemorySize</p>	
Avg Per Entry Memory Size	<p>The current average size of the entries in the entity memory cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> AvgPerEntryMemorySize</p>	

Table 635-1

Attribute Label	Description	Value Constraints
Avg Per Entry Disk Size	The current average size of the entries in the entity disk cache. <i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean <i>Attribute:</i> AvgPerEntryDiskSize	



XMLEntity Cache --> Historical --> Rejections

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

You can specify that a managed WebLogic Server cache external entities that are referenced with a URL or a pathname relative to the Administration server, either at server-startup or when the entity is first referenced.

Caching the external entity in the managed WebLogic Server saves the remote access time and provides a local backup in the event that the Administration server cannot be accessed while an XML document is being parsed, due to the network or the Administration server being down.

A set of statistics that describes the external entity cache is available for you to use to monitor the effectiveness of the cache. These statistics describe:

- The current state of the cache.
- The cumulative activity for the current session.
- The cumulative activity since the cache was created, typically when WebLogic Server started.

This tab provides cumulative information about the external entities that were rejected from the external entity cache for the life of the current WebLogic Server instance.

Tasks

“Monitoring the External Entity Cache” on page 640-8

Related Topics

[Monitoring the External Entity Cache at {DOCROOT}/xml/xml_admin.html#admin012](#)

Attributes

Table 636-1

Attribute Label	Description	Value Constraints
Total Number Of Rejections	<p>The cumulative total number of rejections from the entity cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> TotalNumberOfRejections</p>	
Total Size Of Rejections	<p>The cumulative total size of the rejections from the entity cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> TotalSizeOfRejections</p>	
Percent Rejected	<p>The cumulative percent of the potential entries to the entity cache that have been rejected.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> PercentRejected</p>	

Table 636-1

Attribute Label	Description	Value Constraints
Total Number Of Renewals	<p>The cumulative number of entries that have been refreshed in the entity cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> TotalNumberOfRenewals</p>	



XMLEntity Cache --> Current --> Total Resource Usage

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

You can specify that a managed WebLogic Server cache external entities that are referenced with a URL or a pathname relative to the Administration server, either at server-startup or when the entity is first referenced.

Caching the external entity in the managed WebLogic Server saves the remote access time and provides a local backup in the event that the Administration server cannot be accessed while an XML document is being parsed, due to the network or the Administration server being down.

A set of statistics that describes the external entity cache is available for you to use to monitor the effectiveness of the cache. These statistics describe:

- The current state of the cache.
- The cumulative activity for the current session.
- The cumulative activity since the cache was created, typically when WebLogic Server started.

This tab provides information about the memory and disk usage of the external entity cache for the current session.

Tasks

“Monitoring the External Entity Cache” on page 640-8

Related Topics

[Monitoring the External Entity Cache at {DOCROOT}/xml/xml_admin.html#admin012](#)

Attributes

Table 637-1

Attribute Label	Description	Value Constraints
Memory Usage	Returns current size of the entity memory cache. <i>MBean:</i> weblogic.management.runtime.EntityCacheCurrentStateRuntimeMBean <i>Attribute:</i> MemoryUsage	
Disk Usage	The current size of the entity disk cache. <i>MBean:</i> weblogic.management.runtime.EntityCacheCurrentStateRuntimeMBean <i>Attribute:</i> DiskUsage	

XMLEntity Cache --> Historical --> Entry Resource Usage

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

You can specify that a managed WebLogic Server cache external entities that are referenced with a URL or a pathname relative to the Administration server, either at server-startup or when the entity is first referenced.

Caching the external entity in the managed WebLogic Server saves the remote access time and provides a local backup in the event that the Administration server cannot be accessed while an XML document is being parsed, due to the network or the Administration server being down.

A set of statistics that describes the external entity cache is available for you to use to monitor the effectiveness of the cache. These statistics describe:

- The current state of the cache.
- The cumulative activity for the current session.
- The cumulative activity since the cache was created, typically when WebLogic Server started.

This tab provides resource information about the external entities in the external entity cache for the life of the current WebLogic Server instance.

Tasks

“Monitoring the External Entity Cache” on page 640-8

Related Topics

[Monitoring the External Entity Cache at {DOCROOT}/xml/xml_admin.html#admin012](#)

Attributes

Table 638-1

Attribute Label	Description	Value Constraints
Max Entry Memory Size	<p>The current maximum size of the entries in the entity memory cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> MaxEntryMemorySize</p>	
Min Entry Memory Size	<p>The current minimum size of the entries in the entity memory cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> MinEntryMemorySize</p>	
Avg Per Entry Memory Size	<p>The current average size of the entries in the entity memory cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> AvgPerEntryMemorySize</p>	

Table 638-1

Attribute Label	Description	Value Constraints
Avg Per Entry Disk Size	The current average size of the entries in the entity disk cache. <i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean <i>Attribute:</i> AvgPerEntryDiskSize	



XMLEntity Cache --> Historical --> General

[Tasks](#) [Related Topics](#) [Attributes](#)

Overview

You can specify that a managed WebLogic Server cache external entities that are referenced with a URL or a pathname relative to the Administration server, either at server-startup or when the entity is first referenced.

Caching the external entity in the managed WebLogic Server saves the remote access time and provides a local backup in the event that the Administration server cannot be accessed while an XML document is being parsed, due to the network or the Administration server being down.

A set of statistics that describes the external entity cache is available for you to use to monitor the effectiveness of the cache. These statistics describe:

- The current state of the cache.
- The cumulative activity for the current session.
- The cumulative activity since the cache was created, typically when WebLogic Server started.

This tab provides general cumulative information about the external entity cache for the life of the current WebLogic Server instance.

Tasks

“Monitoring the External Entity Cache” on page 640-8

Related Topics

[Monitoring the External Entity Cache at {DOCROOT}/xml/xml_admin.html#admin012](#)

Attributes

Table 639-1

Attribute Label	Description	Value Constraints
Total Current Entries	<p>The total current number of entries in the entity cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> TotalCurrentEntries</p>	
Total Persistent Current Entries	<p>The total current number of entries in the cache that have been persisted to disk.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> TotalPersistentCurrentEntries</p>	
Avg Percent Persistent	<p>Returns current average percentage of entries in the entity cache that have been persisted to the disk cache.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> AvgPercentPersistent</p>	

Table 639-1

Attribute Label	Description	Value Constraints
Total Transient Current Entries	<p>The total current number of transient entries in the entity cache.</p> <p><i>MBean:</i> <code>weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</code></p> <p><i>Attribute:</i> <code>TotalTransientCurrentEntries</code></p>	
Avg Percent Transient	<p>Returns current average percentage of entries in the entity cache that are transient, or have not been persisted.</p> <p><i>MBean:</i> <code>weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</code></p> <p><i>Attribute:</i> <code>AvgPercentTransient</code></p>	
Min Entry Timeout	<p>The smallest timeout value for any current entry in the entity cache.</p> <p><i>MBean:</i> <code>weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</code></p> <p><i>Attribute:</i> <code>MinEntryTimeout</code></p>	
Max Entry Timeout	<p>The largest timeout value for any current entry in the entity cache.</p> <p><i>MBean:</i> <code>weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</code></p> <p><i>Attribute:</i> <code>MaxEntryTimeout</code></p>	

Table 639-1

Attribute Label	Description	Value Constraints
Avg Timeout	<p>The average amount of time that the entity cache has timed out when trying to retrieve an entity.</p> <p><i>MBean:</i> weblogic.management.runtime.EntityCacheCumulativeRuntimeMBean</p> <p><i>Attribute:</i> AvgTimeout</p>	

1 XML

[“Attributes and Console Screen Reference for XML” on page 641-1]

The XML Registry is a facility for configuring and administering the XML resources of an instance of WebLogic Server. XML resources in WebLogic Server include the parser used by an application to parse XML data, the transformer used by an application to transform XML data, external entity resolution, and caching of external entities.

If an instance of WebLogic Server does not have an XML Registry associated with it, then the built-in parser and transformer are used when parsing or transforming documents. In addition, you cannot configure external entity resolution to increase the performance of your XML applications.

Once you associate an XML Registry with an instance of WebLogic Server, all XML configuration options are available for XML applications that use that server.

You can configure the following two types of entries for a given XML registry:

- to configure parsers and transformers.
- to configure external entity resolution.

Tasks

Configuring a Parser or Transformer Other Than the Built-In

1. In the left pane, right-click the XML node under the Services node and select Configure a new XML Registry from the drop-down menu. The window to create a new XML registry is displayed.
2. Enter a unique registry name in the Name field and set the Document Builder Factory, Sax Parser Factory, and Transformer Factory fields to the appropriate Factory parser and transformer classes.

For example, to use WebLogic FastParser, enter the following information:

```
Name: WebLogic FastParser
DocumentBuilderFactory:
SAXParserFactory: weblogic.xml.babel.jaxp.SAXParserFactoryImpl
TransformerFactory:
```

Note that in the preceding example, DocumentBuilderFactory and TransformerFactory have been left blank. This means that for DOM parsing and transformation, the built-in parser and transformer are used, respectively. The WebLogic FastParser will only be used for SAX parsing.

If you want to directly specify the Apache Xerces parser and Xalan transformer, enter the following information:

```
Name: Apache Xerces/Xalan Registry
DocumentBuilderFactory: org.apache.xerces.jaxp.DocumentBuilderFactoryImpl
SAXParserFactory: org.apache.xerces.jaxp.SAXParserFactoryImpl
TransformerFactory: org.apache.xalan.processor.TransformerFactoryImpl
```

3. Select one of the following options from the When To Cache list box:
 - cache-on-reference—WebLogic Server caches the external entity referenced by a URL the first time the entity is referenced in an XML document.

- `cache-at-initialization`—WebLogic Server caches the entity when the server starts.
 - `cache-never`—WebLogic Server never caches the external entity.
4. Click the Create button. The XML Registry is created and listed under the XML node in the left pane.
 5. Select the Target and Deploy tab.
 6. Select the Server on which you want to deploy this XML Registry.
 7. Click Apply.

Configuring a Parser for a Particular Document Type

When you configure a parser for a particular document type, you can use the document's system id, public id, or root element tag to identify the document type.

Warning: WebLogic Server searches only the first 1000 bytes of an XML document when attempting to identify its document type. If it does not find a DOCTYPE identifier in those first 1000 bytes, it stops searching the document and uses the parser configured for the WebLogic Server instance to parse the document.

Note: The following procedure assumes that you are going to create a new XML registry, add the necessary parser registry entries, and associate it with a server. If you have already associated an existing XML registry with your server, skip to step 7.

1. In the left pane, right-click the XML node under the Services node and select Configure a new XML Registry from the drop-down menu. The window to create a new XML registry is displayed.
2. Enter a unique registry name in the Name field. If you want to configure default parsers and transformer for your server, enter the factory class names in the Document Builder Factory, Sax Parser Factory, and Transformer Factory fields. Otherwise, leave these fields blank.
3. Click the Create button. The XML Registry is created and listed under the XML node in the left pane.

4. Select the Target and Deploy tab.
 5. Select the Server on which you want to deploy this XML Registry.
 6. Click Apply.
 7. Under the XML node in the left pane, right-click the Parser Select Entries node under your XML registry. Select Configure a New XMLParserSelectRegistryEntry from the drop-down menu. A blank window for entering document type information appears in the right pane.
 8. Enter the document type information in one of the following ways:
 - a. Use either the Public Id or the System Id field to specify the doctype. For example, for the `car.xml` (see Listing 1-1), enter `-//BEA Systems, Inc.//DTD for cars//EN` in the Public Id field.
 - b. Specify the Root Element Tag name of the document. For the `car.xml` example, enter `CAR` in the Root Element Tag field.
- If your XML document defines a namespace, be sure to enter the fully qualified root element tag, such as `VEHICLES:CAR`.

Listing 1-1 car.xml File

```
<?xml version="1.0"?>
<!-- This XML document describes a car -->
<!DOCTYPE CAR PUBLIC "-//BEA Systems, Inc.//DTD for cars//EN"
"http://www.bea.com/dtds/car.dtd">
<CAR>
<MAKE>Toyota</MAKE>
<MODEL>Corrolla</MODEL>
<YEAR>1998</YEAR>
<ENGINE>1.5L</ENGINE>
<HP>149</HP>
</CAR>
```

9. Set the Document Builder Factory or Sax Parser Factory fields to the appropriate Factory parser classes.

For example, enter `weblogic.xml.babel.jaxp.SAXParserFactoryImpl` in the Sax Parser Factory field to specify that this document type be parsed by WebLogic FastParser.

Note: Do not enter any information in the Parser Class Name field; this field is for backward compatibility with previous versions of WebLogic Server only.

10. Click the Create button. The XMLParserSelect registry entry is created.

Configuring External Entity Resolution

When you configure external entity resolution with WebLogic Server, you physically copy the entity files to a directory accessible by WebLogic Administration Server and specify that the Administration Server use the local copy whenever the external entity is referenced in an XML document.

Note: In the following procedure it is assumed that you are going to create a new XML registry, add the necessary external entity resolution registry entries, and associate it with a server. If you have already associated an existing XML registry with your server, skip to step 7.

1. Right-click the XML node under the Services node in the left pane and select Configure a new XML Registry from the drop-down menu. The window to create a new XML registry is displayed in the right pane.
2. In the Name field, enter a unique registry name. If you want to configure default parsers and transformer for your server, enter the factory class names in the Document Builder Factory, Sax Parser Factory, and Transformer Factory fields. Otherwise, leave these fields blank.
3. Click the Create button. The XML Registry is created and listed under the XML node in the left pane.
4. Select the Target and Deploy tab.
5. Select the Server on which you want to deploy this XML Registry.
6. Click Apply.
7. Under the XML node in the left pane, right-click the Entity Spec Entries node under your XML registry. Select Configure a New XMLEntitySpecRegistryEntry from the drop-down menu. A blank window for entering entity resolution information appears in the right pane.

8. Enter either the `System Id` or `Public Id` that is used to reference the external entity in the XML document. For example, for the following `car.xml` file, enter `http://www.bea.com/dtds/car.dtd` for the `System Id`:

Listing 1-2 car.xml File

```
<?xml version="1.0"?>
<!-- This XML document describes a car -->
<!DOCTYPE CAR PUBLIC "-//BEA Systems, Inc.//DTD for cars//EN"
"http://www.bea.com/dtds/car.dtd">
<CAR>
<MAKE>Toyota</MAKE>
<MODEL>Corrolla</MODEL>
<YEAR>1998</YEAR>
<ENGINE>1.5L</ENGINE>
<HP>149</HP>
</CAR>
```

9. If you are configuring a local copy of an external entity, ensure that the registries entity directory `DOMAIN/xml/registries/reg_name` exists, where `DOMAIN` refers to your domain directory and `reg_name` is the name of your XML Registry. If it does not exist, create it.

10. In the EntityURI field, enter one of the following two entity paths:

- a. The pathname of the copy of the entity file in the Administration Server. This pathname must be relative to the registries entity directory `DOMAIN/xml/registries/reg_name`.

For example, for the `car.xml` file, you might enter `dtds/car.dtd` in the EntityURI field.

- b. A URL that points to an external entity out on the Web or an entity stored in a repository. For example, enter `http://java.sun.com/j2ee/dtds/application_1_2.dtd` to reference the DTD for the `application.xml` file used to describe J2EE Enterprise Applications or use `jdbc:` to reference an entity in a database.

Use the following protocol declarations to specify an external entity:

`http://`, `file://`, `jdbc:`, or `ftp://`.

11. Select one of the following options from the When To Cache list box:

- `cache-on-reference`—WebLogic Server caches the external entity referenced by a URL the first time the entity is referenced in an XML document.
 - `cache-at-initialization`—WebLogic Server caches the entity when the server starts.
 - `defer-to-registry-setting`—WebLogic Server uses the default caching setting.
 - `cache-never`—WebLogic Server never caches the external entity.
12. In the Cache Timeout Interval field, enter the number of seconds after which the cached external entity becomes stale, or out-of-date. WebLogic Server re-retrieves the external entity from the specified URL or pathname relative to the Administration server if the cached copy has been in the cache for longer than this amount.
- The default value for this field is -1, which means that the global timeout value for WebLogic Server is used.
13. Click the Create button. The XMLEntitySpec registry entry is created.
14. If you specified that a local copy of the entity be used, rather than caching the one from the Web, copy the entity file into the entity directory.

For example, you would copy the `car.dtd` file to the directory `DOMAIN/xml/registries/reg_name/dtds`, where *DOMAIN* refers to your domain directory and *reg_name* is the name of your XML Registry.

Configuring the External Entity Cache

You can configure the following properties of the external entity cache:

- Size, in KB, of the cache memory. The default value for this property is 500 KB.
- Size, in MB, of the persistent disk cache. The default value for this property is 5 MB.
- Number of seconds after which external entities in the cache become stale after they have been cached by WebLogic Server. This is the default value for the entire server - you can override this value for specific external entities when you

configure the entity. The default value for this property is 120 seconds (2 minutes).

To configure the external entity cache, follow these steps:

1. Under the Servers node in the left pane, click the name of the WebLogic Server for which you want to configure the external entity cache.
2. Select the Services tab in the right pane.
3. Select the XML tab. The window to configure XML properties of WebLogic Server appears in the right pane.
4. In the Cache Memory Size field, enter the size, in KB, of the cache memory. The default value for this property is 500 KB.
5. In the Cache Disk Size field, enter the size, in MB, of the persistent disk cache. The default value for this property is 5 MB.
6. In the Cache Timeout Interval field, enter the number of seconds after which entities become stale. The default value for this property is 120 seconds.
7. Click the Apply button.

Monitoring the External Entity Cache

A set of statistics that describes the external entity cache is available for you to use to monitor the effectiveness of the cache. These statistics describe:

- The current state of the cache.
- The cumulative activity for the current session.
- The cumulative activity since the cache was created, typically when WebLogic Server started.

To monitor the external entity cache, follow these steps:

1. Under the Servers node in the left pane, click the name of the WebLogic Server for which you want to configure the external entity cache.
2. Select the Services tab in the right pane.

3. Select the XML tab. The window to configure XML properties of WebLogic Server appears in the right pane.
4. Click Monitor XML Entity Cache in the right pane.
5. Click the Current tab to view the current state of the cache, the Session tab to view the cumulative activity for the current session, and the Historical tab to view the cumulative activity since the cache was created, typically when WebLogic Server started.

Attributes and Console Screen Reference for XML

For information about an Administration Console screen and the attributes you can configure, select one of the following links:

[“XML Entity Spec Registry Entry --> Configuration” on page 621-1](#)

[“XML Entity Spec Registry Entry --> Notes” on page 622-1](#)

[“XML Entity Spec Registry Entry” on page 623-1](#)

[“XML Parser Select Registry Entry --> Configuration” on page 624-1](#)

[“XML Parser Select Registry Entry --> Notes” on page 625-1](#)

[“XML Parser Select Registry Entry” on page 626-1](#)

[“XML Registry --> Configuration” on page 627-1](#)

[“XML Registry --> Notes” on page 628-1](#)

[“XML Registry --> Target and Deploy” on page 629-1](#)

[“XML Registry” on page 630-1](#)



Index

A

- Abandon Timeout Seconds 67-2
- Accept Backlog 438-2, 483-4
- Accept Context Path In Get Real Path 465-4, 535-3
- access control lists (ACLs)
 - Compatibility security 322-30
- Access Point 589-2, 595-2
- Access Point Id 589-2, 595-2
- Acknowledge Mode 199-3
- Acknowledge Policy 155-8
- ACL Cache Negative TTL 294-2
- ACL Cache Positive TTL 294-2
- ACL Cache Size 294-2
- Acl Policy 596-2
- Activation Time 471-2
- Active Types 352-5, 409-5
- ActiveDirectoryAuthenticator
 - Configuration
 - Active Directory
 - Cache Enabled 324-3
 - Cache Size 324-3
 - Cache TTL 324-3
 - Credential 324-3
 - Host 324-2
 - Port 324-2
 - Principal 324-2
 - SSLEnabled 324-2
 - General

- Control Flag 325-3

- Description 325-2

- Name 325-2

- Version 325-2

Groups

- All Groups Filter 326-3

- Group Base DN 326-2

- Group From Name Filter 326-2

- Group Search Scope 326-2

- Static Group Name Attribute 326-3

- Static Group Object Class 326-3

Membership

- Dynamic Group Name Attribute 327-3

- Dynamic Group Object Class 327-2

- Dynamic Member URLAttribute 327-3

- Static Group DN from Member DNFilter 327-2

- Static Member DNAttribute 327-2

- Users
 - All Users Filter 328-4
 - User Base DN 328-3
 - User Dynamic Group DNAttribute 328-2
 - User From Name Filter 328-3
 - User Name Attribute 328-2
 - User Object Class 328-2
 - User Search Scope 328-3
- Adapter Classpath 276-3, 288-3
- Adapter JNDI Name 276-3, 288-3
- Administration Console
 - changing display 6-15
 - changing monitoring interval 6-13
 - changing the graph polling interval 6-13
 - creating scoped roles
 - EJB resources 428-12
 - JNDI resources 428-14
 - other resource types 428-16
 - URL (Web) resources 428-9
 - customizing tables 6-15
 - default language 6-14
 - domain configuration 6-12
 - domains 6-10
 - JMX 6-2
 - MBeans 6-2
 - monitoring 6-12
 - navigation 6-8
 - overview 6-1
 - preferences 3-1
 - starting 6-3
 - stopping WebLogic Servers from 497-28
 - tables 6-15
 - version 2-1
- Administration Port 66-2
- administration port 74-1
- Administration Port Enabled 434-10
- Administration Server
 - notification listener 253-7
 - starting 497-2
- ALERT severity level 253-5
- Algorithm Type 143-3
- All Groups Filter 326-3, 375-3, 382-3, 388-3
- All Users Filter 328-4, 378-3, 385-3, 391-3
- Allow Anonymous 596-4
- Allow Close In On Message 155-7
- Allow Non Standard Types 614-5
- Allow Shrinking 23-2, 113-5
- Apartment Threaded 461-3
- App Password 598-3
- App Password IV 598-4
- Append to classpath 434-8
- AppKey Generator 596-6
- Application
 - Configuration
 - Deployment Protocol 8-3
 - Load Order 8-5
 - Name 8-2
 - Path 8-3
 - Staging Mode 8-4
 - Staging Path 8-4
 - Notes
 - Notes 11-2
- Application Password 243-2, 577-3
- Application Rollbacks 469-3
- Application.Staging Mode 44-2
- archiving log messages 253-16
- Arguments 437-4, 519-3, 522-3
- Assign a JMS Connection Factory 232-12
- Assign a JMS Server 232-7
- Assign a Machine 268-3
- Assign a Shutdown Class 525-5
- Assign a Startup Class 525-5
- Assigning a Mail Session 274-3
- Associating a Virtual Host with a Server 540-5
- Asynchronous Mode Enabled 280-6
- Attribute MBean Name 498-3
- Attribute MBean Type 498-2

Attribute Name 498-3
 attributes of log messages 253-3, 253-6
 Audit Provider Class 365-2
 Auth Program 318-2
 Auth Protocol 308-3
 Auth Realm Name 544-2, 570-2
 Authentication Cache Size 295-2
 Authentication Cache TTLNegative 295-2
 Authentication Cache TTLPositive 295-2
 Auto Kill If Failed 435-2
 Auto Restart 435-2
 Auto Update Interval 65-1
 Autorefresh Every 3-2
 Average Commit Time 469-4
 Avg Per Entry Disk Size 633-7, 635-15,
 638-3
 Avg Per Entry Memory Size 633-6, 635-14,
 638-2
 Avg Percent Persistent 632-2, 634-10, 639-2
 Avg Percent Transient 632-3, 634-11, 639-3
 Avg Timeout 632-4, 634-12, 639-4

B

Backup Copies 362-3
 Backup Hour 362-2
 Backup Minute 362-2
 Basic Realm 296-2
 Batch Interval (milliseconds) 281-3
 Batch Size 281-2
 BEA Home 437-2
 Before Completion Iteration Limit 67-3
 Begin Time 527-2, 529-2
 Blocking Send Policy 193-11
 Blocking Time Out 588-5
 boot identity prompt
 hidden from standard out 253-24
 BridgeDestination
 Configuration
 Adapter Classpath 276-3
 Adapter JNDI Name 276-3

Name 276-2
 Properties (key=value) 276-4
 User Name 276-5
 User Password 276-5

Notes

Notes 277-1

Bytes Maximum 169-2, 176-2, 189-2, 193-2,
 208-2, 215-2
 Bytes Paging Enabled 169-5, 176-5, 189-5,
 193-5, 208-5, 215-5
 Bytes Threshold High 169-3, 176-3, 189-3,
 193-3, 208-3, 215-3
 Bytes Threshold Low 169-4, 176-4, 189-4,
 193-4, 208-4, 215-4

C

Cache Disk Size 458-2
 Cache Enabled 324-3, 362-3, 376-3, 384-3,
 390-3
 Cache Memory Size 458-2
 Cache Size 324-3, 362-3, 376-3, 384-3,
 390-3
 Cache Timeout Interval 458-2, 621-3
 Cache TTL 324-3, 362-3, 376-3, 384-3,
 390-3
 Caching Realm 366-2
 CachingRealm

Configuration

ACL

ACL Cache Negative TTL
294-2

ACL Cache Positive TTL
294-2

ACL Cache Size 294-2

Enable ACL Cache 294-2

Authentication

Authentication Cache
 Size 295-2

Authentication	Cache	User	Cache	TTLPositive
	TTLNegative			299-2
	295-2	Notes		
Authentication	Cache	Notes	300-2	
	TTLPositive	Capacity Increment	23-2, 113-3	
Enable	Authentication	Case Sensitive Cache	296-2	
	Cache	Cert Authenticators		
General	295-2	configuring	322-2	
Basic Realm	296-2	Channel Weight	483-4	
Case Sensitive	Cache	character encoding for log messages	253-6	
	296-2	Checkpoint Interval Seconds	67-4	
Name	296-2	Class Path	437-3	
Groups		ClassName	519-2, 522-2	
Enable Group	Cache	Client Cert Proxy Enabled	28-4	
Group	Cache	Client Id	155-2	
Group	Cache	Clone a Domain Log Filter	81-3	
	TTLNegative	Clone a JDBC Connection Pool	127-6	
	297-2	Clone a JMS Destination Key	232-22	
Group	Cache	Clone a Machine	268-3	
	TTLPositive	Clone a Mail Session	274-2	
	297-2	Clone a Shutdown Class	525-4	
Group	Membership	Clone a Startup Class	525-4	
	Cache	Clone a Web Server	540-2	
	TTL	Cluster	434-3	
	297-2	Configuration		
Permissions		General		
Enable	Permission	Client Cert Proxy Enabled		
	Cache	28-4		
Permission	Cache	Cluster Address	28-2	
	Size	Default Load	Algorithm	
	298-2	28-3		
Permission	Cache	Name	28-2	
	TTL-	Service	Age	Threshold
	Negative	28-3		
Permission	Cache	WebLogic	Plug-In	En-
	TTL-	abled	28-3	
	Positive			
	298-2			
Users		Multicast		
Enable	User	Multicast Address	29-2	
	Cache			
	299-2			
User	Cache			
	Size			
	299-2			
User	Cache			
	TTLNegative			
	299-2			

- Multicast Buffer Size 29-2
- Multicast Port 29-2
- Multicast Send Delay 29-2
- Multicast TTL 29-2
- Monitoring
 - Number of Servers configured for this cluster 40-2
 - Number of Servers currently participating in this cluster 40-2
- Notes
 - Notes 37-1
- Cluster Address 28-2, 483-4
- Cluster Weight 432-2
- Clusters 41-1, 57-1
- clusters
 - cloning 41-3
 - configuring 41-1
 - monitoring 41-4
- Cmp Limit 594-5
- Community 509-2, 514-2
- Community Prefix 69-4
- Compatibility security
 - Cert Authenticators 322-2
 - changing the system passwords 322-3
 - configuration steps 322-1
 - configuring
 - File realm 322-4
 - Realm Adapter Auditing Provider 322-3
 - Identity Assertion provider 322-2
 - using ACLs 322-30
- CompatibilityRealm
 - defining ACLs 322-30
- Compile Command 542-4, 573-4
- Complete Message Timeout 483-5, 484-2
- Compression Limit 588-5
- Concurrent 472-6
- Configuration
 - JMS
 - connection consumers 232-30
 - destination keys 232-21
 - file stores 232-22
 - JDBC stores 232-23, 232-25
 - message paging 235-8
 - overview 232-2
 - session pools 232-28
 - Configuration Data 302-2
 - Connection consumers, JMS 232-30
 - Connection Creation Retry Frequency 23-5, 113-10
 - Connection Factory 199-2
 - Connection Factory JNDI Name 288-4
 - Connection Filter 361-2
 - Connection Filter Rules 361-3
 - Connection Logger Enabled 361-2
 - Connection Policy 588-3, 594-3
 - Connection Pool 183-2
 - Adding a note 127-8
 - Assigning to servers or clusters 127-7
 - Cloning 127-6
 - Creating 127-4
 - Deleting 127-7
 - monitoring 149-2
 - Connection Principal Name 588-4, 594-4
 - Connection Reserve Timeout 23-4, 113-9
 - Connection URL 288-4
 - Connections High 470-2
 - ConnectorComponent
 - Configuration
 - Application.Staging Mode 44-2
 - Deployment Order 44-2
 - Name 44-2
 - Parent.Name 44-2
 - Notes
 - Notes 48-2
 - Console Context Path 66-4
 - Console Enabled 66-4
 - Control Flag 325-3, 341-3, 374-3, 380-3, 386-3, 409-4
 - Create a Domain Log Filter 81-1

Create a Foreign JMS Connection Factory 232-48
 Create a Foreign JMS Destination 232-49
 Create a Foreign JMS Server 232-46, 232-47
 Create a JMS Connection Consumer 232-30
 Create a JMS Connection Factory 232-8, 232-10
 Create a JMS Destination 232-14
 Create a JMS Destination Key 232-21
 Create a JMS Distributed Destination 232-31
 Create a JMS Distributed Queue 232-37
 Create a JMS Distributed Queue Member 232-42
 Create a JMS Distributed Topic Member 232-44
 Create a JMS File Store 232-23
 Create a JMS JDBC Store 232-26
 Create a JMS Queue 232-14
 Create a JMS Server 232-5
 Create a JMS Session Pool 232-29
 Create a JMS Template 232-18, 232-19
 Create a JMS Topic 232-16
 Create a Shutdown Class 525-2
 Create a Startup Class 525-2
 Creating a General Bridge Destination 289-9
 Creating a JMS Bridge Destination 289-7
 Creating a Messaging Bridge 289-11
 cred_maps 428-42
 Credential 308-2, 324-3, 362-2, 376-3, 384-3, 390-3
 Credential Mapping Deployment Enabled 349-3
 Credential Policy 596-3
 CRITICAL severity level 253-5
 Current Connections 470-2
 Current JMS Servers 470-2
 Custom AppKey Class 596-8
 Custom AppKey Param 596-8
 CustomRealm
 Configuration
 Configuration Data 302-2

Name 302-2
 Password 302-2
 Realm Class Name 302-2
 Notes
 Notes 303-2

D

Data Source
 Adding a note 141-7
 Assigning to servers or clusters 141-7
 Cloning 141-6
 Deleting 141-8
 Data Source Factory 141-8
 Debug Enabled 256-2
 Debug Level 69-4
 DEBUG severity level 253-6, 253-20
 Debug to Stdout 473-4
 Default AppKey 596-5
 Default Char Codeset 464-2
 Default Delivery Mode 155-4
 Default GIOP Version 464-2
 Default IIOP Password 464-3
 Default IIOP Username 464-3
 default language 6-14
 Default Load Algorithm 28-3
 Default Priority 155-3
 Default Redelivery Delay 155-4
 Default Reply Delivery Mode 614-4
 Default Retry Count 463-2
 Default Retry Interval 463-2
 Default Server Name 465-1, 535-2
 Default Time To Deliver 155-3
 Default Time To Live 155-3, 463-3
 Default Wide Char Codeset 464-3
 DefaultAdjudicator
 General
 Description 336-2, 406-2, 407-2
 Name 336-2, 406-2, 407-2
 Version 336-3, 406-2, 407-2
 DefaultAuditor

General	Type 355-4
Description 338-3	Version 355-2
Name 338-2	DefaultRoleMapper
Version 338-3	General
DefaultAuthenticator	Description 357-3
General	Name 357-3
Control Flag 341-3	Role Deployment Enabled 357-3
Description 341-2	Version 357-3
Name 341-2	Delete a Domain Log Filter 81-4
Version 341-3	Delete a JMS Distributed Queue Member 232-43
DefaultAuthorizer	Delete a JMS Distributed Topic Member 232-45
General	Delete a JMS File Store 232-23, 232-25
Description 345-2, 410-2	Delete a JMS JDBC Store 232-28
Name 345-2, 410-2	Delete a JMS Session Pool 232-30
Version 345-3, 410-2	Delete a Machine 268-3
DefaultCredentialMapper	Delete a Mail Session 274-3
General	Delete a Shutdown Class 525-8
Credential Mapping Deployment Enabled 349-3	Delete a Startup Class 525-8
Description 349-3	Delete a Web Server 540-4
Name 349-3	Deleting a Mail Session 274-3
Version 349-3	Deleting a Messaging Bridge 289-16
DefaultIdentityAsserter	deleting a server 495-5
General	Delivery Mode Override 187-4, 206-3, 213-4, 614-3
Active Types 352-5	Deployment Order 44-2, 519-3, 522-3, 571-3, 600-2
Description 352-3	Deployment Protocol 8-3
Name 352-3	Description 325-2, 336-2, 338-3, 341-2, 345-2, 349-3, 352-3, 355-2, 357-3, 374-2, 380-2, 386-2, 406-2, 407-2, 409-3, 410-2, 527-2, 529-1
Supported Types 352-5	Destination 151-2
Trusted Client Principals 352-4	Destination JNDI Name 288-4
User Name Mapper Class Name 352-4	Destination keys, JMS 232-21
Version 352-3	Destination Type 288-5
DefaultKeyStore	Direction 161-3, 617-2
General	Directory 181-4
Description 355-2	Disk Usage 637-2
Name 355-2	Display Advanced features by default 3-2
Private Keystore Location 355-3	
Private Keystore Pass Phrase 355-3	
Root CA Keystore Location 355-3	
Root CA Keystore Pass Phrase 355-4	

Display help text for each attribute 3-2	Minimum File Size 68-3
Distributed Topic 178-2	Number Of Files Limited 68-5
Document Builder Factory 624-2	Rotation Time 68-4
DocumentBuilderFactory 627-2	Rotation Type 68-3
Domain 576-3	SNMP
Configuration	Community Prefix 69-4
Applications	Debug Level 69-4
Auto Update Interval 65-1	Enabled 69-2
Filter	MIB Data Refresh Interval 69-2
Connection Filter 361-2	Server Status Check Interval Factor 69-3
Connection Filter Rules 361-3	SNMP Port 69-2
Connection Logger Enabled 361-2	Targeted Trap Destinations 69-5
General	Notes
Administration Port 66-2	Notes 73-1
Console Context Path 66-4	Security
Console Enabled 66-4	Embedded LDAP
Enable Administration Port 66-2	Backup Copies 362-3
Name 66-2	Backup Hour 362-2
Production Mode 66-3	Backup Minute 362-2
JTA	Cache Enabled 362-3
Abandon Timeout Seconds 67-2	Cache Size 362-3
Before Completion Iteration Limit 67-3	Cache TTL 362-3
Checkpoint Interval Seconds 67-4	Credential 362-2
Forget Heuristics 67-4	Master First 362-4
Max Transactions 67-3	Refresh Replica At Start-up 362-3
Max Unique Name Statistics 67-3	Filerealm
Timeout Seconds 67-2	Caching Realm 366-2
Logging	Max ACLs 366-2
File Time Span 68-4	Max Groups 366-2
	Max Users 366-2
	General

- Audit Provider Class 365-2
- Guest Disabled 365-2
- Passwords
 - Lockout Cache Size 364-3
 - Lockout Duration 364-3
 - Lockout Enabled 364-2
 - Lockout Reset Duration 364-3
 - Lockout Threshold 364-2
 - Minimum Password Length 364-2
- domain 74-1
- domain log files
 - defined 253-7
- DomainLogFilter
 - Configuration
 - Name 77-2
 - Severity Level 77-3
 - SubSystem Names 77-4
 - User Ids 77-4
 - Notes
 - Notes 78-1
- domain-wide administration port 74-1
- Driver 313-2
- Driver Class Name 139-2
- Driver Classname 114-3
- Durability Enabled 280-7
- Duration 465-2
- Dynamic Group Name Attribute 327-3, 377-3, 383-3, 389-3
- Dynamic Group Object Class 327-2, 377-2, 383-2, 389-2
- Dynamic Member URLAttribute 327-3, 377-3, 383-3, 389-3

E

EJB

- Creating a JDBC Data Source Factory

- 141-8
- Targets
 - Servers
 - Targets-Server 94-1
- EJB Name 584-2
- EJB resources
 - creating scoped roles 428-12
- EJBComponent
 - Configuration
 - Compiler options
 - Extra Rmic Options 84-2
 - Force Generation 84-3
 - Java Compiler 84-2
 - Keep Generated Source Files 84-3
 - General
 - Load Order 85-2
 - Name 85-1
 - Staging Mode 85-2
- Notes
 - Notes 97-1
- EMERGENCY severity level 253-5
- Emulate Two-Phase Commit for non-XA
 - Driver 136-5
- Enable ACL Cache 294-2
- Enable Administration Port 66-2
- Enable Authentication Cache 295-2
- Enable Certificate Authentication 577-5
- Enable COM 461-2
- Enable Default JMS Connection Factories 454-3
- Enable Group Cache 297-2
- Enable HTTP Logging 446-2
- Enable IIOP 464-1
- Enable JDBC Logging 447-2
- Enable JSP Line Numbers 542-5, 573-5
- Enable Keepalives 465-2
- Enable Memory Logging 461-2
- Enable Native IO 438-2

- Enable Native Mode 461-2
- Enable Permission Cache 298-2
- Enable Post-Bind GID 261-2
- Enable Post-Bind UID 261-2
- Enable Security Context 577-5
- Enable Session Monitoring 542-4, 573-4
- Enable SSL 308-3
- Enable Store 186-3, 211-3
- Enable Tunneling 484-2
- Enable User Cache 299-2
- Enabled 69-2
- Enabled Servers 498-4, 501-2, 504-2, 507-5, 512-2
- End Time 527-2, 529-2
- EntityURI 621-2
- er 322-2
- Error Destination 188-4, 207-4, 214-4
- ERROR severity level 253-5
- ExecuteQueue
 - Configuration
 - Name 489-2
 - Queue Length 489-2
 - Queue Length Threshold Percent 489-2
 - Thread Count 489-3
 - Thread Priority 489-3
 - Threads Increase 489-3
 - Threads Maximum 489-3
 - Threads Minimum 489-3
 - Notes
 - Notes 490-2
- Expiration Logging Property 229-4, 230-4, 231-4
- Expiration Policy 229-3, 230-3, 231-3
- Expiration Scan Interval 192-5
- External Entity Cache
 - Configuring 640-7
- External Entity Resolution
 - Configuring 640-5
- External Listen Address 434-10, 483-3
- External Listen Port 483-3

- Extra Rmic Options 84-2
- Extra rmic Options 434-9

F

- Factory Name 139-3
- Failover Addresses 241-2, 576-3
- Failure is fatal 522-3
- Federation Name 595-4
- Federation URL 595-3
- file size, log files 253-16
- File stores, JMS 232-22
- File Time Span 68-4, 473-7
- filenames
 - log files 253-25
- FileT3
 - Configuration
 - Name 105-1
 - Path 105-2
 - Notes
 - Notes 106-1
- FldTbl classes 598-2
- FldTbl32 classes 598-2
- Flow Control Enabled 154-4
- Flow Interval (seconds) 154-3
- Flow Maximum 154-2
- Flow Minimum 154-3
- Flow Steps 154-4
- Flush Every 446-6
- Force Generation 84-3
- ForeignJMSConnectionFactory
 - Configuration
 - Local JNDI Name 223-2
 - Name 223-2
 - Password 223-3
 - Remote JNDI Name 223-2
 - User Name 223-3
 - Notes
 - Notes 220-2
- ForeignJMSDestination
 - Configuration

- Local JNDI Name 222-2
- Name 222-2
- Remote JNDI Name 222-2
- Notes
 - Notes 219-2
- ForeignJMSSEServer
 - Configuration
 - JNDI Connection URL 225-3
 - JNDI Initial Context Factory 225-2
 - JNDI Properties 225-3
 - Name 225-2
 - Notes
 - Notes 226-2
- Forget Heuristics 67-4
- Format 446-3
- Forward Delay (seconds) 167-4
- Frontend Host 465-3
- Frontend HTTP Port 465-3
- Frontend HTTPS Port 465-3

G

- Gc Algorithm 472-4
- GCHandles Compaction 472-6
- Generational 472-6
- Graceful Shutdown Timeout 439-3
- Group Base DN 326-2, 375-2, 382-2, 388-2
- Group Cache Size 297-2
- Group Cache TTLLNegative 297-2
- Group Cache TTLPositive 297-2
- Group DN 307-2
- Group From Name Filter 326-2, 375-2, 382-2, 388-2
- Group Is Context 307-2
- Group Membership Cache TTL 297-2
- Group Name Attribute 307-2
- Group Search Scope 326-2, 375-2, 382-2, 388-2
- Group Username Attribute 307-2
- Guest Disabled 365-2

H

- Health Check Interval 435-3
- Health Check Timeout 435-3
- Host 324-2, 376-2, 384-2, 390-2, 514-2
- Http Enabled For This Protocol 483-5
- HTTP Log File Name 446-2
- HTTP Max Message Size 465-5
- HTTP Message Timeout 465-5
- HTTP protocol, configuring 495-7
- HTTPS Duration 465-2
- Https Keep Alive Secs 535-3

I

- Identity Assertion providers
 - Compatibility security 322-2
- Idle Connection Timeout 483-6, 484-2
- Idle Threads 468-1
- Ignore Sessions During Shutdown 439-3
- IIOP Protocol, configuring 495-9
- In 41-2
- Inactive Connection Timeout 23-4, 113-11
- Incremental 472-7
- Incremental Delay (seconds) 279-3
- Index Directories 542-4, 545-2, 572-2, 573-3
- INFO severity level 253-5
- Init Table Name 23-5, 113-12
- Initial Capacity 23-2, 113-2
- Initial Context Factory 288-4
- Instrument Stack Traces 473-8
- Interface Address 432-3
- Interoperate 588-5
- Invalid Login Attempts Total Count 405-5
- Invalid Login Users High Count 405-5
- Invalid Logins High 467-3
- IPlanetAuthenticator
 - Configuration
 - General
 - Control Flag 374-3
 - Description 374-2

-
- Name 374-2
 - Version 374-2
 - Groups
 - All Groups Filter 375-3
 - Group Base DN 375-2
 - Group From Name Filter 375-2
 - Group Search Scope 375-2
 - Static Group Name Attribute 375-3
 - Static Group Object Class 375-3
 - IPlanet LDAP
 - Cache Enabled 376-3
 - Cache Size 376-3
 - Cache TTL 376-3
 - Credential 376-3
 - Host 376-2
 - Port 376-2
 - Principal 376-2
 - SSLEnabled 376-2
 - Membership
 - Dynamic Group Name Attribute 377-3
 - Dynamic Group Object Class 377-2
 - Dynamic Member URLAttribute 377-3
 - Static Group DN's from Member DNFilter 377-2
 - Static Member DNAttribute 377-2
 - Users
 - All Users Filter 378-3
 - User Base DN 378-3
 - User Dynamic Group DNAttribute 378-2
 - User From Name Filter 378-3
 - User Name Attribute 378-2
 - User Object Class 378-2
 - User Search Scope 378-3
- J**
- Java Compiler 84-2, 434-6
 - Java Home 437-2
 - Java startup arguments
 - Java Virtual Machines. See JVMs
 - JavaMail
 - API version 1.1.3 272-1, 273-1, 274-1, 456-1
 - JDBC
 - Specifying log file 253-27
 - JDBC Connection Pool
 - Adding a note 127-8
 - Assigning to servers or clusters 127-7
 - Cloning 127-6
 - Creating 127-4
 - Deleting 127-7
 - monitoring 149-2
 - JDBC Data Source
 - Adding a note 141-7
 - Assigning to servers or clusters 141-7
 - Cloning 141-6
 - Deleting 141-8
 - JDBC Data Source Factory 141-8
 - JDBC Log File Name 447-2
 - JDBC MultiPool
 - Adding a note 149-4
 - Assigning to servers or clusters 149-3
 - Cloning 149-3

- Creating 149-2
- Deleting 149-4
- JDBC stores, JMS 232-23, 232-25
- JDBCConnectionPool
 - Configuration
 - Connections
 - Allow Shrinking 23-2, 113-5
 - Capacity Increment 23-2, 113-3
 - Connection Creation Retry Frequency 23-5, 113-10
 - Connection Reserve Timeout 23-4, 113-9
 - Inactive Connection Timeout 23-4, 113-11
 - Init Table Name 23-5, 113-12
 - Initial Capacity 23-2, 113-2
 - Login Delay 113-5
 - Maximum Capacity 23-2, 113-2
 - Maximum connections made unavailable 23-3, 113-11
 - Maximum waiting for connection 23-3, 113-11
 - Prepared Statement Cache Type 113-3
 - Shrink Frequency 23-3, 113-5
 - Statement Cache Size 23-6, 113-4
 - Test Created Connections

- 23-3, 113-8
- Test Frequency 23-4, 113-7
- Test Released Connections 23-4, 113-8
- Test Reserved Connections 23-3, 113-7
- Test Table Name 23-5, 113-12

General

- Driver Classname 114-3
- Name 114-2
- Open String Password 114-4
- Password 114-3
- Properties 114-3
- URL 114-2

JDBCDataSource

Configuration

- JNDI Name 129-2
- Name 129-2
- Pool Name 129-2
- Row Prefetch Enabled 129-3
- Row Prefetch Size 129-3
- Stream Chunk Size 129-4

JDBCDataSourceFactory

Configuration

- Driver Class Name 139-2
- Factory Name 139-3
- Name 139-2
- Properties 139-3
- URL 139-2
- User Name 139-2

JDBCMultiPool

Configuration

General

- Algorithm Type 143-3
- Name 143-2

Pools

Pool List 144-2	Initial Context Factory 288-4
Notes	Name 288-2
Notes 145-2	User Name 288-5
JDBCTxDataSource	User Password 288-5
Configuration	Notes
Emulate Two-Phase Commit for non-XA Driver 136-5	Notes 286-1
JNDI Name 136-2	JMSConnectionConsumer
Name 136-2	Configuration
Row Prefetch Enabled 136-3	Destination 151-2
Row Prefetch Size 136-3	Messages Maximum 151-2
Stream Chunk Size 136-4	Name 151-2
JDK Vendor 471-2	Selector 151-2
JDK Version 471-2	Notes
JMS	Notes 152-2
configuration	JMSConnectionFactory
message paging 235-8	Configuration
configuring	Flow Control
connection consumers 232-30	Flow Control Enabled 154-4
destination keys 232-21	Flow Interval (seconds) 154-3
file stores 232-22	Flow Maximum 154-2
JDBC stores 232-23, 232-25	Flow Minimum 154-3
overview 232-2	Flow Steps 154-4
session pools 232-28	Send Timeout (milliseconds) 154-5
monitoring 234-2	General
tuning 235-2	Acknowledge Policy 155-8
message paging 235-8	Allow Close In On Message 155-7
persistent stores 235-2	Client Id 155-2
JMS Factory 613-2	Default Delivery Mode 155-4
JMS Thread Pool Size 454-3	Default Priority 155-3
Jms To Tux Priority Map 615-2	Default Redelivery Delay 155-4
JMSBridgeDestination	Default Time To Deliver
Configuration	
Adapter Classpath 288-3	
Adapter JNDI Name 288-3	
Connection Factory JNDI Name 288-4	
Connection URL 288-4	
Destination JNDI Name 288-4	
Destination Type 288-5	

155-3	Bytes Threshold High
Default Time To Live	169-3
155-3	Bytes Threshold Low
JNDIName 155-2	169-4
Load Balancing Enabled	Messages Maximum 169-6
155-9	Messages Paging Enabled
Messages Maximum 155-5	169-10
Name 155-2	Messages Threshold High
Overrun Policy 155-6	169-7
Server Affinity Enabled	Messages Threshold Low
155-9	169-8
Transactions	Notes
Transaction Timeout 156-2	Notes 170-2
XA Connection Factory	JMSDistributedQueueMember
Enabled 156-3	Configuration
Notes	JMSQueue 171-2
Notes 157-2	Name 171-2
JMSDestinationKey	Weight 171-2
Configuration	JMSDistributedTopic
Direction 161-3	Configuration
Key Type 161-3	General
Name 161-2	JNDI Name 174-2
Notes	Load Balancing Policy
Notes 162-2	174-3
JMSDistributedQueue	Name 174-2
Configuration	Thresholds & Quotas
General	Bytes Maximum 176-2
Forward Delay (seconds)	Bytes Paging Enabled
167-4	176-5
JNDI Name 167-2	Bytes Threshold High
Load Balancing Policy	176-3
167-3	Bytes Threshold Low
Name 167-2	176-4
Thresholds & Quotas	Messages Maximum 176-6
Bytes Maximum 169-2	Messages Paging Enabled
Bytes Paging Enabled	176-10
169-5	

Messages Threshold High	187-4
176-7	Priority Override 187-2
Messages Threshold Low	Time To Deliver Override
176-8	187-3
Notes	Time To Live Override
Notes 177-2	187-2
JMSDistributedTopicMember	Redelivery
Configuration	Error Destination 188-4
Distributed Topic 178-2	Redelivery Delay Over-
Name 178-2	ride 188-2
Weight 178-2	Redelivery Limit 188-3
JMSFileStore	Thresholds & Quotas
Configuration	Bytes Maximum 189-2
Directory 181-4	Bytes Paging Enabled
Name 181-2	189-5
Synchronous Write Policy 181-3,	Bytes Threshold High
181-4	189-3
Notes	Bytes Threshold Low
Notes 182-1	189-4
JMSJDBCStore	Maximum Message Size
Configuration	189-10
Connection Pool 183-2	Messages Maximum 189-6
Name 183-2	Messages Paging Enabled
Prefix Name 183-3	189-9
Notes	Messages Threshold High
Notes 184-2	189-7
JMSQueue 171-2	Messages Threshold Low
Configuration	189-8
Expiration Policy	Notes
Expiration Logging Prop-	Notes 191-2
erty 229-4	JMSServer
Expiration Policy 229-3	Configuration
General	General
Enable Store 186-3	Expiration Scan Interval
JNDIName 186-2	192-5
Name 186-2	Name 192-2
Template 186-4	
Overrides	
Delivery Mode Override	

Paging Store 192-3	Expiration Policy
Store 192-2	Expiration Logging Property 230-4
Temporary Template 192-4	Expiration Policy 230-3
Thresholds & Quotas	General
Blocking Send Policy 193-11	Name 186-4, 205-2, 211-4
Bytes Maximum 193-2	Override
Bytes Paging Enabled 193-5	Delivery Mode Override 206-3
Bytes Threshold High 193-3	Priority Override 206-2
Bytes Threshold Low 193-4	Time To Deliver Override 206-3
Maximum Message Size 193-10	Time To Live Override 206-2
Messages Maximum 193-6	Redelivery
Messages Paging Enabled 193-9	Error Destination 207-4
Messages Threshold High 193-7	Redelivery Delay Override 207-2
Messages Threshold Low 193-8	Redelivery Limit 207-3
Notes	Thresholds & Quotas
Notes 194-2	Bytes Maximum 208-2
JMSSessionPool	Bytes Paging Enabled 208-5
Configuration	Bytes Threshold High 208-3
Acknowledge Mode 199-3	Bytes Threshold Low 208-4
Connection Factory 199-2	Maximum Message Size 208-10
Listener Class 199-2	Messages Maximum 208-6
Name 199-2	Messages Paging Enabled 208-9
Sessions Maximum 199-3	Messages Threshold High 208-7
Transacted 199-3	Messages Threshold Low
Notes	
Notes 200-2	
JMSTemplate	
Configuration	

	208-8		215-3
Notes		Bytes	Threshold Low
Notes	209-2		215-4
JMSTopic		Maximum	Message Size
Configuration			215-10
Expiration Policy		Messages	Maximum
Expiration Logging Property	231-4	Messages	Paging Enabled
Expiration Policy	231-3		215-9
General		Messages	Threshold High
Enable Store	211-3		215-7
JNDIName	186-2, 211-2	Messages	Threshold Low
Name	211-2		215-8
Template	211-4	Notes	
Multicast		Notes	217-2
Multicast Address	212-2	JMX	6-2
Multicast Port	212-2	JMX	notifications. <i>See</i> notifications
Multicast TTL	212-2	JNDI	Connection URL
Overrides			225-3
Delivery Mode Override	213-4	JNDI	Factory
Priority Override	213-2		613-2
Time To Deliver Override	213-3	JNDI	Initial Context Factory
Time To Live Override	213-2		225-2
Redelivery		JNDI	Name
Error Destination	214-4		129-2, 136-2, 167-2, 174-2
Redelivery Delay Override	214-2	JNDI	Properties
Redelivery Limit	214-3		225-3
Thresholds & Quotas		JNDI	resources
Bytes Maximum	215-2		creating scoped roles
Bytes Paging Enabled	215-5	JNDIName	155-2, 186-2, 211-2, 270-2
Bytes Threshold High		Jolt	
		Assign a Connection Pool	249-3
		Cloning Connection Pools	249-3
		Configuring Connection Pools	249-3
		Delete a Connection Pool	249-3
		Monitor Connection Pool Instances	249-4
		JoltConnectionPool	
		Configuration	
		Addresses	
		Failover Addresses	241-2
		Primary Addresses	241-2
		General	
		Maximum Pool Size	242-2

- Minimum Pool Size 242-2
- Name 242-1
- Recv Timeout 242-2
- Security Context Enabled 242-2
- User
 - Application Password 243-2
 - User Name 243-1
 - User Password 243-2
 - User Role 243-2
- Notes
 - Notes 244-1
- JTA
 - Configuring 237-2
 - Constraining Transaction Recovery
 - Service migration targets 237-22
 - Migrating the Transaction Recovery
 - Service 237-21, 237-23
 - Transaction Recovery Service owner 237-23
- JVMs
 - messages to standard out

K

- Keep Alive Enabled 535-3
- Keep Alive Secs 535-3
- Keep Generated Source Files 84-3, 542-4, 573-4
- Key Type 161-3

L

- language 6-14
- LDAPRealm
 - Configuration
 - General

- Name 306-2
- Realm Class Name 306-2
- Groups
 - Group DN 307-2
 - Group Is Context 307-2
 - Group Name Attribute 307-2
 - Group Username Attribute 307-2
- LDAP Realm V1 (Deprecated)
 - Auth Protocol 308-3
 - Credential 308-2
 - Enable SSL 308-3
 - LDAPURL 308-2
 - Principal 308-2
- Users
 - User Authentication 309-2
 - User DN 309-3
 - User Name Attribute 309-3
 - User Password Attribute 309-3

Notes

- Notes 310-2
- LDAPURL 308-2
- Listen Address 256-2, 262-2, 434-3
- Listen Port 256-2, 262-2, 434-4, 434-5
- Listen Port Enabled 434-4
- listen port,setting 495-13
- Listener Class 199-2
- listeners. *See* notification listeners
- Load Balancing Enabled 155-9
- Load Balancing Policy 167-3, 174-3
- Load Order 8-5, 85-2, 571-2
- Local Access Point 584-2, 586-2, 592-2, 595-2
- Local Administration Port Override (0 no override) 434-11
- Local JNDI Name 222-2, 223-2

- local log files
 - defined 253-7
- Local Password 592-2
- Local Password IV 592-2
- Locked Users 467-3
- Locked Users Current Count 405-5
- Lockout Cache Size 364-3, 405-4
- Lockout Duration 364-3, 405-2
- Lockout Enabled 364-2, 405-2
- Lockout GCThreshold 405-4
- Lockout Reset Duration 364-3, 405-4
- Lockout Threshold 364-2, 405-3
- Log
 - General Settings 253-25
 - HTTP Log File Settings 253-28
 - Rotation 253-16
 - Severity Threshold 253-25
 - Specifying JDBC log file 253-27
 - Specifying Location of 253-25
 - Viewing Server Logs 253-9
 - Working with Domain Log Filters 81-1
- Log Buffer Size 446-3
- Log File BufferK Bytes 536-3
- Log File Flush Secs 536-5
- Log File Format 536-3
- Log File Name 536-2
- log files
 - rotating 253-16
- log message format
 - attributes 253-3
 - output in log file 253-6
- Log Rotation Period Mins 536-4
- Log Rotation Time Begin 536-5
- Log Rotation Type 536-4
- Log to Domain log file 445-2
- Log to Stdout 473-4
- Logging Enabled 536-2
- Login Attempts While Locked Total Count 405-5
- Login Delay 113-5
- Login Timeout 438-3, 483-5

- Low Memory GCThreshold 438-4
- Low Memory Granularity Level 438-4
- Low Memory Sample Size 438-4
- Low Memory Time Interval 438-5

M

- Machine 434-3
 - Configuration
 - General
 - Name 255-2
 - Node Manager
 - Debug Enabled 256-2
 - Listen Address 256-2
 - Listen Port 256-2
 - Notes
 - Notes 259-1
- machine
 - assigning 268-3
 - cloning 268-3
 - configuring 268-1
 - deleting 268-3
- MachineName log message attribute 253-4
- MailSession
 - Configuration
 - JNDIName 270-2
 - Name 270-2
 - Properties 270-2
 - Notes
 - Notes 271-2
- managed server
 - starting 497-5
- Managed Server Independence Enabled 438-5
- managed servers
 - starting with scripts 497-8
- Master First 362-4
- Max ACLs 366-2
- Max Encryption Level 590-3, 596-4
- Max Entry Memory Size 633-6, 635-14,

638-2
 Max Entry Timeout 632-3, 634-11, 639-3
 Max Groups 366-2
 Max Heap Size 472-3
 Max Log File Size KBytes 536-3
 Max Post Size 465-2, 535-2
 Max Post Time Secs 535-2
 Max Restarts within Interval 435-3
 Max Retries 588-4, 594-5
 Max Transactions 67-3
 Max Unique Name Statistics 67-3
 Max Users 366-2
 Maximum Capacity 23-2, 113-2
 Maximum connections made unavailable
 23-3, 113-11
 Maximum Delay (seconds) 279-4
 Maximum Encryption Level 577-4
 Maximum Idle Time (seconds) 280-5
 Maximum Log File Size 446-4
 Maximum Message Size 189-10, 193-10,
 208-10, 215-10, 483-7, 484-3
 Maximum Open Sockets 438-2
 Maximum Pool Size 242-2, 576-4
 Maximum waiting for connection 23-3,
 113-11
 MBeans 6-2
 Memory Usage 637-2
 message attributes. *See* log message format
 Message Ids 507-5
 Message Substring 507-5
 Messages Maximum 151-2, 155-5, 169-6,
 176-6, 189-6, 193-6, 208-6, 215-6
 Messages Paging Enabled 169-10, 176-10,
 189-9, 193-9, 208-9, 215-9
 Messages Threshold High 169-7, 176-7,
 189-7, 193-7, 208-7, 215-7
 Messages Threshold Low 169-8, 176-8,
 189-8, 193-8, 208-8, 215-8
 messages. *See* log messages 253-6
 MessageText log message attribute 253-4
 MessageId log message attribute 253-4

Messaging Bridge Thread Pool Size 451-2
 MessagingBridge
 Configuration
 Connection Retry
 Incremental Delay (sec-
 onds) 279-3
 Maximum Delay (sec-
 onds) 279-4
 Minimum Delay (sec-
 onds) 279-2
 General
 Asynchronous Mode En-
 abled 280-6
 Durability Enabled 280-7
 Maximum Idle Time (sec-
 onds) 280-5
 Name 280-2
 QOS Degradation Al-
 lowed 280-5
 Quality Of Service 280-4
 Selector 280-3
 Source Destination 280-2
 Started 280-8
 Target Destination 280-3
 Transactions
 Batch Interval (millisec-
 onds) 281-3
 Batch Size 281-2
 Transaction Timeout 281-2
 Notes
 Notes 282-1
 Meta Data File 617-3
 MIB Data Refresh Interval 69-2
 Min Encryption Level 590-3, 596-3
 Min Entry Memory Size 633-6, 635-14,
 638-2
 Min Entry Timeout 632-3, 634-11, 639-3

- Minimum Delay (seconds) 279-2
- Minimum Encryption Level 577-3
- Minimum File Size 68-3, 473-6
- Minimum Password Length 364-2
- Minimum Pool Size 242-2, 576-3
- Modulus 500-5
- modulus for CounterMonitor objects 516-22
- Monitor All Active JMS Servers 232-8, 234-3
- Monitor All Active JMS Session Pools 234-4
- Monitor Durable Subscribers 234-5
- Monitor Durable Subscribers 232-7, 234-5
- Monitored Attribute Name 500-3, 503-3, 511-3
- Monitored MBean Name 500-3, 503-3, 511-3
- Monitored MBean Type 500-2, 503-2, 511-2
- Monitoring
 - All active JDBC connection pools 149-2
 - JMS 234-2
 - objects 234-2
- monitoring 6-12
 - changing monitoring interval 6-13
 - changing the graph polling interval 6-13
- Monitoring All Active JMS Connections 232-8, 234-2
- Monitoring All Active Messaging Bridges 289-23
- MSI File Replication Enabled 438-6
- Multicast Address 29-2, 212-2
- Multicast Buffer Size 29-2
- Multicast Port 29-2, 212-2
- Multicast Send Delay 29-2
- Multicast TTL 29-2, 212-2
- MultiPool
 - Adding a note 149-4
 - Assigning to servers or clusters 149-3
 - Cloning 149-3
 - Creating 149-2
 - Deleting 149-4

N

- Name 8-2, 28-2, 44-2, 66-2, 77-2, 85-1, 105-1, 114-2, 129-2, 136-2, 139-2, 143-2, 151-2, 155-2, 161-2, 167-2, 171-2, 174-2, 178-2, 181-2, 183-2, 186-2, 186-4, 192-2, 199-2, 205-2, 211-2, 211-4, 222-2, 223-2, 225-2, 242-1, 255-2, 261-2, 270-2, 276-2, 280-2, 288-2, 296-2, 302-2, 306-2, 311-2, 314-2, 318-2, 325-2, 336-2, 338-2, 341-2, 345-2, 349-3, 352-3, 355-2, 357-3, 374-2, 380-2, 386-2, 406-2, 407-2, 409-2, 410-2, 434-2, 483-2, 489-2, 498-2, 500-2, 503-2, 507-2, 509-2, 511-2, 514-2, 519-2, 522-2, 534-2, 543-2, 571-2, 576-2, 600-2, 627-2
- navigation tree 6-8
- Network Address 589-3, 595-3
- NetworkAccessPoint
 - Configuration
 - Accept Backlog 483-4
 - Channel Weight 483-4
 - Cluster Address 483-4
 - Complete Message Timeout 483-5
 - External Listen Address 483-3
 - External Listen Port 483-3
 - Http Enabled For This Protocol 483-5
 - Idle Connection Timeout 483-6
 - Login Timeout 483-5
 - Maximum Message Size 483-7
 - Name 483-2
 - Outbound Enabled 483-4
 - Protocol 483-2
 - Tunneling Client Ping 483-6
 - Tunneling Client Timeout 483-7
 - Tunneling Enabled 483-6
- Note
 - JDBC Connection Pool 127-8

JDBC Data Source 141-7	Class 383-2
JDBC MultiPool 149-4	Dynamic Member
Notes 11-2, 37-1, 48-2, 73-1, 78-1, 97-1, 106-1, 145-2, 152-2, 157-2, 162-2, 170-2, 177-2, 182-1, 184-2, 191-2, 194-2, 200-2, 209-2, 217-2, 219-2, 220-2, 226-2, 244-1, 259-1, 267-1, 271-2, 277-1, 282-1, 286-1, 300-2, 303-2, 310-2, 312-2, 316-2, 319-2, 430-1, 490-2, 521-1, 524-1, 537-2, 550-2, 565-2, 578-1, 609-3, 622-2, 625-2, 628-2	URLAttribute 383-3
NOTICE severity level 253-5	Static Group DN's from Member DNFilter 383-2
notification listeners for a domain message log 253-7	Static Member DNAttribute 383-2
Notify Differ 511-4	Novell LDAP
Notify Match 511-4	Cache Enabled 384-3
NovellAuthenticator	Cache Size 384-3
Configuration	Cache TTL 384-3
General	Credential 384-3
Control Flag 380-3	Host 384-2
Description 380-2	Port 384-2
Name 380-2	Principal 384-2
Version 380-2	SSLEnabled 384-2
Groups	Users
All Groups Filter 382-3	All Users Filter 385-3
Group Base DN 382-2	User Base DN 385-3
Group From Name Filter 382-2	User Dynamic Group DNAttribute 385-2
Group Search Scope 382-2	User From Name Filter 385-3
Static Group Name Attribute 382-3	User Name Attribute 385-2
Static Group Object Class 382-3	User Object Class 385-2
Membership	User Search Scope 385-3
Dynamic Group Name Attribute 383-3	NT Authentication Host 461-2
Dynamic Group Object	NTRealm
	Configuration
	Name 311-2
	Primary Domain 311-2
	Realm Class Name 311-2
	Notes
	Notes 312-2

<ul style="list-style-type: none"> <ul style="list-style-type: none"> Migration <ul style="list-style-type: none"> Begin Time 529-2 Description 529-1 End Time 529-2 Status 529-1 Status <ul style="list-style-type: none"> Begin Time 527-2 Description 527-2 End Time 527-2 Status 527-2 Number Of Daemon Threads 472-9 Number Of Files Limited 68-5 number of log files 253-16 Number Of Processors 472-7 Number of Servers configured for this cluster 40-2 Number of Servers currently participating in this cluster 40-2 	<ul style="list-style-type: none"> <ul style="list-style-type: none"> <ul style="list-style-type: none"> tribute 388-3 Static Group Object Class 388-3 Membership <ul style="list-style-type: none"> Dynamic Group Name Attribute 389-3 Dynamic Group Object Class 389-2 Dynamic Member URLAttribute 389-3 Static Group DN's from Member DNFilter 389-2 Static Member DNAttribute 389-2 Open LDAP <ul style="list-style-type: none"> Cache Enabled 390-3 Cache Size 390-3 Cache TTL 390-3 Credential 390-3 Host 390-2 Port 390-2 Principal 390-2 SSLEnabled 390-2 Users <ul style="list-style-type: none"> All Users Filter 391-3 User Base DN 391-3 User Dynamic Group DNAttribute 391-2 User From Name Filter 391-3 User Name Attribute 391-2 User Object Class 391-2 User Search Scope 391-3
---	---

O

- Offset 500-4
- Oid Root 509-2
- Open String Password 114-4
- OpenLDAPAuthenticator
 - Configuration
 - General
 - Control Flag 386-3
 - Description 386-2
 - Name 386-2
 - Version 386-2
 - Groups
 - All Groups Filter 388-3
 - Group Base DN 388-2
 - Group From Name Filter 388-2
 - Group Search Scope 388-2
 - Static Group Name At-

- Operating System 471-2
- OS Version 471-2
- other resource types
 - creating scoped roles 428-16
- Outbound Enabled 483-4
- Overrun Policy 155-6

P

- Paging messages, JMS 235-8
- Paging Store 192-3
- Parallel 472-7
- Parent.Name 44-2
- Parser
 - Configuring for a document type 640-3
 - Configuring other than built-in 640-2
- Parser Class Name 624-3
- Password 114-3, 223-3, 302-2, 313-2, 437-5
- password prompt
 - hidden from standard out 253-24
- passwords
 - changing the system password in
 - Compatibility security 322-4
 - use when starting WebLogic Server 497-14
- Path 8-3, 105-2
- pathnames
 - log files 253-25
- Percent Rejected 631-2, 636-18
- Permission Cache Size 298-2
- Permission Cache TTLNegative 298-2
- Permission Cache TTLPositive 298-2
- Poll for Graph Data Every 3-2
- Polling Interval 500-3, 503-3, 511-3
- Pool List 144-2
- Pool Name 129-2
- Port 324-2, 376-2, 384-2, 390-2, 509-2, 514-3
- port
 - administration 74-1
- Post Timeout 465-2

- Post Timeout Secs 535-2
- Post-Bind GID 261-2
- Post-Bind UID 261-2
- Prefer Web Inf Classes 544-2, 570-2
- preferences 3-1
- Preferred Secondary Group 432-2
- Prefetch Enumeration 461-3
- Prefix Name 183-3
- Prepared Statement Cache Type 113-3
- Prepend to classpath 434-8
- Primary Addresses 241-2, 576-2
- Primary Domain 311-2
- Principal 308-2, 324-2, 376-2, 384-2, 390-2
- Priority Override 187-2, 206-2, 213-2
- Private Keystore Location 355-3
- Private Keystore Pass Phrase 355-3
- Production Mode 66-3
- Properties 114-3, 139-3, 270-2
- Properties (key=value) 276-4
- Protocol 483-2
- Public Id 621-2, 624-2

Q

- QOS Degradation Allowed 280-5
- Quality Of Service 280-4
- Queue Length 489-2
- Queue Length Threshold Percent 489-2

R

- RDBMSRealm
 - Configuration
 - Database
 - Driver 313-2
 - Password 313-2
 - URL 313-2
 - User Name 313-2
 - General
 - Name 314-2

Realm Class 314-2	Version 409-3
Schema	Recv Timeout 242-2
Schema Properties	Redelivery Delay Override 188-2, 207-2, 214-2
(key=value) 315-2	Redelivery Limit 188-3, 207-3, 214-3
Notes	Refresh Replica At Startup 362-3
Notes 316-2	relative pathnames
Realm	log files 253-25
User Lockout	Reload Period 542-3, 545-2, 572-2, 573-3, 573-4
Invalid Login Attempts Total Count 405-5	Remember Last Tab 3-2
Invalid Login Users High Count 405-5	Remote Access Point 592-2
Locked Users Current Count 405-5	Remote Access Point List 586-2
Lockout Cache Size 405-4	Remote JNDI Name 222-2, 223-2
Lockout Duration 405-3	Remote Name 584-3, 586-2
Lockout Enabled 405-2	Remote Password 592-3
Lockout GCThreshold 405-4	Remote Password IV 592-3
Lockout Reset Duration 405-4	remote starting and stopping
Lockout Threshold 405-3	configuration of 497-7
Login Attempts While Locked Total Count 405-5	Removing an Associated Virtual Host 540-6
Unlocked Users Total Count 405-6	Replication Group 432-2
User Lockout Total Count 405-5	Reply Q 617-3
Realm Adapter Auditing provider	Resource Name 584-2, 586-2
configuring 322-3	Resource Rollbacks 469-3
use 322-3	resources
Realm Adapter Authentication provider	JNDI
configuring 322-3	creating scoped roles 428-14
Realm Adapter Authentication provider	other types
configuring identity assertion 322-2	creating scoped roles 428-16
Realm Class 314-2	URL (Web) and EJB
Realm Class Name 302-2, 306-2, 311-2	creating scoped roles 428-9, 428-12
Realm Classname 318-2	Restart Delay Seconds 435-3
RealmAdapterAuthenticator	Restart Interval 435-2
General	Resuming a Server 497-24
Active Types 409-5	Retries 612-2
Control Flag 409-4	Retry Delay 612-2
Description 409-3	Retry Interval 588-4, 594-4
Name 409-2	Reverse DNS Allowed 438-3
Supported Types 409-5	Role Deployment Enabled 357-3
	roles
	security

- deleting 428-16
- Root CA Keystore Location 355-3
- Root CA Keystore Pass Phrase 355-4
- Root Directory 437-2
- Root Element Tag 624-2
- Rotation Period 446-5
- Rotation Time 68-4, 446-5, 473-7
- Rotation Type 68-3, 446-4, 473-6
- Row Prefetch Enabled 129-3, 136-3
- Row Prefetch Size 129-3, 136-3
- Run before application deployments 522-4

S

- SAXParser Factory 624-3
- SAXParserFactory 627-2
- Schema Properties (key=value) 315-2
- scoped roles
 - deleting 428-16
- Security 590-2
- Security Context Enabled 242-2
- Security Policy File 437-4
- security roles
 - scoped
 - deleting 428-16
- Selector 151-2, 280-3
- Send Server Header 465-4
- Send Timeout (milliseconds) 154-5
- Server
 - cloning 495-4
 - Configuration
 - Cluster
 - Cluster Weight 432-2
 - Interface Address 432-3
 - Preferred Secondary Group 432-2
 - Replication Group 432-2
 - Deployment
 - Staging Directory Name 433-2

- Staging Mode 433-2
- Upload Directory Name 433-3

General

- Administration Port Enabled 434-10
- Append to classpath 434-8
- Cluster 434-3
- External Listen Address 434-10
- Extra rmic Options 434-9
- Java Compiler 434-6
- Listen Address 434-3
- Listen Port 434-4, 434-5
- Listen Port Enabled 434-4
- Local Administration Port Override (0 no override) 434-11
- Machine 434-3
- Name 434-2
- Prepend to classpath 434-8
- SSL Listen Port Enabled 434-4
- Startup Mode 434-8
- WebLogic Plug-In Enabled 434-7

Health Monitoring

- Auto Kill If Failed 435-2
- Auto Restart 435-2
- Health Check Interval 435-3
- Health Check Timeout 435-3
- Max Restarts within Interval 435-3

Restart Delay Seconds	Start/Stop
435-3	Graceful Shutdown Timeout 439-3
Restart Interval 435-2	Ignore Sessions During Shutdown 439-3
Remote Start	Logging
Arguments 437-4	Domain
BEA Home 437-2	Log to Domain log file 445-2
Class Path 437-3	Use log filter 445-2
Java Home 437-2	HTTP
Password 437-5	Enable HTTP Logging 446-2
Root Directory 437-2	Flush Every 446-6
Security Policy File 437-4	Format 446-3
Username 437-5	HTTP Log File Name 446-2
Tuning	Log Buffer Size 446-3
Accept Backlog 438-2	Maximum Log File Size 446-4
Enable Native IO 438-2	Rotation Period 446-5
Login Timeout 438-3	Rotation Time 446-5
Low Memory GCThreshold 438-4	Rotation Type 446-4
Low Memory Granularity Level 438-4	JDBC
Low Memory Sample Size 438-4	Enable JDBC Logging 447-2
Low Memory Time Interval 438-5	JDBC Log File Name 447-2
Managed Server Independence Enabled 438-5	JTA
Maximum Open Sockets 438-2	Transaction Log File Prefix 448-2
MSI File Replication Enabled 438-6	Server
Reverse DNS Allowed 438-3	Debug to Stdout 473-4
Socket Readers 438-2	File Time Span 473-7
Control	Instrument Stack Traces 473-8

Log to Stdout 473-4	Parallel 472-7
Minimum File Size 473-6	Total Garbage Collection Count 472-5
Rotation Time 473-7	Total Number Of Threads 472-8
Rotation Type 473-6	Total Nursery Size 472-3
Server File Name 473-3	JTA
Stdout Severity Threshold 473-5	Application Rollbacks 469-3
Monitoring	Average Commit Time 469-4
General	Resource Rollbacks 469-3
Activation Time 471-2	System Rollbacks 469-3
JDK Vendor 471-2	Timeout Rollbacks 469-2
JDK Version 471-2	Total Committed 469-2
Operating System 471-2	Total Heuristics 469-3
OS Version 471-2	Total Rolled Back 469-2
State 471-1	Total Transactions 469-2
Weblogic Version 471-2	Total Transactions Abandoned 469-3
JMS	Security
Connections High 470-2	Invalid Logins High 467-3
Current Connections 470-2	Locked Users 467-3
Current JMS Servers 470-2	Total Invalid Logins 467-2
Servers High 470-2	Total Login Attempts while Locked 467-2
Servers Total 470-3	Total Users Unlocked 467-2
Total Connections 470-2	User Lockout Total Count 467-2
JRockit	Notes
Concurrent 472-6	Notes 430-1
Gc Algorithm 472-4	Protocols
GCHandles Compaction 472-6	General
Generational 472-6	Complete Message Time-
Incremental 472-7	
Max Heap Size 472-3	
Number Of Daemon Threads 472-9	
Number Of Processors 472-7	

out 484-2	Default IIOP Password 464-3
Enable Tunneling 484-2	Default IIOP Username 464-3
Idle Connection Timeout 484-2	Default Wide Char Codeset 464-3
Maximum Message Size 484-3	Enable IIOP 464-1
Tunneling Client Ping 484-3	Transaction Mechanism 464-2
Tunneling Client Timeout 484-3	jCOM
HTTP	Apartment Threaded 461-3
Accept Context Path In Get Real Path 465-4	Enable COM 461-2
Default Server Name 465-1	Enable Memory Logging 461-2
Duration 465-2	Enable Native Mode 461-2
Enable Keepalives 465-2	NT Authentication Host 461-2
Frontend Host 465-3	Prefetch Enumeration 461-3
Frontend HTTP Port 465-3	Resuming 497-24
Frontend HTTPS Port 465-3	Server
HTTP Max Message Size 465-5	Services
HTTP Message Timeout 465-5	File T3 Targets 452-1
HTTPS Duration 465-2	Mail Targets 456-2
Max Post Size 465-2	Services
Post Timeout 465-2	Bridge
Send Server Header 465-4	Messaging Bridge Thread Pool Size 451-2
WAP Enabled 465-4	JMS
IIOP	Enable Default JMS Connection Factories 454-3
Default Char Codeset 464-2	JMS Thread Pool Size 454-3
Default GIOP Version 464-2	

- Virtual Hosts
 - Targets 462-2
- Web Services
 - Default Retry Count 463-2
 - Default Retry Interval 463-2
 - Default Time To Live 463-3
 - Store 463-2
- XML
 - Cache Disk Size 458-2
 - Cache Memory Size 458-2
 - Cache Timeout Interval 458-2
 - XMLRegistry 458-2
 - Shutting Down, *see* Stopping a Server
 - Specifying JDBC log file 253-27
 - Stopping 497-28
- server
 - deleting 495-5
 - listen port 495-13
- Server Affinity Enabled 155-9
- Server File Name 473-3
- Server session pools, JMS 232-28
- Server Status Check Interval Factor 69-3
- ServerMonitor
 - Monitoring
 - Performance 468-2
 - Idle Threads 468-1
- ServerName log message attribute 253-4
- Servers High 470-2
- Servers Total 470-3
- Service Age Threshold 28-3
- Session Cookie Max Age Secs 542-2, 573-2
- Session Invalidation Interval Secs 542-3, 573-2
- Session Timeout Secs 542-3, 573-3
- Sessions Maximum 199-3
- Severity Level 77-3, 507-3
- severity levels of log messages, defined 253-5
- Severity log message attribute 253-3
- Shrink Frequency 23-3, 113-5
- ShutdownClass
 - Configuration
 - Arguments 519-3
 - ClassName 519-2
 - Deployment Order 519-3
 - Name 519-2
 - Notes
 - Notes 521-1
- Shutting Down a Server, *see* Stopping a Server
- Single Threaded Servlet Pool Size 542-3, 544-2, 570-2, 573-3
- size of log files 253-16
- SNMP agent, WebLogic
 - setting up 516-3
- SNMP Port 69-2
- SNMPAttributeChange
 - Configuration
 - Attribute MBean Name 498-3
 - Attribute MBean Type 498-2
 - Attribute Name 498-3
 - Name 498-2
- SNMPCounterMonitor
 - Configuration
 - Modulus 500-5
 - Monitored Attribute Name 500-3
 - Monitored MBean Name 500-3
 - Monitored MBean Type 500-2
 - Name 500-2
 - Offset 500-4
 - Polling Interval 500-3
 - Threshold 500-4
 - Servers
 - Enabled Servers 498-4, 501-2
- SNMPGaugeMonitor

General	Host 514-2
Monitored Attribute Name 503-3	Name 514-2
Monitored MBean Name 503-3	Port 514-3
Monitored MBean Type 503-2	Socket Readers 438-2
Name 503-2	Source Access Point 617-4
Polling Interval 503-3	Source Destination 280-2
Threshold High 503-3	Source Name 617-4
Threshold Low 503-4	Source Qspace 617-4
Servers	SSL
Enabled Servers 504-2	administration port 74-1
SNMPLogFilter	SSL Listen Port Enabled 434-4
General	SSLEnabled 324-2, 376-2, 384-2, 390-2
Enabled Servers 507-5	stack traces in log messages 253-6, 253-7
Message Ids 507-5	Staging Directory Name 433-2
Message Substring 507-5	Staging Mode 8-4, 85-2, 433-2, 543-3, 571-3
Name 507-2	Staging Path 8-4
Severity Level 507-3	standard out
Subsystem Names 507-4	Started 280-8
User Ids 507-4	starting Administration Server 497-2
SNMPProxy	starting the Administration Console 6-3
General	startup arguments for JVMs
Community 509-2	Startup Mode 434-8
Name 509-2	StartupClass
Oid Root 509-2	Configuration
Port 509-2	Arguments 522-3
Timeout 509-3	ClassName 522-2
SNMPStringMonitor	Deployment Order 522-3
General	Failure is fatal 522-3
Monitored Attribute Name 511-3	Name 522-2
Monitored MBean Name 511-3	Run before application deployments
Monitored MBean Type 511-2	522-4
Name 511-2	Notes
Notify Differ 511-4	Notes 524-1
Notify Match 511-4	State 471-1
Polling Interval 511-3	Statement Cache Size 23-6, 113-4
String To Compare 511-3	Static Group DN's from Member DNFilter
Servers	327-2, 377-2, 383-2, 389-2
Enabled Servers 512-2	Static Group Name Attribute 326-3, 375-3,
SNMPTrapDestination	382-3, 388-3
General	Static Group Object Class 326-3, 375-3,
Community 514-2	382-3, 388-3

Static Member DNAttribute 327-2, 377-2,
383-2, 389-2
Status 527-2, 529-1
Stdout Severity Threshold 473-5
Stopping a Server 497-28
stopping WebLogic Servers 497-28
Store 192-2, 463-2
Stream Chunk Size 129-4, 136-4
String To Compare 511-3
Subsystem log message attribute 253-3
SubSystem Names 77-4
Subsystem Names 507-4
Supported Types 352-5, 409-5
Synchronous Write Policy 181-3, 181-4
System Id 621-2, 624-2
System Rollbacks 469-3

T

T3 protocol, configuring 495-8
tables 6-15
Targes 456-2
Target Access Point 617-4
Target Destination 280-3
Target Name 617-5
Target Qspace 617-4
Targeted Trap Destinations 69-5
Targeting Web Applications to the Virtual
Host. 540-5
Targets 452-1, 462-2
Targets-Server 94-1
Template 186-4, 211-4
Temporary Template 192-4
Test Created Connections 23-3, 113-8
Test Frequency 23-4, 113-7
Test Released Connections 23-4, 113-8
Test Reserved Connections 23-3, 113-7
Test Table Name 23-5, 113-12
Thread Count 489-3
Thread Priority 489-3
ThreadId log message attribute 253-4

Threads Increase 489-3
Threads Maximum 489-3
Threads Minimum 489-3
Threshold 500-4
Threshold High 503-3
Threshold Low 503-4
thresholds
 for CounterMonitor objects 516-21
Time To Deliver Override 187-3, 206-3,
213-3
Time To Live Override 187-2, 206-2, 213-2
Timeout 509-3, 612-2
Timeout Rollbacks 469-2
Timeout Seconds 67-2
Timestamp log message attribute 253-3
Total Committed 469-2
Total Connections 470-2
Total Current Entries 632-2, 634-10, 639-2
Total Garbage Collection Count 472-5
Total Heuristics 469-3
Total Invalid Logins 467-2
Total Login Attempts while Locked 467-2
Total Number Of Rejections 631-2, 636-18
Total Number Of Renewals 631-3, 636-19
Total Number Of Threads 472-8
Total Nursery Size 472-3
Total Persistent Current Entries 632-2,
634-10, 639-2
Total Rolled Back 469-2
Total Size Of Rejections 631-2, 636-18
Total Transactions 469-2
Total Transactions Abandoned 469-3
Total Transient Current Entries 632-3,
634-11, 639-3
Total Users Unlocked 467-2
Tp User File 596-7
TpUsr File Path 598-4
Transacted 199-3
Transaction
 Configuring 237-2
 Migrating the Transaction Recovery

Service 237-21, 237-22, 237-23
Transaction Log File Prefix 448-2
Transaction Mechanism 464-2
Transaction Recovery Service
 Constraining migration targets 237-22
 Migrating 237-21, 237-23
 viewing current owner 237-23
Transaction Timeout 156-2, 281-2
Transactional 614-2
TransactionId log message attribute 253-4
TransactionID log messages 253-6
transactions
 log message output 253-6
Transformer
 Configuring other than built-in 640-2
Transformer Factory 627-3
TranslateFML 617-3
Trusted Client Principals 352-4
Tuning JMS 235-2
 message paging
 attributes 235-14
 configuring 235-8
 overview 235-8
 persistent stores 235-2
 configuring synchronous write
 policies 235-2
Tunneling Client Ping 483-6, 484-3
Tunneling Client Timeout 483-7, 484-3
Tunneling Enabled 483-6
Tux To Jms Priority Map 615-3
Tuxedo Error Queue 614-3
Tuxedo Factory 613-2
Tuxedo GID Keyword 596-7
Tuxedo UID Keyword 596-7
Tx Data Source
 Deleting 141-8
Type 355-4

U

UnixMachine

Configuration

General

Enable Post-Bind GID
261-2

Enable Post-Bind UID
261-2

Name 261-2

Post-Bind GID 261-2

Post-Bind UID 261-2

Node Manager

Listen Address 262-2

Listen Port 262-2

Notes

Notes 267-1

UnixRealm

Configuration

Auth Program 318-2

Name 318-2

Realm Classname 318-2

Notes

Notes 319-2

Unlocked Users Total Count 405-6

Upload Directory Name 433-3

URI 571-2

URL 114-2, 139-2, 313-2

URL (Web) resources

 creating scoped roles 428-9

Use log filter 445-2

Use Navigation Tree 3-3

User Authentication 309-2

User Base DN 328-3, 378-3, 385-3, 391-3

User Cache Size 299-2

User Cache TTLNegative 299-2

User Cache TTLPositive 299-2

User DN 309-3

User Dynamic Group DNAttribute 328-2,
378-2, 385-2, 391-2

User From Name Filter 328-3, 378-3, 385-3,
391-3

User Id 614-5
 User Ids 77-4, 507-4
 User Lockout Total Count 405-5, 467-2
 User Name 139-2, 223-3, 243-1, 276-5,
 288-5, 313-2, 577-2
 User Name Attribute 309-3, 328-2, 378-2,
 385-2, 391-2
 User Name Mapper Class Name 352-4
 User Object Class 328-2, 378-2, 385-2, 391-2
 User Password 243-2, 276-5, 288-5, 577-2
 User Password Attribute 309-3
 User Role 243-2, 577-2
 User Search Scope 328-3, 378-3, 385-3,
 391-3
 UserId log message attribute 253-4
 Username 437-5
 username prompt
 hidden from standard out 253-24

V

Verbose 542-5, 573-4
 Version 325-2, 336-3, 338-3, 341-3, 345-3,
 349-3, 352-3, 355-2, 357-3, 374-2,
 380-2, 386-2, 406-2, 407-2, 409-3,
 410-2
 version 2-1
 ViewTbl classes 598-3
 ViewTbl32 classes 598-3
 Virtual Host Names 534-2
 VirtualHost
 Assign 274-3, 540-5
 Clone 540-2
 Configuration
 General
 Name 534-2
 Virtual Host Names 534-2
 HTTP
 Accept Context Path In
 Get Real Path

 535-3
 Default Server Name 535-2
 Https Keep Alive Secs
 535-3
 Keep Alive Enabled 535-3
 Keep Alive Secs 535-3
 Max Post Size 535-2
 Max Post Time Secs 535-2
 Post Timeout Secs 535-2
 WAPEnabled 535-2
 Logging
 Log File BufferK Bytes
 536-3
 Log File Flush Secs 536-5
 Log File Format 536-3
 Log File Name 536-2
 Log Rotation Period Mins
 536-4
 Log Rotation Time Begin
 536-5
 Log Rotation Type 536-4
 Logging Enabled 536-2
 Max Log File Size KBytes
 536-3

Configuring 540-1
 Delete 540-4
 Notes
 Notes 537-2
 Target Applications 540-5
 VirtualHosts
 associating 540-5
 removing 540-6

W

WAP Enabled 465-4
 WAPEnabled 535-2

WARNING severity level 253-5
 Web applications
 XML deployment descriptors 62-3
 WebAppComponent
 Configuration
 Descriptor
 Compile Command 542-4,
 573-4
 Enable JSP Line Numbers
 542-5, 573-5
 Enable Session Monitor-
 ing 542-4, 573-4
 Index Directories 542-4,
 573-3
 Keep Generated Source
 Files 542-4, 573-4
 Reload Period 542-3, 573-3,
 573-4
 Session Cookie Max Age
 Secs 542-2, 573-2
 Session Invalidation Inter-
 val Secs 542-3,
 573-2
 Session Timeout Secs
 542-3, 573-3
 Single Threaded Servlet
 Pool Size 542-3,
 573-3
 Verbose 542-5, 573-4
 Files
 Index Directories 545-2
 Reload Period 545-2
 General
 Deployment Order 571-3
 Load Order 571-2
 Name 543-2, 571-2

Staging Mode 543-3, 571-3
 URI 571-2
 Other
 Auth Realm Name 544-2
 Prefer Web Inf Classes
 544-2
 Single Threaded Servlet
 Pool Size 544-2
 Notes
 Notes 550-2, 565-2
 WebLogic 74-1
 WebLogic Plug-In Enabled 28-3, 434-7
 WebLogic Server
 starting 497-2
 version 2-1
 WebLogic Server, remote startup of 497-7
 WebLogic Tuxedo Connector
 Creating a Local TDM 619-3, 619-4,
 619-5, 619-6, 619-12
 Creating a Remote TDM 619-5
 Creating a WTCServer 619-1
 Creating Exported Services 619-7
 Creating Imxported Services 619-8
 Creating Passwords 619-9
 Creating Resources 619-10
 Creating tBridge Connections 619-11
 Creating tBridge Redirections 619-13
 Target a Server 619-2
 Weblogic Version 471-2
 weblogic.Server command
 directing standard out
 WebServiceComponent
 Configuration
 Files
 Index Directories 572-2
 Reload Period 572-2
 Other
 Auth Realm Name 570-2
 Prefer Web Inf Classes

570-2
Single Threaded Servlet
Pool Size 570-2
Weight 171-2, 178-2
When To Cache 621-3, 627-3
WLEC
 Create a WLEC Connection Pool 582-2
 Delete a Connection Pool 582-2
 Monitor Instances of a Connection Pool
 582-2
 Target a Connection Pool 582-3
WLECConnectionPool
 Configuration
 General
 Domain 576-3
 Failover Addresses 576-3
 Maximum Pool Size 576-4
 Minimum Pool Size 576-3
 Name 576-2
 Primary Addresses 576-2
 Security
 Application Password
 577-3
 Enable Certificate Au-
 thentication 577-5
 Enable Security Context
 577-5
 Maximum Encryption
 Level 577-4
 Minimum Encryption
 Level 577-3
 User Name 577-2
 User Password 577-2
 User Role 577-2
 Notes
 Notes 578-1
WLS Error Destination 614-2

WTCEExport
 General
 EJB Name 584-2
 Local Access Point 584-2
 Remote Name 584-3
 Resource Name 584-2
WTCImport
 General
 Local Access Point 586-2
 Remote Access Point List 586-2
 Remote Name 586-2
 Resource Name 586-2
WTCLocalTuxDom
 Connections
 Blocking Time Out 588-5
 Compression Limit 588-5
 Connection Policy 588-3
 Connection Principal Name 588-4
 Interoperate 588-5
 Max Retries 588-4
 Retry Interval 588-4
 General
 Access Point 589-2
 Access Point Id 589-2
 Network Address 589-3
 Security
 Max Encryption Level 590-3
 Min Encryption Level 590-3
 Security 590-2
WTCPassword
 Configuration
 Local Access Point 592-2
 Local Password 592-2
 Local Password IV 592-2
 Remote Access Point 592-2
 Remote Password 592-3
 Remote Password IV 592-3
WTCRemoteTuxDom
 Connections
 Cmp Limit 594-5
 Connection Policy 594-3

Connection Principal Name 594-4	Retry Delay 612-2
Max Retries 594-5	Timeout 612-2
Retry Interval 594-4	Factories
General	JMS Factory 613-2
Access Point 595-2	JNDI Factory 613-2
Access Point Id 595-2	Tuxedo Factory 613-2
Federation Name 595-4	General
Federation URL 595-3	Allow Non Standard Types 614-5
Local Access Point 595-2	Default Reply Delivery Mode 614-4
Network Address 595-3	Delivery Mode Override 614-3
Security	Transactional 614-2
Acl Policy 596-2	Tuxedo Error Queue 614-3
Allow Anonymous 596-4	User Id 614-5
AppKey Generator 596-6	WLS Error Destination 614-2
Credential Policy 596-3	Priority Mapping
Custom AppKey Class 596-8	Jms To Tux Priority Map 615-2
Custom AppKey Param 596-8	Tux To Jms Priority Map 615-3
Default AppKey 596-5	WTCtBridgeRedirect
Max Encryption Level 596-4	General
Min Encryption Level 596-3	Direction 617-2
Tp User File 596-7	Meta Data File 617-3
Tuxedo GID Keyword 596-7	Reply Q 617-3
Tuxedo UID Keyword 596-7	Source Access Point 617-4
WTCResources	Source Name 617-4
Configuration	Source Qspace 617-4
App Password 598-3	Target Access Point 617-4
App Password IV 598-4	Target Name 617-5
FldTbl classes 598-2	Target Qspace 617-4
FldTbl32 classes 598-2	TranslateFML 617-3
TpUsr File Path 598-4	
ViewTbl classes 598-3	
ViewTbl32 classes 598-3	
WTCServer	
General	
Deployment Order 600-2	
Name 600-2	
Notes	
Notes 609-3	
WTCtBridgeGlobal	
Connections	
Retries 612-2	

X

XA Connection Factory Enabled 156-3
XML External Entity Cache
Configuring 640-7
Monitoring 640-8
XML External Entity Resolution
Configuring 640-5
XML Parser
Configuring for a document type 640-3
Configuring other than the built-in 640-2

XML Registry	Max Entry Memory Size
Targets	638-2
629-2	Min Entry Memory Size
XMLEntityCache	638-2
Current	General
Entry Resource Usage	Avg Percent Persistent
Avg Per Entry Disk Size	639-2
633-7	Avg Percent Transient
Avg Per Entry Memory	639-3
Size 633-6	Avg Timeout 639-4
Max Entry Memory Size	Max Entry Timeout 639-3
633-6	Min Entry Timeout 639-3
Min Entry Memory Size	Total Current Entries 639-2
633-6	Total Persistent Current
General	Entries 639-2
Avg Percent Persistent	Total Transient Current
632-2	Entries 639-3
Avg Percent Transient	Rejections
632-3	Percent Rejected 636-18
Avg Timeout 632-4	Total Number Of Rejec-
Max Entry Timeout 632-3	tions 636-18
Min Entry Timeout 632-3	Total Number Of Renew-
Total Current Entries 632-2	als 636-19
Total Persistent Current	Total Size Of Rejections
Entries 632-2	636-18
Total Transient Current	Session
Entries 632-3	Entry Resource Usage
Total Resource Usage	Avg Per Entry Disk Size
Disk Usage 637-2	635-15
Memory Usage 637-2	Avg Per Entry Memory
Historical	Size 635-14
Entry Resource Usage	Max Entry Memory Size
Avg Per Entry Disk Size	635-14
638-3	Min Entry Memory Size
Avg Per Entry Memory	635-14
Size 638-2	

General

Avg Percent Persistent
634-10

Avg Percent Transient
634-11

Avg Timeout 634-12

Max Entry Timeout 634-11

Min Entry Timeout 634-11

Total Current Entries
634-10

Total Persistent Current
Entries 634-10

Total Transient Current
Entries 634-11

Rejections

Percent Rejected 631-2

Total Number Of Rejec-
tions 631-2

Total Number Of Renew-
als 631-3

Total Size Of Rejections
631-2

XMLEntitySpecRegistryEntry

Configuration

Cache Timeout Interval 621-3

EntityURI 621-2

Public Id 621-2

System Id 621-2

When To Cache 621-3

Notes

Notes 622-2

XMLParserSelectRegistryEntry

Configuration

Document Builder Factory 624-2

Parser Class Name 624-3

Public Id 624-2

Root Element Tag 624-2

SAXParser Factory 624-3

System Id 624-2

Notes

Notes 625-2

XMLRegistry 458-2

Configuration

DocumentBuilderFactory 627-2

Name 627-2

SAXParserFactory 627-2

Transformer Factory 627-3

When To Cache 627-3

Notes

Notes 628-2

XSLT Transformer

Configuring other than built-in 640-2