BEA AquaLogic Service Bus™

Deployment Guide

Version 2.6
Document Date: January 2007

Contents

1. Introduction

2. Configuring a Single-Server Deployment

3. Understanding AquaLogic Service Bus Clusters

4. Configuring a Clustered Deployment

5. Understanding AquaLogic Service Bus High Availability

A. Using the Deployment APIs

CHAPTER 1

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" on page 1-1
- "Key Deployment Tasks" on page 1-2
- "Roles in AquaLogic Service Bus Deployment" on page 1-2
- "Key Deployment Resources" on page 1-4

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 http://edocs.bea.com/alsb/docs26/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 completing 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 the *AquaLogic Service Bus Security 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" on page 1-3
- "WebLogic Server Administrators" on page 1-3
- "Database Administrators" on page 1-3

One person can assume multiple roles, and all roles are not equally relevant in all deployment scenarios. A successful deployment, however, 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 experience and expertise in the following areas:

- Hardware and platform knowledge
- 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 and expertise in the following areas:

- Hardware and platform knowledge
- 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. 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 WebLogic Server entities that can be protected from unauthorized access using security roles and security policies.

The key resources that may be modified at deployment time are:

- "WebLogic Server Resources" on page 1-4
- "AquaLogic Service Bus Configuration Resources" on page 1-7
- "Relational Database Management System Resources" on page 1-12
- "Hardware, Operating System, and Network Resources" on page 1-12

WebLogic Server Resources

This section provides general information about WebLogic Server resources that are most relevant to the deployment of an 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 configurable WebLogic Server features that are most relevant to AquaLogic Service Bus deployments are:

- "Clustering" on page 1-5
- "Java Message Service" on page 1-5
- "EJB Pooling and Caching" on page 1-5
- "JDBC Connection Pools" on page 1-6
- "Execution Thread Pool" on page 1-6
- "J2EE Connector Architecture" on page 1-7

Clustering

A cluster is 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. To increase workload capacity, you can run WebLogic Server on a cluster. 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 *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 *Configuring and Managing WebLogic JMS.*

EJB Pooling and Caching

The number of EJBs in an AquaLogic Service Bus deployment 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 `CacheFullExceptions`. 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" under 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 DBMS. To reduce the overhead associated with establishing database connections, WebLogic JDBC provides ready-to-use connection pools.

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 may result in delays while AquaLogic Service Bus waits for connections to become available. A setting that is too high may result in slower DBMS performance.

For more information about WebLogic JDBC connection pools, see:

- "How Connection Pools Enhance Performance" under Performance Tuning Your JDBC Application in *Programming WebLogic JDBC.*
- "Connection Pool Features" under 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 may result in sequential processing and possible deadlocks. A setting that is too high may result 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 to read sockets and can sometimes lead to a deadlock.

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

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" under Business Services in *Using the AquaLogic Service Bus Console*. For information on how to update the value of existing endpoints, see "Finding and Replacing Environment Values" in System Administration 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" under 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 drop-down list of managed servers appears in the AquaLogic Service Bus Console only for clustered AquaLogic Service Bus domains. For information on how to edit the Managed Server value for a proxy service, see “Viewing and Changing Proxy Services” under 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 System Administration 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” under Configuring JMS System Resources in *Using the AquaLogic Service Bus Console.*

- 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 *Using the AquaLogic Service Bus Console.*

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 configuring the service using the AquaLogic Service Bus Console. WSDL files used as the basis for defining services are stored as AquaLogic Service Bus resources. These resources are unlikely to require updating 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→**

Programs→BEA Products→Examples→AquaLogic Service Bus→Start Examples Server), you can view the WSDL for the `loangateway2` proxy service at `http://localhost:7021/crejws_basic_ejb/loangateway2?wsdl`.

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 updating 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 *AquaLogic Service Bus Security 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 *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 *AquaLogic Service Bus Security Guide*.

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

JARS

JARs (Java ARchive) are zipped files that contain a set of Java classes. They are used to store compiled Java classes and associated metadata that can constitute a program. JARs act as callable program libraries for Java code elements (so that a single compilation link provides access to multiple elements, rather than requiring bindings for each element individually). JARs are unlikely to require updating when deployed to a new environment.

Alert Destinations

An Alert Destination resource captures a list of recipients that can receive alert notifications from the AquaLogic Service Bus. In typical system monitoring contexts, alerts generated by AquaLogic Service Bus bear significance to a finite set of users. In AquaLogic Service Bus, each Alert Destination resource may be configured to include a set of recipients according to a given context. Alert Destinations are used by Alert actions configured in the message flow, and also by SLA alert rules. An Alert destination could include one or more of the following types of destinations: Console, Reporting Data stream, SNMP trap, E-mail, JMS queue, or JMS topic. In the case of E-mail and JMS destinations, a destination resource could include a list of E-mail addresses or JMS URIs, respectively. Alert Destinations are unlikely to require updating when deployed to a new environment.

UDDI Registries

A UDDI Registry resource is a global resource that stores information about UDDI Registries used by AquaLogic Service Bus. After the UDDI Registry resource is configured, you can then publish AquaLogic Service Bus proxy services to the associated registry, or import business services from the registry to be used by an AquaLogic Service Bus proxy service. You must be in an active session to configure the UDDI registry resource. UDDI Registry resources must be updated or reconfigured when deployed to a new environment. For more information on updating UDDI Registry resources during deployment, see "Best Practices to Follow When Migrating Global Resources" on page 1-11.

JNDI Providers

JNDI Provider resources are definitions of JNDI providers that describe the URL (or list of URLs in the case of clustered deployments) of the JNDI providers used by AquaLogic Service Bus. They are global resources and can be re-used across projects with an AquaLogic Service Bus domain. If the JNDI provider is secured, then the JNDI Provider resource description also carries a user name and password to gain access. JNDI Provider resources must be updated or reconfigured when deployed to a new environment. For more information on updating JNDI Provider resources during deployment, see "Best Practices to Follow When Migrating Global Resources" on page 1-11.

SMTP Servers

SMTP Server resources are definitions of SMTP Servers that describe the URL and port for the SMTP Servers used by the AquaLogic Service Bus. They are global resources and can be re-used across projects with an AquaLogic Service Bus domain. If the SMTP Server is secured, then the SMTP Server resource description also carries a user name and password to gain access. SMTP Server resources are used while configuring Alert Destination resources and E-mail transport-based Business Services. SMTP Server resources must be updated or reconfigured when deployed to a new environment. For more information on updating SMTP Server resources during deployment, see "Best Practices to Follow When Migrating Global Resources" on page 1-11.

Best Practices to Follow When Migrating Global Resources

Global resources include the UDDI Registry, JNDI Provider, and SMTP Server resources. These resources typically have different values in testing environments from those that will be used in staging or production environments. When you migrate a deployment from a test environment to a production environment, use one of the following methods:

1. Create new global resources on the production domain with the same names as in the test environment. Alternatively, import these resources directly from the test environment for the first time, and then customize them (rather than spend time creating them manually).

2. When exporting artifacts from the test environment, exclude the global resources. Then simply import the `config.jar` file. If there are any resources in the `config.jar` file that

references Alert Destinations, JNDI providers, or SMTP providers, these references will simply snap into place when the import is complete.

3. Alternatively, export all the artifacts (including the global resources), but exclude the global resources when importing the `config.jar` file .

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 *BEA AquaLogic Service Bus Release Notes*.

For additional information on tuning 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 *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" on page 2-1
- "Step 2. Prepare an AquaLogic Service Bus Domain" on page 2-2
- "Step 3. Configure AquaLogic Service Bus Security" on page 2-6
- "Step 4. Deploy an AquaLogic Service Bus Configuration" on page 2-6
- "Step 5. Update Your Domain as Your Production Environment Changes" on page 2-7

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.

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 *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" on page 2-2
- "Configuring JMS Resources" on page 2-5

Creating an AquaLogic Service Bus Domain Using the Configuration Wizard

You begin defining 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 (Weblogic Server is selected by default).
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	Provide a name for the admin server, a listen address, and a listen port (7001 by default). If you want to enable SSL for your configuration, select the SSL enabled check box, and a SSL listen port (7002 by default). **Note:** You can enable an SSL port, an HTTP cleartext port, or both ports for the administration server. In secure installations, the HTTP cleartext port can be disabled. However, when AquaLogic Service Bus is used in combination with a UDDI registry (for example, AquaLogic Service Registry), you must ensure that the server’s HTTP cleartext port is enabled.
Configure Managed Servers	Not applicable, because this is a single-server deployment.
Configure Machines	Not applicable, because this is a single-server deployment.

Table 2-1 Responses to Configuration Wizard Prompts

Configure JDBC Data Sources	Accept the default values for Name and JNDI name of `wlsbjmsrpDataSource`. Select one of the following to identify the database type and driver for the JMS Reporting Provider Data Store: • Oracle and BEA’s Oracle Driver (Type 4) Versions: 9.0.1, 9.2.0, 10 • MS SQL Server and BEA’s MS SQL Server Driver (Type 4) Versions: 7.0, 2000 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” under Configuring JDBC Data Sources in *Configuring and Managing WebLogic JDBC*. Enter your environment-specific database information in the remaining text boxes. For more information, see “Configure JDBC Data Sources” under Customizing Existing JDBC and JMS Settings in *Creating WebLogic Domains Using the Configuration Wizard*. 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.
Run Database Scripts	Select wlsbjmsrpDataSource from the Available JDBC Data Sources list. Select the version of the database from the DB Version drop-down list. Click Run Scripts. 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 `jdbc.log`. If you want the results written to a different file, click the Log File check box and specify the file. **Note:** 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 *`BEA_HOME`*`/weblogic92/integration/common/dbscripts`, where *`BEA_HOME`* is the directory in which you installed AquaLogic Service Bus.
Configure JMS File Stores	Accept the defaults for `rmfilestore` and add any file stores needed by proxy services or business services.
Review WebLogic Domain	Accept the default values.

Table 2-1 Responses to Configuration Wizard Prompts

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” under 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 *AquaLogic Service Bus Security 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 under the following topics in *Using the AquaLogic Service Bus Console*:
  - "Importing Configuration Data" in System Administration
  - "Finding and Replacing Environment Values" in System Administration
- 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 Deployment APIs."

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” under Project Explorer, in *Using the AquaLogic Service Bus Console*.

- Routing destinations

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

- Load balancing settings

  For information about load balancing, see “Viewing and Changing Business Services” under Business Services in *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” on page 2-7
- “Installing a New Version of a Proxy Service” on page 2-8
- “Online Configuration Updates” on page 2-8

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 5-5. For information about using the AquaLogic Service Bus Console to change an endpoint URI for a business service, see

"Viewing and Changing Business Services" under 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 dynamically make changes 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 dynamically change the configuration information for a system 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:

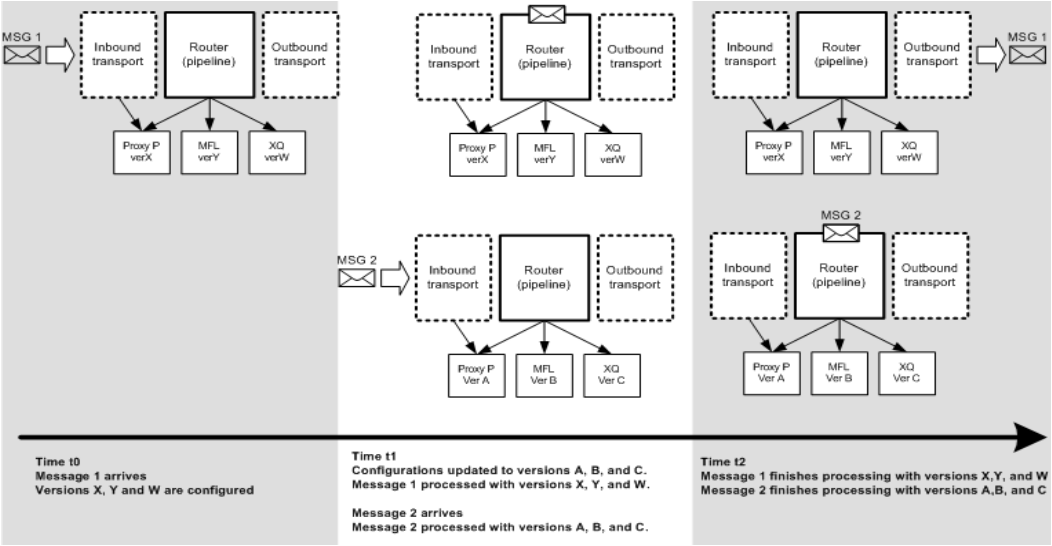
- To import and update a configuration manually, use the AquaLogic Service Bus Console as described in the following topics in the *Using the AquaLogic Service Bus Console*:
  - "Importing Configuration Data" in System Administration
  - "Finding and Replacing Environment Values" in System Administration
- 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 Deployment APIs."

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


MSG 1
Inbound transport
Router (pipeline)
Outbound transport
Proxy P verX
MFL verY
XQ verW
MSG 2
Proxy P Ver A
MFL Ver B
XQ Ver C
MSG 1
Time t0
Message 1 arrives
Versions X, Y and W are configured
Time t1
Configurations updated to versions A, B, and C.
Message 1 processed with versions X, Y, and W.
Message 2 arrives
Message 2 processed with versions A, B, and C.
Time t2
Message 1 finishes processing with versions X,Y, and W
Message 2 finishes processing with versions A,B, and C

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 when 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 to 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.

- 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.
- During online deployment, AquaLogic Service Bus checks whether the correct versions of referenced resources are used for message processing. If this is temporarily not true, an error is returned. However, if the interface artifact of an invoked service changes (for example: MFL, WSDL), the invoking proxy service may not return an error although it temporarily sees a version of the artifact that does not correlate with the proxy service version.

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" on page 3-1
- "Designing a Clustered Deployment" on page 3-2
- "Load Balancing in a AquaLogic Service Bus Cluster" on page 3-5
- "High Availability in an AquaLogic Service Bus Cluster" on page 3-6
- "Deploying Configurations" on page 3-7

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 an 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 dynamically increase capacity 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" on page 3-2
- "AquaLogic Service Bus Deployment Resources" on page 3-3
- "Load Balancing in a AquaLogic Service Bus Cluster" on page 3-5

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.

Note: Managed servers that start without an administration server operate in Managed Server Independence (MSI) mode. For information about MSI mode, see “Managed Server Independence Mode” in “Avoiding and Recovering From Server Failure” in *Managing Server Startup and Shutdown*.

You can enable an SSL port, an HTTP cleartext port, or both ports for the administration server. In secure installations, the HTTP cleartext port can be disabled. However, when AquaLogic Service Bus is used in combination with a UDDI registry (for example, AquaLogic Service Registry), you must ensure that the server’s HTTP cleartext port is enabled.

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” under 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 E-mail 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 Proxy Services in *Using the AquaLogic Service Bus Console*.
AquaLogic Service Bus Data Aggregator `(weblogic92/integration/common/wliaggregator.ear)`	Targeted in the Configuration Wizard	"Creating an AquaLogic Service Bus Domain Using the Configuration Wizard" on page 4-5
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 e-mail 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

When a managed server is down during session activation, configuration changes from the activation are not reflected on that server. In addition, the task status of the session activation is listed as *Partially Activated* to indicate that the activation was not completed on all managed servers. After the managed server is restarted, it synchronizes with the information available with the administration server, and any unactivated changes are activated on the managed server. For more information about session activations, see Using the Change Center in *Using the AquaLogic Service Bus Console*.

You can only re-configure a cluster (for example, add new nodes to the cluster or modify business service configuration) when 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.

Load Balancing in a AquaLogic Service Bus Cluster

One of the goals of clustering your AquaLogic Service Bus application is to achieve scalability. For a cluster to be scalable, each server must be fully utilized. Load balancing distributes the workload proportionately across 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" on page 3-6
- "Load Balancing JMS Functions in a Cluster" on page 3-6

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" under 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. Exceptions 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 *WebLogic Server Performance and Tuning*.

High Availability in an 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 services to JMS producers and consumers. (This is how 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.”

CHAPTER 4

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" on page 4-2
- "Step 2. Prepare an AquaLogic Service Bus Domain" on page 4-5
- "Step 3. Configure AquaLogic Service Bus Security" on page 4-9
- "Step 4. Starting, Stopping, and Monitoring Managed Servers" on page 4-10
- "Step 5. Deploy an AquaLogic Service Bus Configuration" on page 4-11
- "Step 6.Update Your Domain as Your Production Environment Changes" on page 4-11

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 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 configuring 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 an AquaLogic Service Bus domain using the Configuration Wizard. (See "Step 2. Prepare an 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 an 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 multi-homed 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:

ipAddress:portNumber-portNumber	When the port numbers are sequential, for example: `127.0.0.1:7003-7005`
ipAddress:portNumber+...+portNumber	When the port numbers are not sequential, for example: `127.0.0.1:7003+7006+7008`
ipAddress:portNumber,ipAddress:portNumber, ...	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 multi-homed server, that is, multiple IP addresses are assigned to a single computer. Under these circumstances, 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="127.0.0.1:7001,127.0.0.2:7001,127.0.0.3,127.0.0.4:70
01" 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.

  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" on page 4-8 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 *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 an 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" on page 4-5
- "Configuring JMS Resources" on page 4-9

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*

Note: 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 4-1 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 your configuration requires SSL, select the SSL enabled check box.
Configure Managed Servers	Add as many managed servers as required. **Note:** If you need a software HTTP router with `WebLogic HttpClusterServlet` in your domain for load balancing, add one extra server here. If your configuration requires SSL, select the SSL enabled check box for each managed server.
Configure Machines	Configure the type of physical machines used in the cluster.
Configure Clusters	Add a cluster. **Note:** AquaLogic Service Bus domains can support a single cluster.
Assign Servers to Clusters	Add the previously created managed servers to the cluster. **Note:** 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.
Configure Machines	Configure the type of physical machines used in the cluster.

Table 4-1 Responses to Configuration Wizard Prompts

Assign Servers to Machines	Assign each instance of WebLogic Server to the machine in the cluster on which it runs.
Configure JDBC Data Sources	Accept the default values for Name and JNDI name of `wlsbjmsrpDataSource`. Select one of the following to identify the database type and driver for the JMS Reporting Provider Data Store: • Oracle and BEA's Oracle Driver (Type 4) Versions: 9.0.1, 9.2.0, 10 • MS SQL Server and BEA's MS SQL Server Driver (Type 4) Versions: 7.0, 2000 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 Configuring JDBC Data Sources in *Configuring and Managing WebLogic JDBC*. Enter your environment-specific database information in the remaining text boxes. For more information, see "Configure JDBC Data Sources" in Customizing Existing JDBC and JMS Settings in *Creating WebLogic Configurations Using the Configuration Wizard*. **Note:** 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.
Run Database Scripts	Select `wlsbjmsrpDataSource` from the Available JDBC Data Sources list. Select the version of the database from the DB Version drop-down list. Click Run Scripts. 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 `jdbc.log`. If you want the results written to a different file, click the Log File check box and specify the file. **Note:** 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 *`BEA_HOME`*`/weblogic92/integration/common/dbscripts`, where *`BEA_HOME`* is the directory in which you installed AquaLogic Service Bus.

Table 4-1 Responses to Configuration Wizard Prompts

Configure JMS File Stores	Accept the defaults for `rmfilestore` 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 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 about the tasks you must complete, see the *AquaLogic Service Bus Security 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" on page 4-10
- "Monitoring Your Servers" on page 4-11

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.

Tip: Run the `setDomainEnv` script before the `startNodemanager` script to ensure that the AquaLogic Service Bus classes are available to spawned servers. Alternatively, explicitly set classpath in the WebLogic Server Administration Console before attempting to start the server.

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

- The WLIAggregator (`wliaggregator.ear`) is targeted to the first managed server in the cluster. For data aggregation to function properly, the server that `wliaggregator.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" on page 4-12
- "Dropping a Managed Server" on page 4-15
- "Changing a Business Service in a Cluster" on page 4-16
- "Installing a New Version of a Proxy Service in a Cluster" on page 4-16

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 `TemporaryTmplt` 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.

Note: For business services that use a JMS MesageID correlation scheme, you must edit the connection factory settings to add an entry to the table mapping managed servers to queues. For information on how to configure queues and topic destinations, see “JMS Server Targeting” under JMS Server Configuration, in WebLogic Server Administration Console Online Help

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 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" on page 5-1
- "AquaLogic Service Bus Failure and Recovery" on page 5-5
- "High Availability for Poller Based Transports" on page 5-6

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

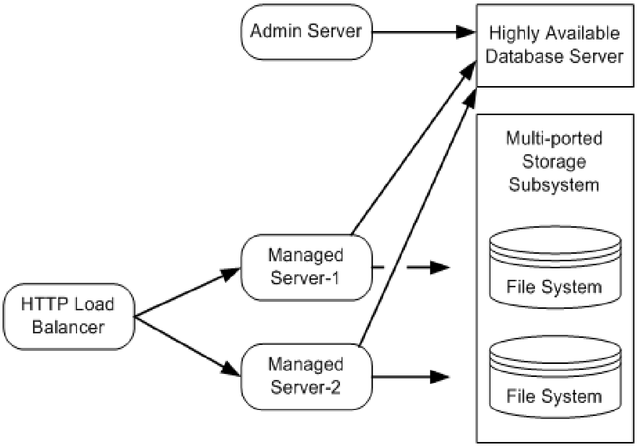
- A set of managed servers in a cluster
- An HTTP load balancer (router)
- Physically shared, highly-available disk subsystems for managed server data—Whole server migration requires that *all* data on a managed server be located on a multi-ported disk. A typical and recommended way to do this is by 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 perform better 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 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 how to prepare to recover 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 not automatically migrated, but must be manually migrated. 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" on page 5-5
- "EIS Instance Failover" on page 5-5

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*.

High Availability for Poller Based Transports

File, FTP, and Email are poller based transports. Since these protocols are not transactional, to increase reliability and high availability, an additional JMS-based framework has been created to allow these transports to recover from failure. These transports use the JMS framework to ensure that the processing of a message is done at least once. However, if the processing is done, but the server crashes or the server is restarted before the transaction is complete, the same file may be processed again. The number of retires depends on the redelivery limit that is set for the poller transport for the domain.

New messages from the target (a directory in case of File and FTP transports and server account in case of Email transport) are copied to the download (stage) directory at the time of polling or pipeline processing.

Note: For FTP transport, a file is renamed as `<name>.stage` in the remote directory. It is copied to the stage directory only at the time of pipeline processing,

For File and FTP transports, a JMS task is created corresponding to each new file in the target directory. For Email transport, an e-mail message is stored in a file in the download directory and a JMS task is created corresponding to each of these files.

These JMS task messages are enqueued to a JMS queue which is pre-configured for these transports when the AquaLogic Service Bus domain is created.

JMS Queues

The following poller transport specific JMS queues are configured for AquaLogic Service Bus domains:

Transport Name	JMS Queue Name
FTP	wlsb.internal.transport.task.queue.ftp

Transport Name	JMS Queue Name
File	wlsb.internal.transport.task.queue.file
Email	wlsb.internal.transport.task.queue.email

A domain-wide message-driven bean (MDB) receives the JMS task. Once the MDB receives the message, it invokes the pipeline in an XA transaction. If the message processing fails in the pipeline due to an exception in the pipeline or server crash, the XA transaction also fails and the message is again enqueued to the JMS queue. This message is re-delivered to the MDB based on the redelivery limit parameter set with the queue. By default, the redelivery limit is `1` (the message is sent once and retried once). If the redelivery limit is exhausted without successfully delivering the message, the message is moved to the error directory. You can change this limit from WebLogic Server Console. For more information, see JMS Topic: Configuration: Delivery Failure in *Administration Console Online Help.*

Note: For a single AquaLogic Service Bus domain transport, the redelivery limit value is global across the domain. For example, within a domain, it is not possible to have an FTP proxy with a redelivery limit of `2` and another FTP proxy with a redelivery limit of `5`.

High Availability in Clusters

For clusters, the JMS queue associated with each of these poller based transport is a distributed queue (each Managed Server has a local JMS queue, which is a member of the distributed queue). The JMS queue for a transport is domain-wide. The task message is enqueued to the distributed queue, which is passed on to the underlying local queues on the Managed Server. The MDB deployed on the Managed Server picks up the message and then invokes the pipeline in a transaction for actual message processing.

Since the JMS queues are distributed queues in cluster domains, high availability is achieved by utilizing the WebLogic Server distributed queue functionality. Instead of all Managed Servers polling for messages, only one of the Managed Servers in a cluster is assigned the job of polling the messages. At the time of proxy service configuration, one of the Managed Servers is configured to poll for new files or e-mail messages. For more information, see “Adding a Proxy Service” in Proxy Services in *Using the AquaLogic Service Bus Console*.

The poller server polls for new messages and puts them to the uniform distributed queue associated with the respective poller transport. From this queue, the message is passed on the local queue on the Managed Server. The Managed Servers receive the messages through MDBs deployed on all the servers through these local queues.

Note: There is a uniform distributed queue with a local queue on all the Managed Servers for each of these poller based transports.

If the managed servers crashes after the distributed queue delivers the message to the local queue, you need to do manual migration. For more information, see "Server Migration" on page 5-4.

When a cluster is created, the uniform distributed queue is created with local queue members - on all the Managed Servers. However, when a new Managed Server is added to an existing cluster, these local queues are not automatically created. You have to manually create the local queues and make them a part of a uniform distributed queue.

To create a local queue:

1. Create a JMS Server and target it to the newly created Managed Server.
2. Create a local JMS queue, set the redelivery count, and target it to the new JMS server.
3. Add this local JMS queue as a member of the uniform distributed queue associated with the transport.

   Note: The JNDI name of the distributed queue is `wlsb.internal.transport.task.queue.file` (for File transport), `wlsb.internal.transport.task.queue.ftp` (for FTP transport) and `wlsb.internal.transport.task.queue.email` (for Email transport).

Load Balancing

Since we use distributed JMS queues, messages are distributed to the Managed Servers based on the load balancing algorithm associated with the distributed queue. By default, the JMS framework uses round-robin load balancing. You can change the algorithm using the JMS module in WebLogic Server Console. For more information, see Load Balancing for JMS in *Using WebLogic Server Clusters*. If one of the Managed Servers fails, the remaining messages are processed by any of the remaining active Managed Servers.

Note: The poller server should always be running. If the poller server fails, the message processing will also stop.

Using the Deployment APIs

You can use the AquaLogic Service Bus MBeans in 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 AquaLogic Service Bus MBeans you can use to programmatically perform deployment operations include:

- `SessionManagementMBean`: used to create, activate and discard a session, or to return the name of an existing session
- `ALSBConfigurationMBean`: used to import and export AquaLogic Service Bus configurations, update environment-specific information (endpoint URIs, etc.), query AquaLogic Service Bus configurations and resources.

The latter MBeans are interfaces in the com.bea.wli.sb.management.configuration package.

Numerous customization options can be applied during deployment. An extended list of environment variables allows you to preserve or tailor settings when moving from one environment to another.

This section contains the following topics:

Tip: AquaLogic Service Bus APIs are documented at http://edocs.bea.com/alsb/docs26/javadoc

Managing Sessions Using Programs and Scripts

AquaLogic Service Bus sessions allow different users to update discrete parts of configuration data without interfering with each other. A session is essentially a named sandbox, in which your changes are abstracted from other users, as well as from the core data (the data on which AquaLogic Service Bus runs), until the changes are activated. In order to modify resources and AquaLogic Service Bus configurations, you must create a session and perform changes in that session. The changes are only reflected in the core data when you activate the session. You can create multiple sessions as long as no two sessions have the same name. A session can only be activated using the instance of the `SessionManagementMBean` that works on that session data.

Each MBean type, except for `SessionManagementMBean`, has one instance per session. When a session is created, a new set of MBean instances (one for each MBean Type) is created automatically. One instance of each MBean Type operates on the core data that is saved to the AquaLogic Service Bus data cache. MBean instances created for a session are destroyed when the session is discarded or activated. MBean instances that operate on core data, however, are never destroyed. MBean instances that work on core data do not support update operations.

Creating, Activating, Discarding, and Locating Sessions

AquaLogic Service Bus sessions are created using the AquaLogic Service Bus Console. The methods in the `SessionManagementMBean` interface directly parallel the 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 methods available in the `SessionManagementMBean` interface and the tasks they perform.

Table A-1 Session Management Methods

To...	Use...
1. Activate a session	`activateSession(String session, String description)`
2. Create a new session with a user-specified name.	`createSession(String session)`

Table A-1 Session Management Methods

To…	Use…
3. Delete the session without activating changes	`discardSession(String session)`
4. Return *true* if a session with the given session name exists	`sessionExists(String session)`

For reference material on the `SessionManagementMBean` interface and Java usage examples, as well as sample code describing how to use MBeans from a Java client and in a script, see the SessionManagementMBean Interface in the `com.bea.wli.sb.management.configuration` package in the AquaLogic Service Bus Javadoc.

Examples

The SessionManagementMBean Interface in the `com.bea.wli.sb.management.configuration` package in the AquaLogic Service Bus javadoc includes example code illustrating how to create a session, how to obtain SessionManagementMBean for creating a session, and ALSBConfigurationMBean for operating on the session that is created and on core data, and so on.

Managing Configuration Tasks Using Programs and Scripts

The AquaLogic Service Bus `ALSBConfigurationMBean` allows you to programmatically query, export and import resources, obtain validation errors, get and set environment values, and in general manage resource configuration in an AquaLogic Service Bus domain. AquaLogic Service Bus configurations are packaged as simple JARs containing AquaLogic Service Bus resources such as proxy services, WSDLs, and business services. These resources can span multiple projects, or contain only partial configuration information. For example, you can export only a subset of a project, a whole project, or subsets of resources from many projects.

The following sections describe how to use the `ALSBConfigurationMBean` to perform these deployment activities from a Java client:

- "Importing, Exporting, and Querying Configurations" on page A-4
- "Updating Environment-Specific Information" on page A-5

For reference material on the `ALSBConfigurationMBean` interface, see the `ALSBConfigurationMBean` Interface in the `com.bea.wli.sb.management.configuration` package in the AquaLogic Service Bus javadoc, which is available at the following URL: http://edocs.bea.com/alsb/docs26/javadoc/

Importing, Exporting, and Querying Configurations

AquaLogic Service Bus configurations are created using the AquaLogic Service Bus Console, and are stored through export in `.jar` files. After 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 `ALSBConfigurationMBean` Interface allow you to manage resources in an AquaLogic Service Bus domain, including tasks such as:

- Query, export, and import resources (includes importing resources from a zip file, and exporting resources at the project level)
- Get and set environment values
- Clone a project, folder, or resource with a new identity
- Modify the existing references from all the resources in the given list to a new set of references
- Customize multiple properties at once
- Obtain validation errors

and so on. See the ALSBConfigurationMBean Javadoc for the comprehensive method summary for the `ALSBConfigurationMBean` methods.

Updating Environment-Specific Information

The methods in ALSBConfigurationMBean and the Customization Class allow you to update environment specific information. This includes.

- Updating the value of endpoints in proxy and business service configurations
- Updating the directory elements in File, E-mail, and FTP transport configurations
- Directly setting environment value(s)
- Searching for environment-specific values specified in a query
- Find environment values specified in a query, and replace all occurrences of the environment value pattern with the given parameter.

and so on. See the ALSBConfigurationMBean Javadoc and the Customization Class Javadoc for a comprehensive summary of methods. Import customizations are supported by the **ALSBImportPlan**—see the ALSBImportPlan Class in the Javadoc.

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.

Examples

The `ALSBConfigurationMBean` Interface in the `com.bea.wli.sb.management.configuration` package in the AquaLogic Service Bus javadoc includes example code illustrating how to import and export AquaLogic Service Bus configurations, how to change environment values, how to query resources, and so on.

APIs Deprecated in AquaLogic Service Bus 2.6

The following deployment APIs (in the ALSBConfigurationMBean) are deprecated in AquaLogic Service Bus 2.6:

1. **To import selected resources/projects/folders from a previously uploaded JAR file**

   ```
   public ImportResult importUploaded(Collection<Ref> refs, boolean
   includeDependencies, boolean preserveExistingEnvValues, String
   passphrase) throws Exception;
   ```

   was deprecated in 2.6 in favor of:

   ```
   ImportResult importUploaded(ALSBImportPlan importPlan)throws Exception;
   ```

   The latter API allows more import customizations via the ALSBImportPlan. For information about the ALSBImportPlan, see

   `http://edocs.bea.com/alsb/docs26/javadoc/com/bea/wli/sb/management/importexport/ALSBImportPlan.html`

2. **To Find and Replace Environment Values**

   Find environment values as indicated by the query and replace all occurrences of the environment value pattern (as returned by EnvValueQuery.getSearchString()) with the given parameter.

   ```
   public int replaceEnvValues(EnvValueQuery query, String replacement)
   throws Exception;
   ```

   Deprecated in 2.6 in favor of:

   ```
   customize(List <Customization> customizations) throws Exception
   ```

   in the ALSBConfigurationMBean.

   Use `FindAndReplaceCustomization` for modifying environment values by doing string substitutions.

3. **To Set One or More Environment Values**

   Set one or more environment values directly

   ```
   public void assignEnvValues(Collection<QualifiedEnvValue> envValues)
   throws NotFoundException, UpdateException;
   ```

   Deprecated in 2.6 in favor of:

   ```
   customize(List <Customization> customizations) throws Exception
   ```

   in the ALSBConfigurationMBean.

Use `EnvValueCustomization` for environment value assignments.

Related Topics

AquaLogic Service Bus APIs in the *BEA AquaLogic Service Bus User Guide*.

AquaLogic Service Bus Deployment Resources

This section describes AquaLogic Service Bus deployment resources such as JMS queues, MDB applications, connection factories, and so on. It includes targeting information for deployment in a cluster configuration.

The AquaLogic Service Bus Console and the UDDI manager run on the administration server. Therefore, you must run an administration server to manage AquaLogic Service Bus and to facilitate publishing and importing to and from a UDDI registry.

The AquaLogic Service Bus metrics aggregator runs on one of the managed servers in a clustered domain. If the managed server to which the aggregator is targeted goes down, alerts will not be fired.

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 toWebLogic Integration

Table B-1 AquaLogic Service Bus Deployment Resources [1]

Resource Target	Resource
Admin Server	httpTransport.war AquaLogic Service Bus HTTP transport application
	FileStore `rmfilestore/FileStore` JMS file store that stores persistent messages
	`wlsbJMSServer`
	`weblogic.wlsb.jms.transporttask.QueueConnectionFactory`
	`wli.reporting.jmsprovider.NonXAConnectionFactory` Connection factory used by reporting provider for transacted sessions
	`wli.reporting.jmsprovider.XAConnectionFactory` Connection factory used by reporting provider for non-transacted sessions
	`wli.reporting.jmsprovider_error.queue` For errors associated with the message reporting queue: `wli.reporting.jmsprovider.queue`
	`ALSB Test Framework`[2] `weblogic92/integration/common/sbTestFwkEjb.ear` The test console application, accessible by users in the following AquaLogic Service Bus roles: `IntegrationAdmin` and `IntegrationDeployer`

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

Resource Target	Resource
Managed Servers in a Cluster	PolledMessageListenerMDBEJB 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.
	PolledMessageListenerMDBEJB 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.
	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.
	WLI Aggregator[3] weblogic92/integration/common/wliaggregator.ear AquaLogic Service Bus metrics aggregator **Note:** WLI Aggregator is targeted to a single managed server.
	Message Reporting Purger(PurgingMDB) weblogic92/integration/common/msgpurger.ear **Note:** Message Reporting Purger is targeted to a single managed server. [4] [5]
	ALSB Resource weblogic92/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.

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

Resource Target	Resource
Distributed Destination	JMS Reporting Provider `weblogic92/integration/common/jmsreportprovider.ear`
	`wlsbJMSServer_auto_`*`1-n`* (AquaLogic Service Bus JMS server on each managed server)
	`QueueIn` Resource for internal use
	`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[4] [5]
	`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
	`wlsb.internal.transport.task.queue.file` Queue for job associated with each new file detected during polling

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

Resource Target	Resource
Distributed Destination (continued)	`wlsb.internal.transport.task.queue.ftp` (Queue for job associated with each new FTP message detected during polling)
	`ReportingMDB` 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`

1. All deployed resources can be viewed and modified in the WebLogic Server Administration Console.
2. 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.
3. For more information, see "Step 4. Starting, Stopping, and Monitoring Managed Servers" on page 4-10
4. 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.
5. 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 5-4.

Index