



# **BEA AquaLogic Service Bus™**

## **Deployment Guide**

# Copyright

Copyright © 2005 BEA Systems, Inc. All Rights Reserved.

## Restricted Rights Legend

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

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

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

## Trademarks or Service Marks

BEA, BEA JRockit, BEA Liquid Data for WebLogic, BEA WebLogic Server, Built on BEA, Jolt, JoltBeans, SteelThread, Top End, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA AquaLogic, BEA AquaLogic Data Services Platform, BEA AquaLogic Enterprise Security, BEA AquaLogic Service Bus, BEA AquaLogic Service Registry, BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Manager, BEA MessageQ, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Enterprise Security, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Java Adapter for Mainframe, BEA WebLogic JDriver, BEA WebLogic JRockit, BEA WebLogic Log Central, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server Process Edition, BEA WebLogic WorkGroup Edition, BEA WebLogic Workshop, and Liquid Computing are trademarks of BEA Systems, Inc. BEA Mission Critical Support is a service mark of BEA Systems, Inc. All other company and product names may be the subject of intellectual property rights reserved by third parties.

All other trademarks are the property of their respective companies.

# Contents

## 1. Introduction

Deployment Goals . . . . .	1-1
Key Deployment Tasks . . . . .	1-2
Roles in AquaLogic Service Bus Deployment . . . . .	1-2
Deployment Specialists. . . . .	1-3
WebLogic Server Administrators . . . . .	1-3
Database Administrators. . . . .	1-3
Key Deployment Resources. . . . .	1-4
WebLogic Server Resources. . . . .	1-4
Clustering. . . . .	1-5
Java Message Service . . . . .	1-5
EJB Pooling and Caching . . . . .	1-5
JDBC Connection Pools . . . . .	1-6
Execution Thread Pool . . . . .	1-6
J2EE Connector Architecture . . . . .	1-7
AquaLogic Service Bus Configuration Resources . . . . .	1-7
WSDLs . . . . .	1-7
Business Services. . . . .	1-8
Proxy Services . . . . .	1-8
Schemas . . . . .	1-9
Service Accounts . . . . .	1-9
Proxy Service Providers . . . . .	1-9

WS-Policies .....	1-9
XQuery and XSLT Transformations.....	1-10
MFLs.....	1-10
Relational Database Management System Resources .....	1-10
Hardware, Operating System, and Network Resources.....	1-10

## 2. Configuring a Single-Server Deployment

Step 1. Configure a Database for the JMS Reporting Provider Data Store .....	2-1
Step 2. Prepare an AquaLogic Service Bus Domain.....	2-2
Creating an AquaLogic Service Bus Domain Using the Configuration Wizard ....	2-2
Configuring JMS Resources .....	2-5
Step 3. Configure AquaLogic Service Bus Security.....	2-6
Step 4. Deploy an AquaLogic Service Bus Configuration .....	2-6
Step 5. Update Your Domain as Your Production Environment Changes .....	2-7
Changing a Business Service.....	2-7
Installing a New Version of a Proxy Service .....	2-8
Online Configuration Updates .....	2-8
Best Practices for Successful Online Configuration Updates.....	2-10

## 3. Understanding AquaLogic Service Bus Clusters

Understanding AquaLogic Service Bus Clusters .....	3-1
Designing a Clustered Deployment.....	3-2
Introducing AquaLogic Service Bus Domains .....	3-2
Creating Domains .....	3-2
Clustered Servers .....	3-3
AquaLogic Service Bus Deployment Resources .....	3-3
Singleton Resources .....	3-3
Monitoring and Alert Resources in a Cluster .....	3-4

Cluster Configuration Changes and Deployment Requests . . . . .	3-5
Load Balancing in a AquaLogic Service Bus Cluster . . . . .	3-5
Load Balancing HTTP Functions in a Cluster . . . . .	3-6
Load Balancing JMS Functions in a Cluster . . . . .	3-6
High Availability in a AquaLogic Service Bus Cluster. . . . .	3-6
Highly Available JMS for AquaLogic Service Bus . . . . .	3-6
Deploying Configurations . . . . .	3-7

## 4. Configuring a Clustered Deployment

Step 1. Comply with Configuration Prerequisites . . . . .	4-2
Step 2. Prepare a AquaLogic Service Bus Domain . . . . .	4-5
Creating an AquaLogic Service Bus Domain Using the Configuration Wizard . . . .	4-5
Adding Proxy Server or Firewall Information to your Domain Configuration .	4-8
Configuring JMS Resources. . . . .	4-9
Step 3. Configure WebLogic Integration Security . . . . .	4-9
Step 4. Starting, Stopping, and Monitoring Managed Servers . . . . .	4-10
Starting and Stopping Managed Servers . . . . .	4-10
Monitoring Your Servers . . . . .	4-10
Step 5. Deploy an AquaLogic Service Bus Configuration . . . . .	4-11
Step 6. Update Your Domain as Your Production Environment Changes. . . . .	4-11
Adding a Managed Server . . . . .	4-11
Adding A Managed Server to an AquaLogic Service Bus Cluster . . . . .	4-12
Updating Business Service Configurations for an Expanded Cluster. . . . .	4-14
Updating Proxy Service Configurations for an Expanded Cluster . . . . .	4-14
Dropping a Managed Server . . . . .	4-15
Changing a Business Service in a Cluster . . . . .	4-15
Installing a New Version of a Proxy Service in a Cluster . . . . .	4-16

5. Understanding AquaLogic Service Bus High Availability

- About AquaLogic Service Bus High Availability. . . . . 5-1
  - Recommended Hardware and Software. . . . . 5-1
    - Regarding JMS File Stores . . . . . 5-3
  - What Happens When a Server Fails . . . . . 5-3
    - Software Faults . . . . . 5-4
    - Hardware Faults . . . . . 5-4
    - Server Migration . . . . . 5-4
    - Message Reporting Purger . . . . . 5-5
- AquaLogic Service Bus Failure and Recovery . . . . . 5-5
  - Transparent Server Reconnection . . . . . 5-5
  - EIS Instance Failover . . . . . 5-6

A. Using the AquaLogic Service Bus Deployment API

- Importing and Exporting Configurations . . . . . A-1
- Updating Environment-Specific Information . . . . . A-4

B. AquaLogic Service Bus Deployment Resources

Index

# Introduction

This document describes how to deploy BEA AquaLogic Service Bus configurations in a production environment. The following sections introduce key concepts and tasks for deploying AquaLogic Service Bus in your organization:

- [Deployment Goals](#)
- [Key Deployment Tasks](#)
- [Roles in AquaLogic Service Bus Deployment](#)
- [Key Deployment Resources](#)

This document focuses on the deployment phase of the AquaLogic Service Bus software lifecycle. For a general overview of AquaLogic Service Bus, see *[BEA AquaLogic Service Bus Concepts and Architecture](#)*.

For information about configuring AquaLogic Service Bus, see the documentation available at the following URL:

<http://edocs.bea.com/alsb/docs20/index.html>

## Deployment Goals

AquaLogic Service Bus combines intelligent message brokering with service monitoring and administration to provide a unified software product for implementing and deploying your Service-Oriented Architecture (SOA). When deploying AquaLogic Service Bus configurations, consider the following goals:

- *High Availability.* A deployment must be sufficiently available and accessible, with provisions for failover in the event of hardware or network failures.
- *Performance.* A deployment must deliver sufficient performance at peak and off-peak loads.
- *Scalability.* A deployment must be capable of handling anticipated increases in loads simply by using additional hardware resources, rather than requiring code changes.
- *Security.* A deployment must sufficiently protect data from unauthorized access or tampering.

You can achieve these goals and others with every AquaLogic Service Bus configuration.

## Key Deployment Tasks

Deploying AquaLogic Service Bus may require that you complete some or all of the following tasks:

1. Define the goals for your AquaLogic Service Bus deployment, as described in [“Deployment Goals” on page 1-1](#).
2. Deploy your AquaLogic Service Bus configuration in a cluster. To do so, you must first design the cluster, and before you can start designing, you need to understand the components of a AquaLogic Service Bus deployment. [Chapter 3, “Understanding AquaLogic Service Bus Clusters”](#) provides descriptions of these components that will help you design the best possible environment for your configuration. For the procedure to deploy a highly available AquaLogic Service Bus configuration, see [Chapter 4, “Configuring a Clustered Deployment.”](#)
3. Set up security for your AquaLogic Service Bus deployment as described in [Securing Inbound and Outbound Messages](#) in the *BEA AquaLogic Service Bus User Guide*.

## Roles in AquaLogic Service Bus Deployment

To deploy an integrated solution successfully, a deployment team must include people who perform the following roles:

- [Deployment Specialists](#)
- [WebLogic Server Administrators](#)
- [Database Administrators](#)



One person can assume multiple roles, and all roles are not equally relevant in all deployment scenarios, but a successful deployment requires input by people in each role.

## Deployment Specialists

Deployment specialists coordinate the deployment effort. They are knowledgeable about the features of the AquaLogic Service Bus product. They provide expertise in designing the deployment topology for an ESB solution, based on their knowledge of how to configure various AquaLogic Service Bus features on one or more servers. Deployment specialists have experience in the following areas:

- Resource requirements analysis
- Deployment topology design
- Project management

## WebLogic Server Administrators

WebLogic Server administrators provide in-depth technical and operational knowledge about WebLogic Server deployments in an organization. They have knowledge of the hardware and platform, and experience managing all aspects of a WebLogic Server deployment, including installation, configuration, monitoring, security, performance tuning, troubleshooting, and other administrative tasks.

## Database Administrators

Database administrators provide in-depth technical and operational knowledge about database systems deployed in an organization. They have experience in the following areas:

- Hardware and platform knowledge
- Expertise in managing all aspects of a relational database (RDBMS), including installation, configuration, monitoring, security, performance tuning, troubleshooting, and other administrative tasks

## Key Deployment Resources

This section provides an overview of resources that can be modified at deployment time:

- [WebLogic Server Resources](#)
- [AquaLogic Service Bus Configuration Resources](#)
- [Relational Database Management System Resources](#)
- [Hardware, Operating System, and Network Resources](#)

**Note:** The term *resource* is used in this document to refer to technical assets in general, except in discussions of security where it is used to refer only to those underlying WebLogic Server entities that can be protected from unauthorized access using security roles and security policies.

## WebLogic Server Resources

This section provides general information about WebLogic Server resources that are most relevant to the deployment of a AquaLogic Service Bus solution. You can configure these resources from the WebLogic Server Administration Console or through J2EE and WebLogic resource descriptors.

WebLogic Server provides many configuration options and tunable settings for deploying AquaLogic Service Bus solutions in any supported environment. The following sections describe the configurable WebLogic Server features that are most relevant to AquaLogic Service Bus deployments:

- [Clustering](#)
- [Java Message Service](#)
- [EJB Pooling and Caching](#)
- [JDBC Connection Pools](#)
- [Execution Thread Pool](#)
- [J2EE Connector Architecture](#)

## Clustering

To increase workload capacity, you can run WebLogic Server on a cluster: a group of servers that can be managed as a single unit. Clustering provides a deployment platform that is more scalable than a single server. For more information about clustering, see [Chapter 3, “Understanding AquaLogic Service Bus Clusters.”](#)

## Java Message Service

The WebLogic Java Message Service (JMS) enables Java applications sharing a messaging system to exchange (create, send, and receive) messages. WebLogic JMS is based on the [Java Message Service Specification](#) version 1.0.2 from Sun Microsystems, Inc.

JMS servers can be clustered and connection factories can be deployed on multiple instances of WebLogic Server. For more information about WebLogic JMS, see the following topics:

- [Understanding WebLogic JMS](#) in *Programming WebLogic JMS*
- [Configuring Clustered WebLogic JMS Resources](#) and [Monitoring JMS Statistics and Managing Messages](#) in the *Configuring and Managing WebLogic JMS*.

## EJB Pooling and Caching

In an AquaLogic Service Bus deployment, the number of EJBs affects system throughput. You can tune the number of EJBs in the system through either the EJB pool or the EJB cache, depending on the type of EJB. The following table describes types of EJBs and their associated tunable parameter.

**Table 1-1 Parameters for Tuning EJBs**

EJB Type	Tunable Parameter Name	Tunable Parameter Description
Message-Driven Beans	max-beans-in-free-pool	The maximum number of listeners that pull work from a queue.
Stateless Session Beans	max-beans-in-free-pool	The maximum number of beans available for work requests.
Stateful Session Beans	max-beans-in-cache	The number of beans that can be active at once. A setting that is too low results in <code>CacheFullExceptions</code> . A setting that is too high results in excessive memory consumption.
Entity Beans		

For more information about controlling throughput, see “Server Self-Tuning for Production Environments” in [New and Changed Features in WebLogic Server Environments](#) in *Designing and Configuring WebLogic Server Environments*.

## JDBC Connection Pools

Java Database Connectivity (JDBC) enables Java applications to access data stored in SQL databases. To reduce the overhead associated with establishing database connections, WebLogic JDBC provides connection pools that offer ready-to-use pools of connections to a DBMS.

JDBC connection pools are used to optimize DBMS connections. If you are using the AquaLogic Service Bus JMS Reporting Provider, you can tune AquaLogic Service Bus performance by configuring the size of JDBC connection pools. A setting that is too low results in delays while AquaLogic Service Bus waits for connections to become available. A setting that is too high results in slower DBMS performance.

For more information about WebLogic JDBC connection pools, see:

- “How Connection Pools Enhance Performance” in [Performance Tuning Your JDBC Application](#) in *Programming WebLogic JDBC*
- “Connection Pool Features” in [Configuring JDBC Data Sources](#) in *Configuring and Managing WebLogic JDBC*.

## Execution Thread Pool

The *execution thread pool* controls the number of threads that can execute concurrently on WebLogic Server. A setting that is too low results in sequential processing and possible deadlocks. A setting that is too high results in excessive memory consumption and may cause thrashing.

The number of execute threads also determines the number of threads that read incoming socket messages (socket-reader threads). This number is, by default, one-third of the number of execute threads. A number that is too low can result in contention for threads for reading sockets and can sometimes lead to a deadlock.

Set the execution thread pool high enough so that all candidate threads run, but not so high that performance is hampered due to excessive context switching in the system. Monitor your running system to determine empirically the best value for the execution thread pool.

**Note:** Most production applications require an execution thread count greater than the default value. A thread count of 50 is a commonly used value. Be sure to adjust your JDBC connection pool to match your thread count value.

For more information about controlling throughput, see “Server Self-Tuning for Production Environments” in [New and Changed Features in WebLogic Server Environments](#) in *Designing and Configuring WebLogic Server Environments*.

## J2EE Connector Architecture

The WebLogic J2EE Connector Architecture (JCA) integrates the J2EE Platform with one or more heterogeneous Enterprise Information Systems (EIS). The WebLogic JCA is based on the *J2EE Connector Specification*, Version 1.0, from Sun Microsystems, Inc.

For information about the WebLogic J2EE-CA, see [J2EE Connector Architecture](#) in *Programming WebLogic Resource Adapters*.

## AquaLogic Service Bus Configuration Resources

AquaLogic Service Bus configuration resources contain environment-specific settings that you will want to change or tune when deploying the configuration to a new domain. The following sections describe the resources that you may need to reconfigure after deploying a configuration.

### WSDLs

AquaLogic Service Bus uses WSDL (Web Service Definition Language) to describe proxy services and business services. WSDL is used to describe what a Web service can do, where it resides, and how to invoke it.

You can base the definition of proxy services and business services on existing WSDL files, and complete the configuration of the services using the AquaLogic Service Bus Console. WSDL files used as the basis for the definition of services are stored as AquaLogic Service Bus resources. These resources are unlikely to require update when deployed to a new environment, because AquaLogic Service Bus does not use the URLs in these WSDL files at run time.

**Note:** AquaLogic Service Bus creates a new WSDL file for each HTTP proxy service. You can view the contents of this WSDL file by appending `?wsdl` to the endpoint for the service. For example, when running the AquaLogic Service Bus Examples Server (**Start→All Programs→BEA Products→Examples→AquaLogic Service Bus→Start Examples Server**), you can view the WSDL for the `loadgateway2` proxy service at `http://localhost:7001/crejws_basic_ejb/loadgateway2?wsdl`.

## Business Services

Business services are AquaLogic Service Bus definitions of the enterprise information services with which you want to exchange messages. Business services in a production environment could specify multiple endpoints (URLs) for load balancing purposes and high availability. For information on how to add endpoints to a business service, see “Viewing and Changing Business Services” in [Business Services](#) in the *Using the AquaLogic Service Bus Console*. For information on how to update the value of existing endpoints, see “Finding and Replacing Environment Values” in [Using the Change Center](#) in *Using the AquaLogic Service Bus Console*.

## Proxy Services

Proxy services are AquaLogic Service Bus definitions of intermediary Web services that AquaLogic Service Bus implements locally on WebLogic Server. While the majority of the metadata that defines a proxy service can be deployed without change in a new environment, there is some information you may need to update:

- Proxy service message flows route messages to named destinations (business services, other proxy services, and so on). Message routing definitions may need to be updated in a new environment. For information on how to configure this resource appropriately for your environment, see “Viewing and Changing Message Flow” in [Proxy Services](#) in *Using the AquaLogic Service Bus Console*.
- Definitions of proxy services for File, FTP, and Email message types must specify a single managed server for deployment of polling runtime components in a cluster. The Managed Server drop-down list appears in the AquaLogic Service Bus Console in clustered AquaLogic Service Bus domains only. For information on how to edit the Managed Server value for a proxy service, see “Viewing and Changing Proxy Services” in [Proxy Services](#) in *Using the AquaLogic Service Bus Console*.
- Proxy service definitions can include directory names that may need to be updated for a new environment. For information on how to configure this resource appropriately for your environment, see “Finding and Replacing Environment Values” in [Using the Change Center](#) in *Using the AquaLogic Service Bus Console*.
- Proxy service definitions include references to other AquaLogic Service Bus resources. It is important to verify the validity of these references in a new environment.
- JMS queues and connection factories in the proxy service URL may need to be updated. For more information, see “JMS Queue and Topic Destination Tasks” and “JMS Connection Factory Tasks” in [Configuring JMS System Resources](#) in *Configuring and Managing WebLogic JMS*.

- Each proxy service relies on an instance of WebLogic Server Work Manager for its dispatch policy. You can tune the Work Manager instance to meet the requirements of your production environment. For more information, see [Using Work Manager to Optimize Scheduled Work](#) in *Designing and Configuring WebLogic Server Environments*.

For more information about proxy services, see [Proxy Services](#) in the *Using the AquaLogic Service Bus Console*.

## Schemas

A schema is a document that defines valid content for an XML document. Schemas are used to add XML information to messages exchanged in AquaLogic Service Bus. These resources are unlikely to require update when deployed to a new environment.

## Service Accounts

AquaLogic Service Bus uses service accounts to provide authentication when connecting to a service or server. For information about using this resource appropriately in your production environment, see “Transport Security for JMS, Email, FTP, and Files” in [Securing Inbound and Outbound Messages](#) in the *BEA AquaLogic Service Bus User Guide*.

## Proxy Service Providers

AquaLogic Service Bus uses proxy service providers to supply credential-level validation to proxy services. The following types of security are available:

- SSL client authentication
- Digital signature
- Encryption
- Web services security X509 token

For information about how to configure this resource appropriately for your environment, see [Proxy Service Providers](#) in the *Using the AquaLogic Service Bus Console*.

## WS-Policies

AquaLogic Service Bus uses Web Service Policies (WS-Policies) to associate Web service security policy with proxy services and business services. For information about how to configure this resource appropriately for your production environment, see “Web Service Policy” in [Securing Inbound and Outbound Messages](#) in the *BEA AquaLogic Service Bus User Guide*.

These resources are unlikely to require update when deployed to a new environment.

### **XQuery and XSLT Transformations**

Transformation maps describe the mapping between two data types. AquaLogic Service Bus supports data mapping using either XQuery or the eXtensible Stylesheet Language Transformation (XSLT) standard. These resources are unlikely to require update when deployed to a new environment.

### **MFLs**

Message Format Language (MFL) is a BEA proprietary language used to define rules to transform formatted binary data into XML data. MFL documents are unlikely to require update when deployed to a new environment.

## **Relational Database Management System Resources**

AquaLogic Service Bus relies on database resources for storing message reporting data by the JMS Reporting Provider. Database performance is a factor in overall AquaLogic Service Bus performance. For information about database tuning requirements associated with AquaLogic Service Bus applications, see the [BEA AquaLogic Service Bus Release Notes](#).

For additional information on turning your database, see your database vendor's documentation.

## **Hardware, Operating System, and Network Resources**

Hardware, operating system, and network resources play a crucial role in AquaLogic Service Bus performance. Deployments must comply with the hardware and software requirements described in the [BEA AquaLogic Service Bus Release Notes](#).



# Configuring a Single-Server Deployment

This section describes the tasks that you must perform to configure AquaLogic Service Bus for deployment in a single WebLogic Server environment.

To set up and deploy AquaLogic Service Bus in a single-server configuration, complete the following steps:

- [Step 1. Configure a Database for the JMS Reporting Provider Data Store](#)
- [Step 2. Prepare an AquaLogic Service Bus Domain](#)
- [Step 3. Configure AquaLogic Service Bus Security](#)
- [Step 4. Deploy an AquaLogic Service Bus Configuration](#)
- [Step 5. Update Your Domain as Your Production Environment Changes](#)

## Step 1. Configure a Database for the JMS Reporting Provider Data Store

AquaLogic Service Bus requires a database for the JMS Reporting Provider. The local copy of the PointBase database that is installed with WebLogic Server is for evaluation purposes only. Non-evaluation development or other use of the PointBase Server requires that a separate PointBase license be obtained by the end user directly from DataMirror.

For a complete list of the databases that you can use, see [Supported Database Configurations](#) in *Supported Configurations for AquaLogic Service Bus*.

**Note:** It is important to configure your database appropriately for production use. You must provide adequate space to log messages, and follow best practices for administering your database.

For the latest information about specific databases, see the [BEA AquaLogic Service Bus Release Notes](#).

## Step 2. Prepare an AquaLogic Service Bus Domain

To prepare a AquaLogic Service Bus environment, complete the tasks described in the following sections:

- [Creating an AquaLogic Service Bus Domain Using the Configuration Wizard](#)
- [Configuring JMS Resources](#)

### Creating an AquaLogic Service Bus Domain Using the Configuration Wizard

You begin the definition of an AquaLogic Service Bus deployment by creating a domain using the BEA Configuration Wizard.

**Note:** The procedure described in this section for setting up your domain is based on the assumption that you are running the Configuration Wizard in GUI mode from the Windows Start menu. For information about using the Configuration Wizard in different modes, see [Creating WebLogic Domains Using the Configuration Wizard](#).

To create an AquaLogic Service Bus domain using the Configuration Wizard, complete the following steps:

1. From the Start Menu, choose **All Programs→BEA Products→Tools→Configuration Wizard**.

The Configuration Wizard is launched. It prompts you for data with which to configure your domain.

2. Respond to the Configuration Wizard prompts by providing the information described in the following table.

**Note:** To comply with WebLogic Server resource naming rules, you must specify unique names for domains, WebLogic Server instances, JMS servers, and JMS stores. AquaLogic Service Bus has the same interoperability naming requirements as the WebLogic Messaging Bridge. For more information, see “Naming Guidelines for WebLogic Servers and Domains” in “Interoperating with Different WebLogic Server

Releases” in [Interoperating with Different WebLogic Server Releases or Foreign Providers](#) in *Configuring and Managing the WebLogic Messaging Bridge*.

**Table 2-1 Responses to Configuration Wizard Prompts**

In this window . . .	Perform the following action . . .
Welcome	Select Create a new WebLogic domain.
Select Domain Source	Select AquaLogic Service Bus.
Configure Administrator Username and Password	Enter user name and password.
Configure Server Start Mode and JDK	Select Production Mode. Select either the Sun SDK or JRockit SDK, or specify the location of another JDK.
Customize Environment and Services Settings	Select Yes.
Configure the Administration Server	If you want to enable SSL for your configuration, select the SSL enabled check box.
Configure Managed Servers	Because this is a single-server deployment, you do not add any managed servers.
Configure Machines	Because this is a single-server deployment, you do not add any machines.

## Configuring a Single-Server Deployment

Configure JDBC Data Sources	<p>Accept the default values for Name and JNDI name of <code>wlsbjmsrpDataSource</code>.</p> <p>Select one of the following to identify the database type and driver for the JMS Reporting Provider Data Store:</p> <ul style="list-style-type: none"><li>• Oracle and BEA's Oracle Driver (Type 4) Versions: 9.0.1, 9.2.0, 10</li><li>• MS SQL Server and BEA's MS SQL Server Driver (Type 4) Versions: 7.0, 2000</li></ul> <p>Confirm that the Supports global transactions check box and Logging last resource option are selected. For more information about Logging last resource (LLR), see "Understanding the Logging Last Resource Transaction Option" in <a href="#">Configuring JDBC Data Sources</a> in <i>Configuring and Managing WebLogic JDBC</i>.</p> <p>Enter your environment-specific database information in the remaining text boxes. For more information, see "Configure JDBC Data Sources" in <a href="#">Customizing Existing JDBC and JMS Settings</a> in <i>Creating WebLogic Domains Using the Configuration Wizard</i>.</p> <p>Click Test Connections to verify that you can contact the database you want to use for the JMS Reporting Provider Data Store using this data source configuration.</p>
Run Database Scripts	<p>Select <code>wlsbjmsrpDataSource</code> from the Available JDBC Data Sources list.</p> <p>Select the version of the database from the DB Version drop-down list.</p> <p>Click Run Scripts.</p> <p>The scripts create the tables and indexes for the JMS Reporting Provider Data Store. The SQL output is displayed in the Results box and written to <code>jdbc.log</code>. If you want the results written to a different file, click the Log File check box and specify the file.</p> <p><b>Note:</b> AquaLogic Service Bus does not automatically run database scripts for the JMS Reporting Provider Data Store the first time you start a production domain, as it does for domains in development mode. If you do not run the database scripts while creating your AquaLogic Service Bus production domain, you must run the scripts manually. The scripts are located in <code>BEA_HOME/weblogic90/integration/common/dbscripts</code>, where <code>BEA_HOME</code> is the directory in which you installed AquaLogic Service Bus.</p>
Configure JMS File Stores	<p>Accept the defaults for <code>rmfilestore</code> and add any file stores needed by proxy services or business services.</p>
Review WebLogic Domain	<p>Accept the default values.</p>

Create WebLogic Configuration	Enter a name and location for your domain.
Creating Domain	After your domain is created, select Start Admin Server to start AquaLogic Service Bus when you exit the Configuration Wizard.

When you complete the domain configuration using the Configuration Wizard, your new domain is created in the location you specified.

Your AquaLogic Service Bus domain includes a configuration file (`config.xml`) that contains a definition for the administration server. For more information, see “`config.xml`” in [Domain Configuration Files](#) in *Understanding Domain Configuration*.

For information about configuring domains without using the Configuration Wizard, see [Understanding Domain Configuration](#).

## Configuring JMS Resources

In addition to configuring JMS file stores in the Configuration Wizard, proxy services and business services that use JMS require configuration of the following resources:

- JMS connection factories. You must configure XA or non-XA JMS connection factories for all business services and proxy services implemented using JMS.
- JMS queues/topics. AquaLogic Service Bus automatically configures JMS queues for proxy services that are implemented using BEA JMS. You must configure JMS queues/topics for all business services using JMS and for any proxy services that are implemented using non-BEA JMS.

If you want to concentrate all AquaLogic Service Bus JMS resources in a single JMS module, use the WebLogic Server Administration Console to create a new JMS module containing the destination to be used for the proxy services’ endpoint.

For more information about configuring JMS resources, see [Configuring and Managing WebLogic JMS](#).

## Step 3. Configure AquaLogic Service Bus Security

AquaLogic Service Bus leverages the security features of WebLogic Server to ensure message confidentiality and integrity (message-level security), secure connections between clients and WebLogic Server (transport-level security), and authentication and authorization (access control). For information on how to configure security for AquaLogic Service Bus, see [Securing Inbound and Outbound Messages](#) in the *BEA AquaLogic Service Bus User Guide*.

**Warning:** You must configure security separately for each AquaLogic Service Bus domain. AquaLogic Service Bus does not export or import security configurations.

## Step 4. Deploy an AquaLogic Service Bus Configuration

Once you have configured your AquaLogic Service Bus domain, secured it, and added any JMS resources required for its services, you are ready to import the JAR file that contains your AquaLogic Service Bus configuration. After you have imported the configuration metadata, you can update environment-specific information for your domain.

The following steps describe the basic procedure for deploying the contents of configuration JAR file:

1. Create a Session.
2. Import all or selected objects from a configuration JAR file.
3. Update environment-specific information such as service endpoint URIs and directory names.
4. Activate the Session.

You can perform these steps manually or programmatically:

- To import and update a configuration manually, use the AquaLogic Service Bus Console as described in the following topics in the *Using the AquaLogic Service Bus Console*:
  - “Importing Configuration Data” in [System Administration](#)
  - “Finding and Replacing Environment Values” in [Using the Change Center](#)
- To import and update a configuration programmatically, use the WebLogic Scripting Tool (WLST) and the AquaLogic Service Bus `deploymentMBean` as described in [Appendix A](#), “Using the AquaLogic Service Bus Deployment API.”

In addition to service endpoint URIs, directory names, and security configuration, your AquaLogic Service Bus configuration may contain other settings that must be updated to operate correctly in the new environment. Items that commonly require update include the following:

- Service references

For information about service references, see “Viewing References” in [Project Explorer](#) in the *Using the AquaLogic Service Bus Console*.

- Routing destinations

For information about routing configuration, see “Viewing and Changing Message Flow” in [Project Explorer](#) in the *Using the AquaLogic Service Bus Console*.

- Load balancing settings

For information about load balancing, see “Viewing and Changing Business Services” in [Business Services](#) in the *Using the AquaLogic Service Bus Console*.

Use the AquaLogic Service Bus Console to confirm and change your configuration, as necessary.

## Step 5. Update Your Domain as Your Production Environment Changes

Production environments change over time and as application use increases. This section describes how to update your domain in response to common production environment change scenarios:

- [Changing a Business Service](#)
- [Installing a New Version of a Proxy Service](#)
- [Online Configuration Updates](#)

### Changing a Business Service

Enterprise information services (EIS) are sometimes phased out, and new instances (possibly with new versions of EIS software, new hardware, and so on) are brought online. When this happens, AquaLogic Service Bus administrators need to gracefully transition to the new EIS instance by modifying any affected AquaLogic Service Bus business services.

This situation is similar to an EIS instance failure, but not as urgent. For a description of deployment considerations, see “[EIS Instance Failover](#)” on [page 5-6](#). For information about using the AquaLogic Service Bus Console to change an endpoint URI for a business service, see

“Viewing and Changing Business Services” in [Business Services](#) in *Using the AquaLogic Service Bus Console*.

## Installing a New Version of a Proxy Service

As your business requirements change, you may need to make changes to your proxy services. If the changes you need to make are backward compatible, you can make changes dynamically online using the AquaLogic Service Bus Console to create a new version of the proxy service. Changes are backward compatible if they meet one of the following criteria:

- The interface of the changed object is unchanged.
- Old and new clients will work with the interface.

If the changes you need to make are not backward compatible, there are two alternatives to consider that would enable you to make the changes online:

- Create and deploy a new proxy service having a different name and URL from that of the earlier version. Clients upgrade by accessing the new proxy service. This enables you to run the old and new versions of a proxy service in parallel, and supports a gradual migration to the new proxy service.
- Force backwards compatibility by changing the proxy service interface to support both the new interface and the old interface (for example, using XML schema choice) and perform different logic in the message flow based on the document received. Clients continue to access the proxy service by using its original URL.

AquaLogic Service Bus cluster domains have additional system administration requirements for deployment of proxy services that are not backward compatible. For more information, see [“Installing a New Version of a Proxy Service in a Cluster”](#) on page 4-16.

## Online Configuration Updates

AquaLogic Service Bus allows you to change the configuration information for a system dynamically without the need to restart the server for changes to take affect.

You can change a resource, a project, or a number of resources (related or unrelated) using the AquaLogic Service Bus Console using the following procedure:

1. Create a session. All changes to AquaLogic Service Bus configurations require a session. (Security-related changes are the exception.)



2. Modify resources in the session, or import all or selected objects from a configuration JAR file.
3. Update environment-specific information such as service endpoint URIs and directory names.
4. Activate the session..

The changes are consolidated and sent to all servers (administration and managed servers, if you are working in a cluster environment). These changes update the persisted configuration data and also cause other run-time tasks to be performed (such as, creating proxy services and JMS queues, compiling XQueries, and so on).

You can perform these steps manually or programmatically:

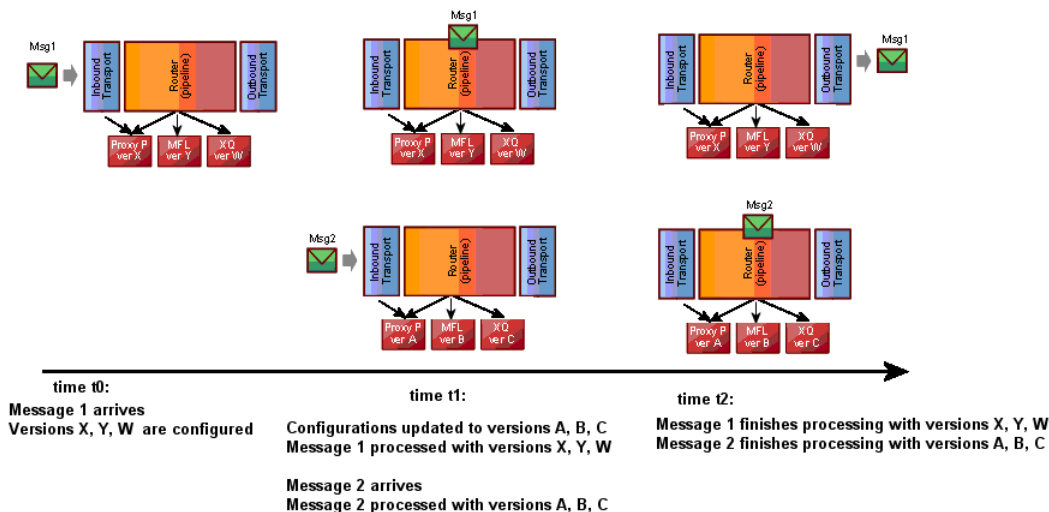
- To import and update a configuration manually, use the AquaLogic Service Bus Console as described in the following topics in the *Using the AquaLogic Service Bus Console*:
  - “Importing Configuration Data” in [System Administration](#)
  - “Finding and Replacing Environment Values” in [Using the Change Center](#)
- To import and update a configuration programmatically, use the WebLogic Scripting Tool (WLST) and the AquaLogic Service Bus `deploymentMBean` as described in [Appendix A](#), “Using the AquaLogic Service Bus Deployment API.”

[Figure 2-1](#) illustrates how the system behaves to process messages in the event that the configuration is updated while messages are being processed through the system. [Table 2-2](#) describes the versions for the resources for the sample system illustrated in [Figure 2-1](#).

**Table 2-2 Initial and Updated Configuration for a Sample System**

Resource	Initial Version	Updated Version
Proxy Service	X	A
MFL	Y	B
XQuery	W	C

**Figure 2-1 Sample Online Update Scenario**



Note the following characteristics of the message processing illustrated in the preceding figure:

- Message 1 is already in the system at t1 (the time the configuration is updated)
- Message 1 completes processing by the original (pre-update) resources (X, Y, W)
- Message 2 starts and completes processing with the new configuration (resources A, B, C)

AquaLogic Service Bus tries to execute messages with the version of the proxy service and artifacts available as of the time the messages enters the proxy service.

This ensures that a message has a consistent view of the artifacts. If the message processor cannot guarantee this behavior for a message, it will reject it rather than process it incorrectly. If you want the system to retry rejected messages, use a JMS proxy service with retries.

## Best Practices for Successful Online Configuration Updates

This section describes best practices you should follow and limitations to be aware of when you update a configuration in a running AquaLogic Service Bus system.

- If you are concerned about message rejection by AquaLogic Service Bus, use the JMS transport protocol with retries. In this case, any messages that are rejected because the system cannot guarantee their processing by compatible resources will be retried.

## Step 5. Update Your Domain as Your Production Environment Changes

- Security-related configuration updates must be performed first, then update the AquaLogic Service Bus resources in your system. To learn about updating security resources, see [Overview of Security Management](#) in *Securing WebLogic Server*.
- Updates must be compatible with existing clients using the system. See [“Installing a New Version of a Proxy Service” on page 2-8](#).
- If you are updating the configuration to a cluster, it is possible that the updates are done at different times on different managed servers. Consequently, it is possible that messages are processed by different versions of a proxy service, depending on which managed server gets the message to process. This is dependent on load balancing across managed servers.

## Configuring a Single-Server Deployment

# Understanding AquaLogic Service Bus Clusters

The following sections describe how AquaLogic Service Bus is configured and deployed in a clustered environment. It contains the following topics:

- [Understanding AquaLogic Service Bus Clusters](#)
- [Designing a Clustered Deployment](#)
- [Load Balancing in a AquaLogic Service Bus Cluster](#)
- [High Availability in a AquaLogic Service Bus Cluster](#)
- [Deploying Configurations](#)

## Understanding AquaLogic Service Bus Clusters

Clustering allows AquaLogic Service Bus to run on a group of servers that can be managed as a single unit. In a clustered environment, multiple machines share the processing load. AquaLogic Service Bus provides load balancing so that resource requests are distributed proportionately across all machines. An AquaLogic Service Bus deployment can use clustering and load balancing to improve scalability by distributing the workload across nodes. Clustering provides a deployment platform that is more scalable than a single server.

A WebLogic Server cluster domain consists of only one administration server, and one or more managed servers. The managed servers in a AquaLogic Service Bus domain can be grouped in a cluster. When you configure AquaLogic Service Bus clusterable resources, you normally target the resources to the named cluster. The advantage of specifying a cluster as the target for resource deployment is that it makes it possible to increase capacity dynamically by adding managed servers to your cluster.

**Note:** AquaLogic Service Bus domains can support a single cluster, and all managed servers in the domain must belong to that cluster.

The topics in this section provide the information you need to configure AquaLogic Service Bus in a clustered environment. Although some background information about how WebLogic Server supports clustering is provided, the focus is on procedures that are specific to configuring AquaLogic Service Bus for a clustered environment.

Before proceeding, we recommend that you review the following sections of the WebLogic Server documentation to obtain a more in-depth understanding of clustering:

- [Using WebLogic Server Clusters](#)
- [Configuring WebLogic Server Environments](#)

## Designing a Clustered Deployment

The following sections provide the information you need to design a clustered deployment:

- [Introducing AquaLogic Service Bus Domains](#)
- [AquaLogic Service Bus Deployment Resources](#)
- [Load Balancing in a AquaLogic Service Bus Cluster](#)

## Introducing AquaLogic Service Bus Domains

Before you begin designing the architecture for your clustered domain, you need to learn how WebLogic Server clusters operate.

### Creating Domains

Domain and cluster creation are simplified by a Configuration Wizard that lets you generate domains from basic and extension domain templates. Based on responses to user queries, the Configuration Wizard generates a domain, server, and enterprise application with the appropriate components preconfigured and assets included. For information about creating AquaLogic

Service Bus domains using the Configuration Wizard, see [Creating a New WebLogic Domain](#) in *Creating WebLogic Domains Using the Configuration Wizard*.

## Clustered Servers

A server can be either a managed server or an administration server. A WebLogic Server running the administration service is called an *administration server* and hosts the WebLogic Server Administration Console. In a domain with multiple WebLogic Servers, only one server is the administration server; the other servers are called *managed servers*. Each managed server obtains its configuration at startup from the administration server.

For general information about WebLogic clusters, see [Using WebLogic Server Clusters](#) in the WebLogic Server documentation set. This document includes details regarding recommended basic, multi-tiered, and proxy architectures. For information about security considerations in the design of WebLogic clusters, see “Security Options for Cluster Architectures” in [Cluster Architectures](#) in *Using WebLogic Server Clusters*.

## AquaLogic Service Bus Deployment Resources

For each server in a clustered domain, you can configure a variety of attributes that define the functionality of the server in the domain. These attributes are configured automatically when you create an AquaLogic Service Bus domain using the Configuration Wizard. Advanced users can also configure these attributes manually using the Servers node in the WebLogic Server Administration Console.

For a list of configurable AquaLogic Service Bus deployment resources, see [Appendix B, “AquaLogic Service Bus Deployment Resources.”](#) It describes the default targeting of each resource in a clustered AquaLogic Service Bus domain and provides instructions on how to navigate to each resource in the WebLogic Server Administration Console.

## Singleton Resources

While most resources used by AquaLogic Service Bus are deployed homogeneously across the cluster, there are a number of resources that must be pinned to a single managed server in order to operate correctly. The following table lists these components, describes how they are targeted, and provides a link for more information.

**Table 3-1 AquaLogic Service Bus Singleton Resources**

Resource	Managed server target selection	For more information, see...
File, FTP, and Email pollers for proxy services	Specified manually in the proxy service definition using the AquaLogic Service Bus Console. The poller is deployed on all managed servers, but the poller on only one managed server will poll for a given proxy service.	“Viewing and Changing Proxy Services” in <a href="#">Proxy Services</a> in <i>Using the AquaLogic Service Bus Console</i> .
AquaLogic Service Bus Data Aggregator (weblogic90/integration/communication/wliaggregator.ear)	Targeted in the Configuration Wizard	“Creating an AquaLogic Service Bus Domain Using the Configuration Wizard” on <a href="#">page 4-5</a>
SLA Manager	Targeted in the Configuration Wizard	
AquaLogic Service Bus JMS server on each managed server wlsbJMSServer_auto_1-n	Automatically targeted in the Configuration Wizard	

### Monitoring and Alert Resources in a Cluster

The AquaLogic Service Bus Data Aggregator runs as a singleton service on one of the managed servers in the cluster. Data collection is performed on each of the managed servers in the domain. The aggregator is responsible for the collection and aggregation of data from all managed servers in the domain. The aggregated data are processed and classified by each AquaLogic Service Bus configuration.

AquaLogic Service Bus configurations can include rules defining Service Level Agreements (SLAs) for system performance. The Alert Manager is responsible for storing rules, and evaluates these rules against the data aggregated for the cluster. When a rule evaluates to `true`, the Alert Manager sends an email message, posts a message on a JMS queue, or logs a message according to the action associated with the rule.

For more information about these features, see [Monitoring](#) in *Using the AquaLogic Service Bus Console*.



## Cluster Configuration Changes and Deployment Requests

You can only change configuration for a cluster (for example, add new nodes to the cluster or modify business service configuration) while its administration server is active.

If the administration server for a cluster is down, deployment or undeployment requests are interrupted, but managed servers continue serving requests. You can boot or reboot managed servers using an existing configuration, as long as the required configuration files (`msi-config.xml`, `SerializedSystemIni.dat`, and optionally `boot.properties`) exist in each managed server's root directory.

**Note:** Managed servers that start without an administrative server operate in Managed Server Independence (MSI) mode. For information about MSI mode, see “Managed Server Independence Mode” in [Managing Server Startup and Shutdown](#).

## Load Balancing in a AquaLogic Service Bus Cluster

One of the goals of clustering your AquaLogic Service Bus application is to achieve scalability. In order for a cluster to be scalable, each server must be fully utilized. Load balancing distributes the workload proportionally among all the servers in a cluster so that each server can run at full capacity. The following sections describe inbound message processing load balancing for AquaLogic Service Bus clusters:

- [Load Balancing HTTP Functions in a Cluster](#)
- [Load Balancing JMS Functions in a Cluster](#)

For more information about inbound message load balancing, see [Load Balancing in a Cluster](#) in *Using WebLogic Server Clusters*. For information about configuring load balancing for business services, see “To Add a Business Service - Transport Configuration” in “Adding a Business Service” in [Business Services](#) in *Using the AquaLogic Service Bus Console*.

## Load Balancing HTTP Functions in a Cluster

Web services (SOAP or XML over HTTP) can use HTTP load balancing. External load balancing can be accomplished through the WebLogic HttpClusterServlet, a WebServer plugin, or a hardware router. For an overview of a cluster topology that includes load balancing, see [Figure 5-1](#).

WebLogic Server supports load balancing for HTTP session states and clustered objects. For more information, see [Communications in a Cluster](#) in *Using WebLogic Server Clusters*.

## Load Balancing JMS Functions in a Cluster

Most JMS queues used by AquaLogic Service Bus are configured as distributed destinations. The exceptional cases are JMS queues that are targeted to single managed servers.

For detailed information on JMS load balancing, see “Controlling the Flow of Messages on JMS Servers and Destinations” in [Tuning WebLogic JMS](#) in *Configuring and Managing WebLogic JMS*.

## High Availability in a AquaLogic Service Bus Cluster

Message-driven beans consume messages from JMS destinations. A number of message-driven beans are deployed on each AquaLogic Service Bus destination. For a complete list of AquaLogic Service Bus destinations (JMS queues and topics), see the resource type of Services—JMS in [Table B-1, “AquaLogic Service Bus Deployment Resources,” on page B-2](#).

## Highly Available JMS for AquaLogic Service Bus

The ability to configure multiple physical destinations as members of a single distributed destination set provides a highly available implementation of WebLogic JMS. Specifically, for each node in a cluster, an administrator should configure one physical destination for a distributed destination. If one node in the cluster fails, making the physical destination for that node unavailable, then other physical destinations configured as members of the distributed destination can provide service to JMS producers and consumers. (This is the way the Configuration Wizard generates domains for a cluster.)

Message-driven beans consume messages from distributed destinations. Distributed destinations contain one physical destination for each instance of WebLogic Server. A single message producer on a distributed queue is bound to a single physical destination. Message-driven beans are bound to the physical destination in the server on which they are deployed (server affinity).

For more information, see [Chapter 5, “Understanding AquaLogic Service Bus High Availability.”](#)

## Deploying Configurations

Configurations are deployed by importing the contents of one or more JAR files exported from the AquaLogic Service Bus Console. You deploy an AquaLogic Service Bus configuration in a clustered environment following the same procedure as for a single-server deployment. For a description of the deployment procedure, see [“Step 4. Deploy an AquaLogic Service Bus Configuration”](#) on page 2-6.

Preparations for deployment of an AquaLogic Service Bus configuration in a production cluster environment involve more system administration tasks than for a single-server testing or staging environment. For a full description of the steps involved in a production cluster deployment, see [Chapter 4, “Configuring a Clustered Deployment.”](#)



# Configuring a Clustered Deployment

This section describes the tasks that you must perform to configure AquaLogic Service Bus for deployment in a clustered environment.

After planning the architecture of your clustered domain, as described in [“Designing a Clustered Deployment” on page 3-2](#), you are ready to set up AquaLogic Service Bus in a clustered environment. To do this, you must configure an administration server and managed servers, and then deploy AquaLogic Service Bus resources to the servers. You also need a router (hardware or software), if you need inbound HTTP load balance functions. The persistent configuration for a domain of WebLogic Server instances and clusters is stored in an XML configuration file (`config.xml`) in the `config` directory of the root directory of your AquaLogic Service Bus domain.

To set up and deploy AquaLogic Service Bus in a clustered domain, complete the following steps:

- [Step 1. Comply with Configuration Prerequisites](#)
- [Step 2. Prepare a AquaLogic Service Bus Domain](#)
- [Step 3. Configure WebLogic Integration Security](#)
- [Step 4. Starting, Stopping, and Monitoring Managed Servers](#)
- [Step 5. Deploy an AquaLogic Service Bus Configuration](#)
- [Step 6. Update Your Domain as Your Production Environment Changes](#)

For information about deploying AquaLogic Service Bus on a single server, see [Chapter 2, “Configuring a Single-Server Deployment.”](#)

## Step 1. Comply with Configuration Prerequisites

This section describes prerequisites for configuring AquaLogic Service Bus to run in a clustered environment:

- Obtain a WebLogic Server cluster license for each required installation.

To use WebLogic Server in a clustered configuration, you must have a special cluster license. Contact your BEA representative for information about obtaining one.

- Obtain an IP address for the administration server you will use for the cluster.

All WebLogic Server instances in a cluster use the same administration server for configuration and monitoring. When you add servers to a cluster, you must specify the administration server that each will use.

- Define a multicast address for each cluster

**Note:** You are prompted to provide a multicast address when you create a BEA AquaLogic Service Bus domain using the Configuration Wizard. (See [“Step 2. Prepare a AquaLogic Service Bus Domain” on page 4-5.](#))

The multicast address is used by cluster members to communicate with each other. Clustered servers must share a single, exclusive multicast address. For each cluster on a network, the combination of multicast address and port must be unique. If two clusters on a network use the same multicast address, they should use different ports. If the clusters use different multicast addresses, they can use the same port or accept the default port, 7001. To support multicast messages, the administration server and the managed servers in a cluster must be located on the same subnet.

- Define IP addresses for the servers in your cluster. You can do this in a number of ways:

**Note:** You are prompted to provide listen addresses for servers when you create a BEA AquaLogic Service Bus domain using the Configuration Wizard. (See [“Step 2. Prepare a AquaLogic Service Bus Domain” on page 4-5.](#))

- Assign a single IP address and different listen port numbers to the servers in the cluster.

By assigning a single IP address for your clustered servers with a different port number for each server, you can set up a clustered environment on a single machine without the need to make your machine a multihomed server.

To access such an IP address from a client, structure the IP address and port number in your URL in one of the following ways:

**Table 4-1 IP Address Structure**

<i>ipAddress:portNumber-portNumber</i>	When the port numbers are sequential, for example: 127.0.0.1:7003-7005
<i>ipAddress:portNumber+...+portNumber</i>	When the port numbers are not sequential, for example: 127.0.0.1:7003+7006+7008
<i>ipAddress:portNumber, ipAddress:portNumber, ...</i>	Verbose, explicit specification, for example: 127.0.0.1:7003, 127.0.0.1:7004, 127.0.0.1:7005

- Assign a static IP address for each WebLogic Server instance to be started on each machine in the cluster.

In this case, when multiple servers are run on a single machine, that machine must be configured as a multihomed server, that is, multiple IP addresses are assigned to a single computer. Under these circumstances, you structure the cluster address as a comma-separated list of IP addresses.

For example, the following listing is an example of a cluster address specified in a `config.xml` file. It specifies a static IP address for each of the four servers in a cluster named `MyCluster`:

```
<Cluster
ClusterAddress="127.0.0.1:7001,127.0.0.2:7001,127.0.0.3,127.0.0.4:7001" Name="MyCluster"/>
```

You can also use a DNS approach to identifying servers.

For more information on addressing issues, see “Avoiding Listen Address Problems” in [Setting Up WebLogic Clusters](#) in *Using WebLogic Server Clusters*.

**Note:** In test environments, it is possible to have multiple WebLogic Server instances on a single machine. In these circumstances, you can have some WebLogic Server instances on the same node with different port numbers and some on different nodes with the same port number.

- Configure one of the following databases for your clustered domain:
  - Microsoft SQL Server
  - Oracle

**Note:** The local copy of the PointBase database that is installed with WebLogic Server is for evaluation purposes only. Non-evaluation development or other use of the PointBase Server requires that a separate PointBase license be obtained by the end user directly from DataMirror.

It is important to configure your database appropriately for production use. You must provide adequate space to store data and log messages, and follow best practices for administering your database.

**Note:** You can configure your database to use concurrent access.

For the latest information about issues regarding specific databases, see the [BEA AquaLogic Service Bus Release Notes](#).

- Include a shared file system. A shared file system is required for any cluster you want to be highly available. We recommend either a Storage Area Network (SAN) or a multiported disk system.

For information about configuring a highly available cluster, see “Configuring WebLogic JMS Clustering” in [Configuring Clustered WebLogic JMS Resources](#) in *Configuring and Managing WebLogic JMS*.

- Configure a hardware or software router for your system. Load balancing can be accomplished using either the built-in load balancing capabilities of a WebLogic proxy plug-in or separate load balancing hardware.

For information about hardware and software routers, see [Using WebLogic Server Clusters](#).

**Note:** Additional requirements apply when you design your domain to include one or more firewalls. For a description of how to add firewall information to your domain configuration file, see “[Adding Proxy Server or Firewall Information to your Domain Configuration](#).” For additional information, see [Communications in a Cluster](#) in *Using WebLogic Server Clusters*.

- AquaLogic Service Bus load balances File, Email, and FTP transport processing across the managed servers in a cluster. All managed servers in the cluster should be able to access the Archive, Stage, and Error directories specified in any File, Email, or FTP proxy service configuration. These directories should be configured in a shared file system such as NFS. By using a shared file system, users and programs can access files on remote systems almost as if they were local files.

For information on how to configure the Archive, Stage, and Error directories, see “Viewing and Changing Proxy Services” in [Proxy Services](#) in the *Using the AquaLogic Service Bus Console*.



For more information about setting up clustered WebLogic Server instances, see [Setting Up WebLogic Clusters](#) in *Using WebLogic Server Clusters*.

## Step 2. Prepare a AquaLogic Service Bus Domain

When preparing an AquaLogic Service Bus domain, you must add a definition for each managed server to the domain configuration file (`config.xml`), assign all managed servers to a cluster, specify the AquaLogic Service Bus components on the servers in your domain, and so on.

To prepare an AquaLogic Service Bus environment in a clustered domain, complete the tasks described in the following sections:

- [Creating an AquaLogic Service Bus Domain Using the Configuration Wizard](#)
- [Configuring JMS Resources](#)

### Creating an AquaLogic Service Bus Domain Using the Configuration Wizard

You begin the definition of an AquaLogic Service Bus deployment by creating a domain using the Configuration Wizard.

**Note:** The procedure described in this section for setting up your domain is based on the assumption that you are running the Configuration Wizard in GUI mode from the Windows Start menu. For information about using the Configuration Wizard in different modes, see [Creating WebLogic Configurations Using the Configuration Wizard](#).

To create an AquaLogic Service Bus domain using the Configuration Wizard, complete the following steps:

1. From the Start Menu, choose **All Programs**→**BEA Products**→**Tools**→**Configuration Wizard**.

The Configuration Wizard is launched. It prompts you for data with which to configure your domain.

2. Respond to the Configuration Wizard prompts by providing the information described in the following table.

**Note:** To comply with WebLogic Server resource naming rules, you must specify unique names for domains, WebLogic Server instances, JMS servers, and JMS stores. AquaLogic Service Bus has the same interoperability naming requirements as the WebLogic Messaging Bridge. For more information, see “Naming Guidelines for

WebLogic Servers and Domains” in “Interoperating with Different WebLogic Server Releases” in [Interoperating with Different WebLogic Server Releases or Foreign Providers](#) in *Configuring and Managing the WebLogic Messaging Bridge*.

**Table 4-2 Responses to Configuration Wizard Prompts**

In this window . . .	Perform the following action . . .
Welcome	Select Create a new WebLogic domain.
Select Domain Source	Select AquaLogic Service Bus.
Configure Administrator Username and Password	Enter user name and password.
Configure Server Start Mode and JDK	Select Production Mode. Select either the Sun SDK or JRockit SDK, or specify the location of another JDK.
Customize Environment and Services Settings	Select Yes.
Configure the Administration Server	If your configuration requires SSL, select the SSL enabled check box.
Configure Managed Servers	<p>Add as many managed servers as required.</p> <p><b>Note:</b> If you need a software HTTP router with WebLogic HttpClusterServlet in your domain for load balancing, add one extra server here.</p> <p>If your configuration requires SSL, select the SSL enabled check box for each managed server.</p>
Configure Machines	Configure the type of physical machines used in the cluster.
Configure Clusters	<p>Add a cluster.</p> <p><b>Note:</b> AquaLogic Service Bus domains can support a single cluster.</p>
Assign Servers to Clusters	<p>Add the previously created managed servers to the cluster.</p> <p><b>Note:</b> If you had previously configured a managed server as an HTTP router, do not add it to the cluster. Select this managed server from the Proxy Server drop-down list on the next window, Create HTTP Proxy Applications.</p>
Configure Machines	Configure the type of physical machines used in the cluster.

Assign Servers to Machines	Assign each instance of WebLogic Server to the machine in the cluster on which it runs.
Configure JDBC Data Sources	<p>Accept the default values for Name and JNDI name of <code>wlsbjmsrpDataSource</code>.</p> <p>Select one of the following to identify the database type and driver for the JMS Reporting Provider Data Store:</p> <ul style="list-style-type: none"> <li>• Oracle and BEA's Oracle Driver (Type 4) Versions: 9.0.1, 9.2.0, 10</li> <li>• MS SQL Server and BEA's MS SQL Server Driver (Type 4) Versions: 7.0, 2000</li> </ul> <p>Confirm that the Supports global transactions check box and Logging last resource option are selected. For more information about Logging last resource (LLR), see "Understanding the Logging Last Resource Transaction Option" in <a href="#">Configuring JDBC Data Sources</a> in <i>Configuring and Managing WebLogic JDBC</i>.</p> <p>Enter your environment-specific database information in the remaining text boxes. For more information, see "Configure JDBC Data Sources" in <a href="#">Customizing Existing JDBC and JMS Settings</a> in <i>Creating WebLogic Configurations Using the Configuration Wizard</i>.</p> <p><b>Note:</b> Click Test Connections to verify that you can contact the database you want to use for the JMS Reporting Provider Data Store using this data source configuration.</p>
Run Database Scripts	<p>Select <code>wlsbjmsrpDataSource</code> from the Available JDBC Data Sources list.</p> <p>Select the version of the database from the DB Version drop-down list.</p> <p>Click Run Scripts.</p> <p>The scripts create the tables and indexes for the JMS Reporting Provider Data Store. The SQL output is displayed in the Results box and written to <code>jdbc.log</code>. If you want the results written to a different file, click the Log File check box and specify the file.</p> <p><b>Note:</b> AquaLogic Service Bus does not automatically run database scripts for the JMS Reporting Provider Data Store the first time you start a production domain, as it does for domains in development mode. If you do not run the database scripts while creating your AquaLogic Service Bus production domain, you must run the scripts manually. The scripts are located in <code>BEA_HOME/weblogic90/integration/common/dbscripts</code>, where <code>BEA_HOME</code> is the directory in which you installed AquaLogic Service Bus.</p>

Configure JMS File Stores	Accept the defaults for <code>rmfilestore</code> and add any file stores needed by proxy services or business services.
Review WebLogic Domain	Accept the default values.
Create WebLogic Configuration	Enter a name and location for your domain.
Creating Domain	After your domain is created, select Start Admin Server to start AquaLogic Service Bus when you exit the Configuration Wizard.

When you complete the domain configuration using the Configuration Wizard, your new domain is created in the location you specified.

Your AquaLogic Service Bus domain includes a configuration file (`config.xml`) that contains a definition for the administration server. For more information, see “`config.xml`” in [Domain Configuration Files](#) in *Understanding Domain Configuration*.

For information about configuring domains without using the Configuration Wizard, see [Understanding Domain Configuration](#).

## Adding Proxy Server or Firewall Information to your Domain Configuration

If you will be using Web services behind a proxy server or firewall, you must edit the `config.xml` file to include information about that proxy server or firewall.

To add proxy server or firewall information to your domain configuration, complete the following steps:

1. Open `config.xml` with an ASCII editor.
2. Find the line that starts with the following tag in the `config.xml` file:

```
<Cluster
```

3. Add the following three attributes to the Cluster attribute list:

```
FrontendHTTPPort="proxyPort" FrontendHTTPSPort="proxySSLPort"  
FrontendHost="proxyServerHost"
```

For example, the following listing is an example of a cluster address with a firewall specified in a `config.xml` file for a cluster named `MyCluster` and a proxy server named `MyProxy`:

```
<Cluster  
ClusterAddress="127.0.0.1:7001,127.0.0.2:7001,127.0.0.3,127.0.0.4:7001"
```

```
FrontendHTTPPort="7006" FrontendHTTPSPort="7007" FrontendHost="MyProxy"
MulticastAddress="127.0.0.5" MulticastPort="7010" Name="MyCluster"/>
```

4. Save your changes and close the `config.xml` file.

## Configuring JMS Resources

In addition to configuring JMS file stores in the Configuration Wizard, proxy services and business services that use JMS require configuration of the following resources:

- JMS queues/topics. AquaLogic Service Bus automatically configures JMS queues for proxy services that are implemented using BEA JMS. You must configure JMS queues/topics for all business services using JMS and for any proxy services that are implemented using non-BEA JMS.

Proxy services can consume messages from a remote queue on a separate BEA domain. In this case, AquaLogic Service Bus will not create the queue for you. The JMS queues can be created for proxy services only if the queues are on the same local AquaLogic Service Bus domain.

- JMS connection factories. You must configure JMS connection factories for all business services and proxy services implemented using JMS.

For information about configuring JMS resources, see [Configuring and Managing WebLogic JMS](#).

## Step 3. Configure WebLogic Integration Security

AquaLogic Service Bus leverages the security features of WebLogic Server to ensure message confidentiality and integrity (message-level security), secure connections between clients and WebLogic Server (transport-level security), and authentication and authorization (access control). For information about the tasks you must complete, see [Securing Inbound and Outbound Messages](#) in the *BEA AquaLogic Service Bus User Guide*.

**Warning:** You must configure security separately for each AquaLogic Service Bus domain. AquaLogic Service Bus does not export or import security configurations.

If you want to configure SSL for your cluster, you can do so when creating your domain or by using the WebLogic Server Administration Console. For a domain in which security functionality is deployed in a multinode cluster, you also need to configure keystores, server certificate and private key for each managed server, and so on, for every machine in a cluster. You either need to use a separate keystore for each machine or you can use a single keystore if it is available to all machines.

## Step 4. Starting, Stopping, and Monitoring Managed Servers

This section describes the basic management tasks for the managed servers in your clustered domain:

- [Starting and Stopping Managed Servers](#)
- [Monitoring Your Servers](#)

### Starting and Stopping Managed Servers

Node Manager is a utility that enables you to start, stop, and migrate your WebLogic Server instances. You can start your managed servers using Node Manager in conjunction with the WebLogic Server Administration Console, or you can create WLST scripts to automate Node Manager functionality.

By default, when the Configuration Wizard generates an AquaLogic Service Bus cluster domain:

- The WLIAggregator (`wliagggregator.ear`) is targeted to the first managed server in the cluster. For data aggregation to function properly, the server that `wliagggregator.ear` is targeted to must be started first and must be available when other managed servers are started.
- The PurgingMDB (`msgpurger.ear`) is targeted to the first managed server in the cluster. For message purging to function properly, the server where `msgpurger.ear` is targeted must be available.

For more information on Node Manager, see [Using Node Manager to Control Servers](#) in *Managing Server Startup and Shutdown*. For a complete overview of methods to start and stop managed servers, see [Starting and Stopping Servers](#) in *Managing Server Startup and Shutdown*.

### Monitoring Your Servers

Once startup is complete, you can use the AquaLogic Service Bus Console to verify the status of servers. For information about using AquaLogic Service Bus Console to monitor your servers, see “Listing and Locating Servers” in [Monitoring](#) in *Using the AquaLogic Service Bus Console*.

## Step 5. Deploy an AquaLogic Service Bus Configuration

You deploy an AquaLogic Service Bus configuration in a clustered environment following the same procedure as for a single-server deployment. For a description of the deployment procedure, see [“Step 4. Deploy an AquaLogic Service Bus Configuration” on page 2-6](#).

**Note:** If you have imported a configuration from a single-server environment and that configuration includes proxy services that use File, FTP, or Email transports, you must specify a Managed Server for each of those proxy services. The Managed Server drop-down list appears in the AquaLogic Service Bus Console in clustered AquaLogic Service Bus domains only.

For information on how to edit the Managed Server value for a proxy service, see “Viewing and Changing Proxy Services” in [Proxy Services](#) in *Using the AquaLogic Service Bus Console*.

## Step 6. Update Your Domain as Your Production Environment Changes

Production environments change over time and as application use increases. This section describes how to update your domain in response to common production environment change scenarios:

- [Adding a Managed Server](#)
- [Dropping a Managed Server](#)
- [Changing a Business Service in a Cluster](#)
- [Installing a New Version of a Proxy Service in a Cluster](#)

### Adding a Managed Server

As use of AquaLogic Service Bus grows, you can add new managed servers to your AquaLogic Service Bus cluster to increase capacity. You add a managed server to the cluster using the WebLogic Server Administration Console. Depending on your configuration, you may also need to perform administrative tasks using the AquaLogic Service Bus Console.

## Adding A Managed Server to an AquaLogic Service Bus Cluster

To add a new managed server to your AquaLogic Service Bus cluster, perform the steps below using the WebLogic Server Administration Console:

1. Verify that the managed server that you want to add to the AquaLogic Service Bus cluster is not running. Stop the server, if necessary. For information on how to stop a managed server, see [Starting and Stopping Servers](#) in *Managing Server Startup and Shutdown*.
2. If you have not already done so, in the Change Center, click Lock & Edit.
3. Expand Environment, and select Clusters. On the Summary of Clusters panel, change the cluster address to include the address of the new server.
4. Expand Services, and select Persistent Stores. Define a new FileStore on the new server, and assign it to a non-migratable server.

**Note:** Be sure to create the directory that you specify in the FileStore definition on the new server.

5. Expand Services and Messaging, and then select JMS Servers. Create a new JMS server on the new server using the new FileStore, and target it to the same non-migratable server.
6. On the Summary of JMS Servers panel, click the name of the new JMS server. Select `jmsResources` in the Module Containing Temporary Template list, enter `TemporaryTplt` in the Temporary Template Name box, and click Save.
7. Create the necessary queues and add them to the corresponding distributed destinations by performing the following steps:
  - a. If you have not already done so, expand Services and Messaging.
  - b. Select JMS Modules, and click `jmsResources`.
  - c. Create a new queue named `QueueIn_auto_x`, accept the default value for Template, and target the queue to the new server.

**Note:** For the names of queues specified in this step through step 8., `x` is the number of managed servers currently in the AquaLogic Service Bus cluster incremented by 1. For example, if you were adding a managed server to a cluster that currently contains two managed servers, then `x` equals 3. You would be creating `QueueIn_auto_3` in this step, and the names of the queues you create in step d through step g would also end with 3.

While defining `QueueIn_auto_x`, create a new subdeployment (for example, `sub_new`), and then select the JMS server on the new server as the target.



**Note:** You will reuse this subdeployment in step d through step g, instead of creating a new subdeployment for each queue.

- d. Create `wli.reporting.jmsprovider.queue_auto_x`, and target it to the new server.
  - e. Create `wlsb.internal.transport.task.queue.email_auto_x`, and target it to the new server.
  - f. Create `wlsb.internal.transport.task.queue.file_auto_x`, and target it to the new server.
  - g. Create `wlsb.internal.transport.task.queue.ftp_auto_x`, and target it to the new server.
8. Expand Services, Messaging, and JMS Modules, and then select the `configwiz-jms` module. Add the queues that you created in step c through step g to their corresponding distributed destinations:
    - a. Add `QueueIn_auto_x` to `dist_QueueIn_auto`.
    - b. Add `wli.reporting.jmsprovider.queue_auto_x` to `dist_wli.reporting.jmsprovider.queue_auto`.
    - c. Add `wlsb.internal.transport.task.queue.email_auto_x` to `dist_wlsb.internal.transport.task.queue.email_auto`.
    - d. Add `wlsb.internal.transport.task.queue.file_auto_x` to `dist_wlsb.internal.transport.task.queue.file_auto`.
    - e. Add `wlsb.internal.transport.task.queue.ftp_auto_x` to `dist_wlsb.internal.transport.task.queue.ftp_auto`.
  9. In the Change Center, click Activate.
  10. You can now start the new managed server. For information on how to start a managed server, see [Starting and Stopping Servers](#) in *Managing Server Startup and Shutdown*.
  11. If your cluster has an HTTP load balancer (software or hardware), add the new managed server to the server list of the load balancer. For example, if you are using the WebLogic `HttpClusterServlet`, you should add the new server to the `web.xml` file for the corresponding application. For more information, see [Load Balancing in a Cluster](#) in *Using WebLogic Server Clusters*.

## Updating Business Service Configurations for an Expanded Cluster

If your AquaLogic Service Bus configuration includes one or more business services that use JMS request/response functionality, then you must also perform the following procedure using the AquaLogic Service Bus Console after adding the new managed server to the cluster:

1. In the Change Center, click Create to create a session.
2. Using the Project Explorer, locate and select a business service that uses JMS request/response. Business services of this type display Messaging Service as their Service Type.
3. At the bottom of the View Details page, click Edit.
4. If there is a cluster address in the endpoint URI, add the new server to the cluster address.
5. On the Edit a Business Service - Summary page, click Save.
6. Repeat step 2. through step 5. for each remaining business service that uses JMS request/response.
7. In the Change Center, click Activate.

The business services are now configured for operation in the extended domain.

## Updating Proxy Service Configurations for an Expanded Cluster

If your AquaLogic Service Bus configuration includes one or more proxy services that use JMS endpoints with cluster addresses, then you must also perform the following procedure using the AquaLogic Service Bus Console after adding the new managed server to the cluster:

1. In the Change Center, click Create to create a session.
2. Using the Project Explorer, locate and select a proxy service that uses JMS endpoints with cluster addresses.
3. At the bottom of the View Details page, click Edit.
4. If there is a cluster address in the endpoint URI, add the new server to the cluster address.
5. On the Edit a Proxy Service - Summary page, click Save.
6. Repeat step 2. through step 5. for each remaining proxy service that uses JMS endpoints with cluster addresses.
7. In the Change Center, click Activate.

The proxy services are now configured for operation in the extended domain.

## Dropping a Managed Server

Using WebLogic Server administration tools, you can drop a managed server from your AquaLogic Service Bus cluster. Before deciding to drop a managed server, you should take into account the following considerations:

- If your AquaLogic Service Bus configuration includes one or more proxy services that use File, FTP, or Email transports that have pinned transport pollers to the managed server that you want to remove from the cluster, then you must select a different managed server for each of those proxy services *before* removing the managed server from the cluster. For information on how to edit the Managed Server value for a proxy service, see “Viewing and Changing Proxy Services” in [Proxy Services](#) in *Using the AquaLogic Service Bus Console*.
- The managed server that hosts the WLI Aggregator application must not be dropped from the cluster. If the managed server that hosts the WLI Aggregator application fails, you must perform a manual migration.
- If you want to remove the managed server that hosts the Message Reporting Purger application, you must select a different manager server for the Message Reporting Purger and its associated queue (`wli.reporting.purge.queue`).

For information about dropping a managed server from a cluster, see “Delete Managed Servers” in [Configure Domains](#) in the WebLogic Server Administration Console Online Help.

## Changing a Business Service in a Cluster

The procedure for changing a business service is the same in both single-server and cluster environments. For information about changing a business service, see “[Changing a Business Service](#)” on page 2-7.

However, the procedure for deploying changes to a business service in a cluster depends on the types of changes made to the business service and the nature of any other changes that might be deployed simultaneously. For more information, see the description of installation strategies in the following section.

## Installing a New Version of a Proxy Service in a Cluster

As your business requirements change, you may need to make changes to your proxy services. You can make these changes dynamically online, partially offline, or completely offline. If your changes are backward compatible (that is, you are making no changes to interfaces), you can make your changes dynamically online using the AquaLogic Service Bus Console. Making other types of changes should be done partially or completely offline, which requires additional system administration steps.

For information about performing online updates, see [“Online Configuration Updates” on page 2-8](#).

Making non-backward compatible interface changes to resources referenced by a proxy service requires a partial offline deployment to avoid causing transient problems for messages being processed. To install the new version, follow the procedure below:

1. Gracefully shut down the managed servers in the AquaLogic Service Bus cluster.

For more information about shutting down managed servers, see “Control Graceful Shutdowns” in [Start and Stop Servers](#) in the WebLogic Server Administration Console Online Help.

2. Make the necessary changes in the proxy service.
3. Restart the managed servers.

For more information about restarting managed servers, see “Start Managed Servers from the Administration Console” in [Start and Stop Servers](#) in the WebLogic Server Administration Console Online Help.

Making changes that include non-backward compatible changes to proxy service interfaces requires complete offline deployment. To install the new version, follow the procedure below while all servers are operational:

1. Quiesce all inbound messages.
2. Confirm all asynchronous backlogged messages have been processed.
3. Make the necessary changes in the proxy service, and test to verify the proxy service operates as required.
4. Resume accepting inbound messages.

For more information about backward compatibility and installation strategies, see [“Installing a New Version of a Proxy Service” on page 2-8](#).

# Understanding AquaLogic Service Bus High Availability

A clustered AquaLogic Service Bus domain provides high availability. A highly available deployment has recovery provisions in the event of hardware or network failures, and provides for the transfer of control to a backup component when a failure occurs.

The following sections describe clustering and high availability for a AquaLogic Service Bus deployment:

- [About AquaLogic Service Bus High Availability](#)
- [AquaLogic Service Bus Failure and Recovery](#)

## About AquaLogic Service Bus High Availability

For a cluster to provide high availability, it must be able to recover from service failures. WebLogic Server supports failover for clustered objects and services pinned to servers in a clustered environment. For information about how WebLogic Server handles such failover scenarios, see [Communications in a Cluster](#) in *Using WebLogic Server Clusters*.

## Recommended Hardware and Software

The basic components of a highly available AquaLogic Service Bus environment include the following:

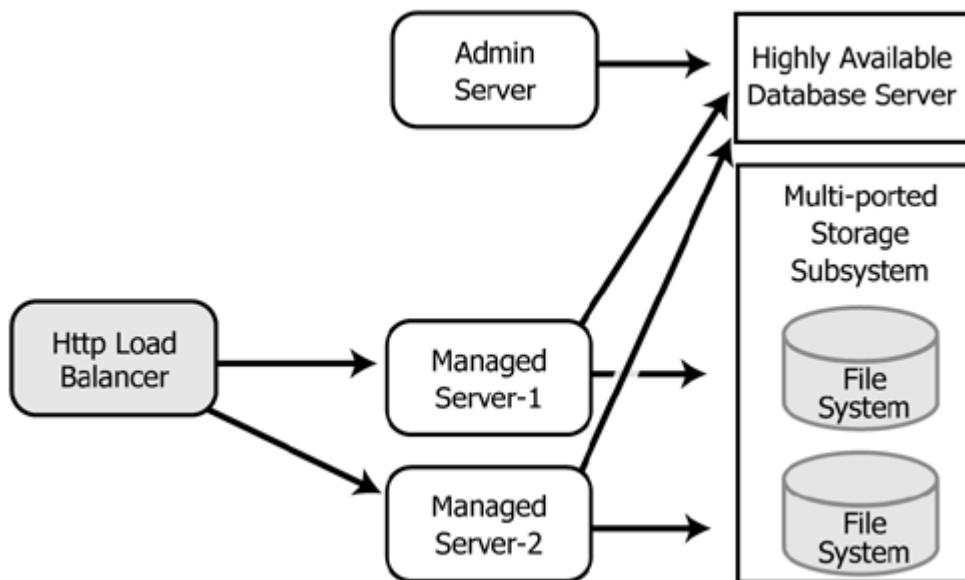
- An administration server
- A set of managed servers in a cluster

- An HTTP load balancer (router)
- Physically shared, highly-available disk subsystems for managed server data—Whole server migration requires that *all* data on a managed server be located on a multi-ported disk. A typical and recommended way to do this is by using a multi-ported disk subsystem or SAN, and allowing two or more servers to mount file systems within the disk subsystem. The file system does not need to be simultaneously shared; it is only necessary for one server to mount a file system at any one time.
- A Microsoft SQL Server or Oracle database configured for failover with a cluster manager—You should take advantage of any high availability or failover solutions offered by your database vendor in addition to using a commercial cluster manager. (For database-specific information, see your database vendor’s documentation.)

**Note:** For information about availability and performance considerations associated with the various types of JDBC drivers, see “Configure Database Connectivity” in [Configure JDBC](#) in WebLogic Server Administration Console Online Help.

A full discussion of how to plan the network topology of your clustered system is beyond the scope of this section. For information about how to fully utilize inbound load balancing and failover features for your AquaLogic Service Bus configuration by organizing one or more WebLogic Server clusters in relation to load balancers, firewalls, and Web servers, see [Cluster Architectures](#) in *Using WebLogic Server Clusters*. For information on configuring outbound load balancing, see “To Add a Business Service - Transport Configuration” in “Adding a Business Service” in [Business Services](#) in *Using the AquaLogic Service Bus Console*.

For a simplified view of a cluster, showing the http load balancer, highly available database and multi-ported file system, see the following figure.

**Figure 5-1 Simplified View of a Cluster**

## Regarding JMS File Stores

The default AquaLogic Service Bus domain configuration uses a file store for JMS persistence to store collected metrics for monitoring purposes and alerts. The configuration shown relies on a highly available multi-ported disk that can be shared between managed servers to optimize performance. This will typically be more performant than a JDBC store.

For information about configuring JMS file stores, see [Using the WebLogic Persistent Store](#) in *Configuring WebLogic Server Environments*.

## What Happens When a Server Fails

A server can fail due to either software or hardware problems. The following sections describe the processes that occur automatically in each case and the manual steps that must be taken in these situations.

## Software Faults

If a software fault occurs, the Node Manager (if configured to do so) will restart the WebLogic Server. For more information about Node Manager, see [Using Node Manager to Control Servers](#) in *Managing Server Startup and Shutdown*. For information about the steps to take to prepare for recovering a secure installation, see “Directory and File Back Ups for Failure Recovery” in [Avoiding and Recovering from Server Failure](#) in *Managing Server Startup and Shutdown*.

## Hardware Faults

If a hardware fault occurs, the physical machine may need to be repaired and could be out of operation for an extended period. In this case, the following events occur:

- The HTTP load balancer will detect the failed server and will redirect to other managed servers. (The actual algorithm for doing this will depend on the vendor for the http load balancer.)
- All new internal requests will be redirected to other managed servers.
- All in-flight transactions on the failed server are terminated.
- JMS messages that are already enqueued are automatically migrated with the whole server migration functionality. For more information, see [“Server Migration” on page 5-4](#).
- If your AquaLogic Service Bus configuration includes one or more proxy services that use File, FTP or Email transports with transport pollers pinned to a managed server that failed, you must select a different managed server in the definition of each of those proxy services in order to resume normal operation.
- If the managed server that hosts the WLI Aggregator application fails, collected metrics for monitoring and alerts are not automatically migrated. You must perform a manual migration. For more information, see [“Server Migration” on page 5-4](#).

## Server Migration

AquaLogic Service Bus leverages WebLogic Server’s whole server migration functionality to enable transparent failover of managed servers from one system to another. For detailed information regarding WebLogic Server whole server migration, see the following topics in the WebLogic Server documentation set:

- [Failover and Replication in a Cluster](#) in *Using WebLogic Server Clusters*
- [Avoiding and Recovering from Server Failure](#) in *Managing Server Startup and Shutdown*



## Message Reporting Purger

The Message Reporting Purger for the JMS Message Reporting Provider is deployed in a single managed server in a cluster (see [Appendix B, “AquaLogic Service Bus Deployment Resources”](#)).

If the managed server that hosts the Message Reporting Purger application fails, you must select a different managed server for the Message Reporting Purger and its associated queue (`wli.reporting.purge.queue`) to resume normal operation.

Any pending purging requests in the managed server that failed are not automatically migrated. You must perform a manual migration. Otherwise, target the Message Reporting Purger application and its queue to a different managed server and send the purging request again.

## AquaLogic Service Bus Failure and Recovery

In addition to the high availability features of WebLogic Server, AquaLogic Service Bus has failure and recovery characteristics that are based on the implementation and configuration of your AquaLogic Service Bus solution. The following sections discuss specific AquaLogic Service Bus failure and recovery topics:

- [Transparent Server Reconnection](#)
- [EIS Instance Failover](#)

## Transparent Server Reconnection

AquaLogic Service Bus provides transparent reconnection to external servers and services when they fail and restart. If AquaLogic Service Bus sends a message to a destination while the connection is unavailable, you may see one or more runtime error messages in the server console.

Transparent reconnection is provided for the following types of servers and services:

- SMTP
- JMS
- FTP
- DBMS
- Business services

AquaLogic Service Bus Console also provides monitoring features that enable you to view the status of services and to establish a system of SLAs and alerts to respond to service failures. For more information, see [Monitoring](#) in *Using the AquaLogic Service Bus Console*.

## EIS Instance Failover

Most business services in production environments will be configured to point to a number of EIS instances for load balancing purposes and high availability. If you expect that an EIS instance failure will have an extended duration or a business service points to a single, failed EIS instance, you can reconfigure the business service to point at an alternate, operational EIS instance. This change can be made dynamically.

For information about using the AquaLogic Service Bus Console to change an endpoint URI for a business service, see “Viewing and Changing Business Services” in [Business Services](#) in *Using the AquaLogic Service Bus Console*.

# Using the AquaLogic Service Bus Deployment API

Resources within a domain use Java Management Extensions (JMX) Managed Beans (MBeans) to expose their management functions. An MBean is a concrete Java class that is developed per JMX specifications. It can provide getter and setter operations for each management attribute within a managed resource along with additional management operations that the resource makes available.

The AquaLogic Service Bus `DeploymentMBean` enables programmatic import and export of AquaLogic Service Bus configurations, as well as modification of environment-specific configuration information. Using the `DeploymentMBean` interface, you can create Java programs and WLST scripts to automate promotion of AquaLogic Service Bus configurations from development environments through testing, staging, and finally to production environments.

The following sections describe how to use the `DeploymentMBean` to perform these deployment activities from within a WLST script:

- [Importing and Exporting Configurations](#)
- [Updating Environment-Specific Information](#)

For reference material on the `DeploymentMBean` interface and Java usage examples, see the [com.bea.wli.management](#) package in the Javadoc for AquaLogic Service Bus Classes.

## Importing and Exporting Configurations

AquaLogic Service Bus configurations are created using the AquaLogic Service Bus Console, and are stored through export in `.jar` files. Once a configuration `.jar` file has been exported, you can promote the configuration by importing it into a different AquaLogic Service Bus

domain and changing the environment-specific values in the configuration to match those of the new environment.

The methods in the `DeploymentMBean` interface directly parallel interactive features provided in the AquaLogic Service Bus Console, and require execution in the same order as their GUI counterparts. The following table lists the required steps and the methods available in the `DeploymentMBean` interface to perform them.

**Table A-1** Deployment steps and corresponding methods

To...	Use...
1. Create a session.	<code>openImportSession()</code>
2. Import all or part of the configuration .jar file.	<code>importIntoSession()</code>
3. Update environment-specific information to match the values required in the domain.	<code>findAndReplaceEnvValues()</code>
4. Verify that there are no conflicts in your configuration.	<code>isSessionReadyForCommit()</code>
5. Activate the session.	<code>commitImportSession()</code>
6. When you are ready to promote the configuration to another domain, export the configuration.	<code>export()</code> or <code>exportAll()</code>

The following example shows how to use the `DeploymentMBean` to perform these activities from within a WLST script.

**Note:** The following JAR files must be on the classpath: `sb-public.jar` and `sb-internal.jar`.

**Listing A-1** Importing, Updating, and Exporting a Configuration Using WLST

```
import wlstModule
from java.io import File
from java.util import List
from com.bea.wli.config import Ref
from com.bea.wli.config import TypeIds

def connectToServer():
    connect("weblogic", "weblogic", "t3://localhost:7001")
```

```

domainRuntime()

# imports a configuration jar file
def importConfig(jarfilename):
    infile = File(jarfilename)
    infile = infile.getAbsoluteFile() # enables server to find the file

    # find the deployment mbean
    mbean = findService("Deployment", "com.bea.wli.management.DeploymentMBean")

    # all changes to Service Bus configuration must be done in a session
    session = mbean.openImportSession()

    # import the whole jar file into the session.
    mbean.importIntoSession(session, infile)

    # typically you will change certain values that are environment dependent
    # here we are changing "localhost:7001" to "productionserver:7001"
    mbean.findAndReplaceEnvValues(session, TypeIds.URI_ENV_VALUE_TYPE,
    "localhost:7001", "productionserver:7001")

    # finally commit the session. The changes will be submitted to the
    # core state and the session will be destroyed
    mbean.commitImportSession(session)

# exports all resources in
def exportConfig(jarfilename, projectname, exportDependencies):
    outfile = File(jarfilename)
    outfile = outfile.getAbsoluteFile() # enables server to find the file

    # find the deployment mbean
    mbean = findService("Deployment", "com.bea.wli.management.DeploymentMBean")

    # export all resources in the given project. If exportDependencies
    # is true, any other resources that these resources depend on will
    # also be exported. If false, only the resources in the given project
    # will be exported.
    mbean.export(outfile, [projectname], [], exportDependencies)

# call the defined functions here.
try:
    # connect to running server
    connectToServer()

    # import configuration
    importConfig(sys.argv[1])

    # export all resources in the default project and their dependencies
    exportConfig("temp.jar", "default", true)

```

```
except:
    print "Unexpected error: ", sys.exc_info()[0]
    dumpStack()
    raise
```

---

In this example, the full contents of one configuration file are imported. One or more service endpoints had been set to `localhost:7001` in the environment from which the configuration had been exported. This example uses `findAndReplaceEnvValues()` to set each service endpoint having the value of `localhost:7001` to `productionserver:7001`. The change affects any matching endpoints in both proxy service and business service configurations. The updated configuration is committed, and then exported into another configuration file.

For reference material on the `DeploymentMBean` interface, see the [com.bea.wli.management](#) package in the Javadoc for AquaLogic Service Bus Classes.

For more information on creating WLST scripts, see [WebLogic Scripting Tool](#).

## Updating Environment-Specific Information

The `findAndReplaceEnvValues()` method enables you to update the value of the following artifacts to the appropriate environment-specific value:

- endpoints in proxy and business service configurations
- directory elements in File, Email, and FTP transport configurations

When using the `findAndReplaceEnvValues()` method, you indicate the type of artifact you want to update with the `envValueType` parameter. As shown in [Listing A-1](#), you use `TypeIds.URI_ENV_VALUE_TYPE` as the `envValueType` parameter value for search and replacement of service endpoint URIs. To update directory elements (such as error or archive directories), you use `TypeIds.FILE_PATH_ENV_VALUE_TYPE` as the `envValueType` parameter value for search and replacement of fully qualified directory paths.

The following code excerpt shows an example of updating directory values.

#### Listing A-2

---

#### Listing A-3 Updating directory values

---

```
...  
mbean.findAndReplaceEnvValues(session, TypeIds.FILE_PATH_ENV_VALUE_TYPE,  
"c:/myArchiveDirectory", "d:/production/ArchiveDirectory")  
...
```

---

You must update your security configuration and all other environment-specific settings interactively using the AquaLogic Service Bus Console. For information on configuring security, see [Securing Inbound and Outbound Messages](#) in the *BEA AquaLogic Service Bus Console Online Help*. For information on configuring other environment-specific settings, see “[Step 4. Deploy an AquaLogic Service Bus Configuration](#)” on page 2-6.





# AquaLogic Service Bus Deployment Resources

This section describes the AquaLogic Service Bus deployment resources, including targeting information for deployment in a cluster configuration. The table below contains the following information about each resource:

- **Resource Target**—The target for a resource depends on the resource’s type.
  - For queues and topics, the target is a single, migratable JMS server (`wlsbjmsServer`), a JMS server on the administration server, or a distributed destination.

For information about migratable JMS servers and distributed destinations, see “Configuring Migratable Targets for JMS Servers” and “Configuring WebLogic JMS Clustering” in [Configuring Clustered WebLogic JMS Resources](#) in *Configuring and Managing WebLogic JMS*.

- For all other resources (including JMS and JDBC services) the target is either the administration server or a cluster.
- **Resource**—The name of the resource as shown in the WebLogic Server Administration Console and the individual package or service (in a resource group).

Note that some resources contain abbreviations that are a legacy from prior BEA product releases:

- `wl` corresponds to WebLogic
  - `wli` corresponds to WebLogic Integration
- **Deployment Order**—A numerical value that indicates when this resource is deployed during startup, relative to other deployable resources on a server. Resources with lower

values are deployed before those with higher values. The deployment order value appears in this table only for those resources to which it is applicable.

- Administration Console Navigation—Route through the WebLogic Server Administration Console navigation tree to the specified package or service. All resources can be viewed and modified in the WebLogic Server Administration Console.

**Table B-1 AquaLogic Service Bus Deployment Resources**

Resource Target	Resource	Deployment Order	Administration Console Navigation
Admin Server	ServiceBus_Console webapp AquaLogic Service Bus Console Web application	810	<i>Domain</i> → Deployments→ ServiceBus_Console
	ALSB UDDI Manager weblogic90/servicebus/lib/uddiEar Provides UDDI Import and Publish capabilities to the AquaLogic Service Bus Console via a JMX interface.	122	<i>Domain</i> → Deployments→ ALSB UDDI Manager
	ALSB Configuration System weblogic90/integration/common/wliconfig.ear	100	<i>Domain</i> → Deployments→ ALSB Configuration System
	ALSB Logging weblogic90/servicebus/lib/stages/xbuslogging.war AquaLogic Service Bus logging application	430	<i>Domain</i> → Deployments→ ALSB Logging
	ALSB Publish weblogic90/servicebus/lib/stages/xbuspublish.war AquaLogic Service Bus message publishing application	420	<i>Domain</i> → Deployments→ ALSB Publish
	ALSB Routing weblogic90/servicebus/lib/stages/xbusrouting.war AquaLogic Service Bus message routing application	400	<i>Domain</i> → Deployments→ ALSB Routing

**Table B-1 AquaLogic Service Bus Deployment Resources (Continued)**

Resource Target	Resource	Deployment Order	Administration Console Navigation
Admin Server (continued)	ALSB Transform weblogic90/servicebus/lib/stages/xbustransform.war AquaLogic Service Bus message transformation application	410	Domain→ Deployments→ ALSB Transform
	Xbus Kernel weblogic90/servicebus/lib/kernelEar	250	Domain→ Deployments→ ALSB Kernel
	kerneladmin AquaLogic Service Bus kernel administration EJB		Domain→ Deployments→ ALSB Kernel→ Modules
	httpTransport.war AquaLogic Service Bus HTTP transport application		Domain→ Deployments→ ALSB Kernel→ Modules
	FileStore rmfilestore/FileStore JMS file store that stores persistent messages		Domain→ Services→ JDBC→ Persistent Stores
	wlsbJMSServer		Domain→ Services→ Messaging→ JMS Servers
	weblogic.wlsb.jms.transporttask.QueueConnectionFactory		Domain→ Services→ Messaging→ JMS Modules→ jmsResources

**Table B-1 AquaLogic Service Bus Deployment Resources (Continued)**

Resource Target	Resource	Deployment Order	Administration Console Navigation
Admin Server (continued)	wli.reporting.jmsprovider.NonXAConnectionFactory  Connection factory used by reporting provider for transacted sessions		Domain→ Services→ Messaging→ JMS Modules→ jmsResources
	wli.reporting.jmsprovider.XAConnectionFactory  Connection factory used by reporting provider for non-transacted sessions		Domain→ Services→ Messaging→ JMS Modules→ jmsResources
	wli.reporting.jmsprovider_error.queue  For errors associated with the message reporting queue: wli.reporting.jmsprovider.queue		Domain→ Services→ Messaging→ JMS Modules→ jmsResources
	ALSB Test Framework <sup>1</sup> weblogic90/integration/common/sbTestFrameworkEjb.ear  The test console application, accessible by users in the following AquaLogic Service Bus roles: IntegrationAdmin and IntegrationDeployer	910	Domain→ Deployments→ ALSB Test Framework

**Table B-1 AquaLogic Service Bus Deployment Resources (Continued)**

<b>Resource Target</b>	<b>Resource</b>	<b>Deployment Order</b>	<b>Administration Console Navigation</b>
Managed Servers in a Cluster	Email Transport Provider	152	<i>Domain</i> → Deployments→ Email Transport Provider
	weblogic90/servicebus/lib/email_transport.ear		
	emailtransport.jar		<i>Domain</i> → Deployments→ Email Transport Provider→ Modules
	PolledMessageListenerMDBEJB	153	<i>Domain</i> → Deployments→ Email Transport Provider→ EJBs
	Listens on wlsb.internal.transport.task.queue.email. Creates the job objects corresponding to each mail message, puts them in the task queue, and processes them.		
	FileTransport Provider		<i>Domain</i> → Deployments→ File Transport Provider
	weblogic90/servicebus/lib/file_transport.ear		
	filepoll.jar		<i>Domain</i> → Deployments→ File Transport Provider→ Modules
	PolledMessageListenerMDBEJB		<i>Domain</i> → Deployments→ File Transport Provider→ EJBs
	Listens on wlsb.internal.transport.queue.file. Creates the job objects corresponding to each new file, puts them in the task queue, and processes them.		

Table B-1 AquaLogic Service Bus Deployment Resources (Continued)

Resource Target	Resource	Deployment Order	Administration Console Navigation
Managed Servers in a Cluster (continued)	FTP Transport Provider	151	<i>Domain</i> → Deployments→ FTP Transport Provider
	weblogic90/servicebus/lib/ftp_transport.ear		
	ftptransport.jar		<i>Domain</i> → Deployments→ FTP Transport Provider→ Modules
	PolledMessageListenerMDBEJB Listens on wlsb.internal.transport.task.queue.ftp. Creates the job objects corresponding to each mail message, puts them in the task queue, and processes them.		<i>Domain</i> → Deployments→ FTP Transport Provider→ EJBs
	WLI Aggregator <sup>2</sup> weblogic90/integration/common/wliagggregator.ear AquaLogic Service Bus metrics aggregator <b>Note:</b> WLI Aggregator is targeted to a single managed server.	110	<i>Domain</i> → Deployments→ WLI Aggregator
	WLI Common weblogic90/integration/common/wlicommon.ear (BEA common framework components)	120	<i>Domain</i> → Deployments→ WLI Common
	ALSB Configuration System weblogic90/integration/common/wliconfig.ear	100	<i>Domain</i> → Deployments→ ALSB Configuration System

**Table B-1 AquaLogic Service Bus Deployment Resources (Continued)**

Resource Target	Resource	Deployment Order	Administration Console Navigation
Managed Servers in a Cluster (continued)	Message Reporting Purger (PurgingMDB) weblogic90/integration/common/msgpurger.ear  <b>Note:</b> Message Reporting Purger is targeted to a single managed server. <sup>3 4</sup>	126	<i>Domain</i> → Deployments→ Message Reporting Purger
	ALSB Resource weblogic90/servicebus/lib/sbresourceWar/sbresource.war  Used by the UDDI subsystem to expose resources (WSDLs, policies, schemas, and so on) contained in AquaLogic Service Bus.  You can undeploy ALSB Resource from the WebLogic Server Administration Console if you do not want the AquaLogic Service Bus resources to be available via HTTP. However, in doing so, you will no longer be able to use the UDDI subsystem.	500	<i>Domain</i> → Deployments→ ALSB Resource
	ALSB Logging weblogic90/servicebus/lib/stages/xbuslogging.war  AquaLogic Service Bus logging application	430	<i>Domain</i> → Deployments→ ALSB Logging
	ALSB Publish weblogic90/servicebus/lib/stages/xbuspublish.war  AquaLogic Service Bus message publishing application	420	<i>Domain</i> → Deployments→ ALSB Publish
	ALSB Routing weblogic90/servicebus/lib/stages/xbusrouting.war  AquaLogic Service Bus message routing application	400	<i>Domain</i> → Deployments→ ALSB Routing

Table B-1 AquaLogic Service Bus Deployment Resources (Continued)

Resource Target	Resource	Deployment Order	Administration Console Navigation
Managed Servers in a Cluster (continued)	ALSB Transform	410	Domain→ Deployments→ ALSB Transform
	weblogic90/servicebus/lib/stages/xbustransform.war AquaLogic Service Bus message transformation application		
Distributed Destination	JMS Reporting Provider	125	Domain→ JMS Reporting Provider
	weblogic90/integration/common/jmsreportprovider.ear  wlsbJMSServer_auto_1-n (AquaLogic Service Bus JMS server on each managed server)		
			Domain→ Services→ Messaging→ JMS Servers



**Table B-1 AquaLogic Service Bus Deployment Resources (Continued)**

Resource Target	Resource	Deployment Order	Administration Console Navigation
Distributed Destination (continued)	QueueIn Resource for internal use		<i>Domain</i> → <i>Services</i> → <i>Messaging</i> → <i>JMS Modules</i> → <i>jmsResources</i>
	QueueIn.Quota Resource for internal use		
	TemporaryTmplt Resource for internal use		
	wli.reporting.jmsprovider.queue Queue for reporting messages		
	wli.reporting.purge.queue JMS queue for the Message Reporting Purger; deployed in a single managed server in a cluster <sup>3 4</sup>		
	wlsb.internal.transport.poller.queue.email Polling interval timeout event queue for email transport		
	wlsb.internal.transport.poller.queue.file Polling interval timeout event queue for file transport		
	wlsb.internal.transport.poller.queue.ftp Polling interval timeout event queue for FTP transport		
	wlsb.internal.transport.task.queue.email Queue for jobs associated with each new email message detected during polling		

**Table B-1 AquaLogic Service Bus Deployment Resources (Continued)**

Resource Target	Resource	Deployment Order	Administration Console Navigation
Distributed Destination (continued)	wlsb.internal.transport.task.queue.file		Domain→ Services→ Messaging→ JMS Modules→ jmsResources
	Queue for job associated with each new file detected during polling		
	wlsb.internal.transport.task.queue.ftp		
	(Queue for job associated with each new FTP message detected during polling)		
	ReportingMDB		Domain→ Deployments→ JMS Reporting Provider→ EJBs
	Listens to wli.reporting.jmsprovider.queue and asynchronously persists reporting messages to the JMS Reporting Provider Data Store. The reporting action configured in the proxy service pipeline logs messages to the reporting stream while utilizing the JMS reporting provider (wli.reporting.jmsprovider.queue) to write to the queue.		
	jmsreportprovider.jar		Domain→ Deployments→ JMS Reporting Provider→ Modules

1. In clusters, the Test Framework EJB is deployed on the admin server and on each managed server by default. It must be deployed on the admin server if you want to test XQueries from the XQuery builders. To use the test console, a cluster address must be configured. The cluster address must contain at least one of the managed servers where the Test Framework EJB is deployed.
2. For more information, see [“Step 4. Starting, Stopping, and Monitoring Managed Servers” on page 4-10](#)
3. The AquaLogic Service Bus Message Reporting Purger runs as a singleton instance on one of the managed servers in the cluster. The Message Reporting Purger queues the purging requests and processes them one by one.
4. For failover information in the event the managed server that hosts the Message Reporting Purger and the wli.reporting.purge.queue fails, see [“Message Reporting Purger” on page 5-5](#).

# Index

## A

- access control 2-6
- administration server
  - configuring 2-3, 4-6
  - deployment 3-5
  - IP address 4-2
  - resources targeted to B-2
  - Start Admin Server 2-5
- administrator username 2-3, 4-6
- Alert Manager 3-4
- ALSB Configuration System B-2, B-6
- ALSB Logging B-2, B-7
- ALSB Publish B-2, B-7
- ALSB Routing B-2, B-7
- ALSB Transform B-3, B-8
- AquaLogic Service Bus Data Aggregator 3-4
- AquaLogic Service Bus domains 3-2
- Archive directory 4-4
- authentication 2-6
- authorization 2-6
- Available JDBC Data Sources list 2-4

## B

- boot.properties 3-5
- business processes
  - load balancing 3-6
- business service
  - and WS-Policies 1-9
  - changing a business service 4-15
  - configuration 1-8
  - connection factories 2-5
  - transparent server reconnection 5-5

- updating for expanded cluster 4-14

## C

- CacheFullExceptions 1-5
- clusters
  - about clusters 1-5
  - adding servers to 4-2
  - configuration tasks 2-1, 4-1
  - designing 3-2
  - domains in 3-2, 3-3
  - machines in 4-6
  - prerequisites for configuring 2-1, 4-2
  - scalability 3-1
  - security 2-6, 4-9
  - simplified view of 5-3
  - updating address 4-12
  - updating business services after expansion 4-14
  - updating proxy services after expansion 4-14
- config.xml 2-5, 4-1, 4-5, 4-8, 4-9
- configuration
  - administrative username and password 2-3, 4-6
  - clusters 2-1, 4-1
  - deploy 2-8
  - import and export A-1
  - of administration server 2-3, 4-6
  - of business services 1-8
  - of Java SDK 2-3, 4-6
  - of JRockit SDK 2-3, 4-6
  - of machines in cluster 4-6

- of proxy services 1-8
  - of servers to machines 4-7
  - prerequisites 4-1
  - security 2-6, 4-9
  - server start mode 2-3, 4-6
  - using WLST script A-2
  - WSDL resources 1-7
- configuration files
  - boot.properties 3-5
  - config.xml 2-5, 4-1, 4-8, 4-9
  - msi-config.xml 3-5
  - SerializedSystemIni.dat 3-5
  - web.xml 4-13
- Configuration Wizard 2-2, 4-5
- connection factories
  - updating in proxy service 1-8
  - weblogic.wlsb.jms.transporttask.QueueCon
    - nectionFactory B-3
  - wli.reporting.jmsprovider.NonXAConnecti
    - onFactory B-4
  - wli.reporting.jmsprovider.XAConnectionFa
    - ctory B-4
  - XA and non-XA 2-5
- connection pools 1-6

## D

- data sources 2-4, 4-7
- data transformations
  - formatted binary data 1-10
  - MFL 1-10
  - XML 1-10
  - XQuery 1-10
  - XSLT 1-10
- database administrators 1-3
- database tables 4-9
- DataMirror 2-1
- DB Version drop-down list 2-4
- DBMS 1-10, 2-1, 5-5
- deployment
  - and administration server 3-5

- goals 1-1
- JAR file 2-6, 4-11
- resources
  - business services 1-8
  - connection factories 1-8
  - data transformation maps 1-10
  - databases 1-10
  - hardware 1-10
  - JMS queues 1-8
  - network 1-10
  - online update 2-8
  - operating system 1-10
  - overview 1-4
  - proxy service providers 1-9
  - proxy services 1-8
  - resource groups 3-3
  - service accounts 1-9
  - singleton 3-3
  - WebLogic Server 1-4
  - WebLogic Server Work Manager 1-9
  - WSDLs 1-7
  - WS-Policies 1-9
  - XML schemas 1-9
- specialists 1-3
- tasks 1-2, 2-1, 4-1
- DeploymentMBean A-1, A-2
- digital signature 1-9
- directories
  - Archive 4-4
  - Error 4-4
  - Stage 4-4
  - updating in proxy services 1-8, 2-7
  - updating using WLST A-4
- dispatch policies 1-9
- dist\_wli.reporting.jmsprovider.queue\_auto 4-13
- dist\_wlsb.internal.transport.task.queue.email\_au
  - to 4-13
- dist\_wlsb.internal.transport.task.queue.file\_auto
  - 4-13
- dist\_wlsb.internal.transport.task.queue.ftp\_auto
  - 4-13

- distributed destinations B-8, B-9, B-10
- domains
  - adding managed server to 2-7, 4-11
  - adding proxy servers to 4-8
  - AquaLogic Service Bus 3-2
  - clustered servers in 3-3
  - Configuration Wizard, using the 2-2, 4-5
  - creating 2-1, 2-2, 3-2, 4-1, 4-5
  - creating using Configuration Wizard 2-2, 4-5
  - dropping managed server from 4-15
  - expanding cluster 4-14
  - naming 2-5, 4-8
  - shutting down servers in 4-10
  - source 2-3, 4-6
  - starting servers in 2-6, 4-10, 4-11
  - updating 2-7, 4-11, 4-15
- dynamic update 2-8

## E

### EIS

- and business services 1-8
- changing a business service 2-7
- failover 5-6

### EJBs

- cache 1-5
- DeploymentMBean A-1, A-2
- kerneladmin B-3
- Message Reporting Purger B-7
- parameters
  - max-beans-in-free-cache 1-5
  - max-beans-in-free-pool 1-5
- PolledMessageListenerMDBEJB B-5, B-6
- pools 1-5
- PurgingMDB B-7
- ReportingMDB B-10

### Email

- Email Transport Provider B-5
- email\_transport.ear B-5
- message type 1-8

- poller 3-4, 4-15, 5-4
- transport load balancing 4-4

- encryption 1-9

- endpoint URIs 2-7, 4-14, A-4

- Error directory 4-4

- execution thread pool 1-6

## F

- failover 5-6

- File message type 1-8

- File poller 3-4, 4-15, 5-4

- file stores 2-4, 4-8, 5-3

- file system 2-1, 4-2

- File Transport Provider B-5

- FileStore B-3

- findAndReplaceEnvValues A-4

- firewalls 4-4, 4-8

- formatted binary data 1-10

### FTP

- message type 1-8

- poller 3-4, 4-15, 5-4

- transparent reconnection 5-5

- transport load balancing 4-4

## G

- goals 1-1

## H

- hardware

- faults 5-4

- requirements 1-10

- router 4-4

- high availability

- about high availability 5-1

- and JDBC 5-3

- and JMS file stores 5-3

- JMS 3-6

### HTTP

- functions 3-6

- load balancer 4-13
- load balancing 3-6
- router 4-6

HttpClusterServlet 3-6, 4-6

## I

IP addresses 2-1, 4-2, 4-3

## J

J2EE Connector Architecture (J2EE-CA) 1-7

J2EE Connector Architecture *See* JCA

JAR file

- deploying 4-11

- exporting and importing 2-6

Java Management Extensions A-1

Java Message Service (JMS) 1-5

Java SDK 2-3, 4-6

JCA 1-7

JDBC

- and high availability 5-3

- connection pools 1-6

- data sources 2-4, 4-7

- jdbc.log 4-7

- wlsjmsrpDataSource 4-7

jdbc.log 2-4, 4-7

JMS

- connection factories 2-5

- file stores 2-4, 4-8, 5-3

- functions 3-6

- high availability 3-6

- jmsResources module 4-12

- message migration 5-4

- modules 2-5

- rmfilestore 4-8

- transport 5-5

- WebLogic JMS 1-5

JMS Reporting Provider 1-10, 2-1, 2-4, 4-7, B-8

jmsreportprovider.jar B-10

JRockit SDK 2-3, 4-6

## K

kerneladmin B-3

keystores 2-6, 4-9

## L

license

- cluster 2-1, 4-2

- PointBase 2-1, 4-4

listen port numbers 4-2

load balancing

- business processes 3-6

- Email transport 4-4

- File transport 4-4

- FTP transport 4-4

- HTTP 4-13, 5-4

- HTTP functions 3-6

- JMS functions 3-6

- router 4-4

- settings 2-7

- WebLogic Server 3-6

logging B-2, B-7

Logging last resource option 2-4, 4-7

## M

managed servers

- adding to domain 2-7, 4-11

- dropping from domain 4-15

- Managed Server drop-down list 1-8

- Managed Server Independence (MSI) mode 3-5

- migration of 5-4

- resources targeted to B-5, B-6, B-7, B-8

- selecting for File, FTP, and Email message types in cluster 1-8

- shutting down 4-10

- starting 2-6, 3-5, 4-10, 4-11

max-beans-in-free-cache 1-5

max-beans-in-free-pool 1-5

message flows 1-8

- Message Reporting Purger 4-15, B-7
- message types
  - Email 1-8
  - File 1-8
  - FTP 1-8
- message-level security 2-6
- MFL 1-10
- Microsoft SQL Server 2-4, 4-3, 4-7
- Module Containing Temporary Template list 4-12
- msgpurger.ear 4-10
- msi-config.xml 3-5
- multicast addresses 2-1, 4-2
- multihome machine 2-1, 4-2

## N

- NFS 4-4
- non-XA connection factories 2-5

## O

- online update 2-8
- Oracle 2-4, 4-3, 4-7

## P

- password configuration 2-3, 4-6
- PointBase 2-1, 4-4
- PolledMessageListenerMDBEJB B-5, B-6
- pollers
  - Email 3-4
  - File 3-4
  - FTP 3-4
- pool size 1-5
- port numbers 2-1, 4-2
- Production Mode 2-3, 4-6
- proxy servers
  - adding to domain configuration 4-8
  - Proxy Server drop-down list 4-6
- proxy service
  - and proxy service providers 1-9

- and service accounts 1-9
- and WebLogic Server Work Manager 1-9
- and WS-Policies 1-9
- backward compatibility 2-8
- configuration 1-8
- connection factories 2-5
- directories
  - Archive 4-4
  - Error 4-4
  - Stage 4-4
- Email message type 1-8
- File message type 1-8
- FTP message type 1-8
- installing new version 2-8, 4-16
- message flows 1-8
- schemas 1-9
- updating
  - connection factories for 1-8
  - directory names 1-8
  - for expanded cluster 4-14
  - Managed Server selection 1-8
  - queues for 1-8
  - when dropping a managed server 4-15
- proxy service providers
  - digital signature 1-9
  - encryption 1-9
  - SSL client authentication 1-9
  - Web services security X509 token 1-9
- PurgingMDB B-7

## Q

- QueueIn B-9
- QueueIn.Quota B-9
- queues
  - and business services 2-5
  - and proxy services 2-5
  - dist\_wli.reporting.jmsprovider.queue\_auto 4-13
  - dist\_wlsb.internal.transport.task.queue.email\_auto 4-13

- dist\_wlsb.internal.transport.task.queue.file\_
  - auto 4-13
- dist\_wlsb.internal.transport.task.queue.ftp\_
  - auto 4-13
- load balancing 3-6
- QueueIn B-9
- QueueIn.Quota B-9
- updating in proxy service 1-8
- wli.reporting.jmsprovider.queue B-4, B-9
- wli.reporting.jmsprovider\_error.queue B-4
- wli.reporting.purge.queue B-9
- wlsb.internal.transport.poller.queue.email
  - B-9
- wlsb.internal.transport.poller.queue.file B-9
- wlsb.internal.transport.poller.queue.ftp B-9
- wlsb.internal.transport.task.queue.email B-9
- wlsb.internal.transport.task.queue.file B-10
- wlsb.internal.transport.task.queue.ftp B-10

## R

- recovery
  - from hardware faults 5-4
  - from software faults 5-4
- redploy configuration 2-8
- ReportingMDB B-10
- resource naming rules 2-2
- rmfilestore 2-4, 4-8, B-3
- roles
  - database administrators 1-3
  - deployment specialists 1-3
  - WebLogic Server administrators 1-3
- router 3-6, 4-4, 4-6
- routing destinations 2-7
- Run Scripts 2-4

## S

- schemas 1-9
- security 4-9
  - access control 2-6
  - authentication 2-6

- authorization 2-6
- configuring in clusters 2-6, 4-9
- digital signature 1-9
- encryption 1-9
- message level 2-6
- proxy service providers 1-9
- service accounts 1-9
- SSL client authentication 1-9
- transport level 2-6
- Web services security X509 token 1-9
- WS-Policies 1-9
- SerializedSystemIni.dat 3-5
- server affinity 3-6
- servers
  - adding to domain 2-7, 4-11
  - and deployment 3-5
  - dropping from domain 4-15
  - failure and recovery 5-3
  - in domains 3-3
  - migration 5-4
  - monitoring 4-10
  - multiple instances on single machine 4-3
  - shutting down in the domain 4-10
  - start mode 2-3, 4-6
  - starting in the domain 2-6, 4-10, 4-11
  - transparent reconnection 5-5
  - See also* administration servers, managed servers
- service accounts 1-9
- Service Level Agreements 3-4
- service references 2-7
- shared file system 2-1, 4-2, 4-4
- shutting down servers 4-10
- SLA Manager 3-4
- SMTP 5-5
- SOAP 3-6
- software faults 5-4
- software router 4-4
- SSL client authentication 1-9, 4-9
- Stage directory 4-4
- Start Admin Server 2-5



- starting servers 2-6, 4-10, 4-11
- Sun SDK 4-6
- Supports global transactions check box 2-4, 4-7

## T

- Temporary Template Name box 4-12
- TemporaryTplt 4-12, B-9
- Test Connections 2-4
- threads, execution 1-6
- transparent reconnection 5-5
- transport-level security 2-6
- TypeIds.FILE\_PATH\_ENV\_VALUE\_TYPE A-4
- TypeIds.URI\_ENV\_VALUE\_TYPE A-4

## U

- UDDI Manager B-2

## W

- Web services security X509 token 1-9
- web.xml 4-13
- webapp B-2
- WebLogic Messaging Bridge 2-2
- WebLogic Server Administration Console
  - location in cluster 3-3
  - Servers node 3-3
- WebLogic Server administrators 1-3
- WebLogic Server Work Manager 1-9
- weblogic.wlsb.jms.transporttask.QueueConnecti  
onFactory B-3
- WLI Aggregator 4-15, 5-4, B-6
- WLI Common B-6
- wli.reporting.jmsprovider.NonXAConnectionFa  
ctory B-4
- wli.reporting.jmsprovider.queue B-4, B-9
- wli.reporting.jmsprovider.XAConnectionFactory B-4
- wli.reporting.jmsprovider\_error.queue B-4
- wli.reporting.purge.queue B-9

- wliagggregator.ear 3-4, 4-10
- wlsb.internal.transport.poller.queue.email B-9
- wlsb.internal.transport.poller.queue.file B-9
- wlsb.internal.transport.poller.queue.ftp B-9
- wlsb.internal.transport.task.queue.email B-9
- wlsb.internal.transport.task.queue.file B-10
- wlsb.internal.transport.task.queue.ftp B-10
- wlsbjmsrpDataSource 2-4, 4-7
- wlsbJMSServer 3-4, B-3, B-8
- WLST A-2
- WSDLs 1-7
- WS-Policies 1-9

## X

- XA connection factories 2-5
- XBus Kernel B-3
- Xbus Kernel B-3
- XML 1-9, 1-10, 3-6
- XQuery 1-10
- XSLT 1-10

