BEA WebLogic Server®

Configuring and Managing WebLogic Store-and-Forward

Version 9.0 BETA
Revised: December 15, 2004

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Contents

CHAPTER 1

Introduction and Roadmap

This section describes the contents and organization of this guide—*Configuring and Managing WebLogic Store-and-Forward*.

Document Scope and Audience

This document is a resource for system administrators responsible for configuring, managing, and monitoring the WebLogic Store-and-Forward service for use with WebLogic JMS and Web Services Reliable Messaging (WSRM).

The topics in this document are relevant to production phase administration, monitoring, or performance tuning topics. This document does not address the pre-production development or testing phases of a software project. For links to WebLogic Server documentation and resources for these topics, see "Related Documentation" on page 1-2.

It is assumed that the reader is familiar with WebLogic Server system administration. This document emphasizes the value-added features provided by WebLogic Server JMS and key information about how to use WebLogic Server features and facilities to maintain WebLogic Server in a production environment.

Guide to This Document

- This chapter, Chapter 1, "Introduction and Roadmap," introduces the organization of this guide.
- Chapter 2, "Understanding the Store-and-Forward Service," explains the Store-and-Forward service concepts and features, and describe how they work with WebLogic Server.
- Chapter 3, "Configuring Store-and-Forward for JMS Messages," describes how to configure the Store-and-Forward resources for JMS messages.

Related Documentation

This document contains JMS-specific configuration and maintenance information.

For comprehensive guidelines for developing, deploying, and monitoring WebLogic Server applications, see the following documents:

- *Configuring and Managing WebLogic JMS* contains instructions for configuring and managing JMS resources.
- Using the WebLogic Persistent Store provides information about the benefits and usage of the system-wide WebLogic Persistent Store.
- *Configuring and Managing WebLogic Message Bridge* for instructions on configuring a messaging bridge between any two messaging products—thereby, providing interoperability between separate implementations of WebLogic JMS, including different releases, or between WebLogic JMS and another messaging product.

Samples and Tutorials for the System Administrator

In addition to this document, BEA Systems provides a variety of code samples and tutorials that show WebLogic Server configuration and API use, and provide practical instructions on how to perform key development tasks. BEA recommends that you run some or all of the samples before configuring your own system.

Avitek Medical Records Application (MedRec) and Tutorials

MedRec is an end-to-end sample J2EE application shipped with WebLogic Server that simulates an independent, centralized medical record management system. The MedRec application

provides a framework for patients, doctors, and administrators to manage patient data using a variety of different clients.

MedRec demonstrates WebLogic Server and J2EE features, and highlights BEA-recommended best practices. MedRec is included in the WebLogic Server distribution, and can be accessed from the Start menu on Windows machines. For Linux and other platforms, you can start MedRec from the *WL_HOME*\samples\domains\medrec directory, where *WL_HOME* is the top-level installation directory for WebLogic Platform.

MedRec includes a service tier comprised primarily of Enterprise Java Beans (EJBs) that work together to process requests from web applications, web services, and workflow applications, and future client applications. The application includes message-driven, stateless session, stateful session, and entity EJBs.

As companion documentation to the MedRec application, BEA provides tutorials that provide step-by-step procedures for key development tasks, including JMS-specific tasks, such as:

Examples in the WebLogic Server Distribution

WebLogic Server 9.0 optionally installs API code examples in *WL_HOME*\samples\server\examples\src\examples, where *WL_HOME* is the top-level directory of your WebLogic Server installation. You can start the examples server, and obtain information about the samples and how to run them from the WebLogic Server 9.0 Start menu.

Additional Examples Available for Download

Additional API examples for download at http://dev2dev.bea.com/code/index.jsp. These examples are distributed as ZIP files that you can unzip into an existing WebLogic Server samples directory structure.

You build and run the downloadable examples in the same manner as you would an installed WebLogic Server example. See the download pages of individual examples for more information at http://dev2dev.bea.com/code/index.jsp.

New Store-and-Forward Features In This Release

The Store-and-Forward service is new for release 9.0.

CHAPTER 2

Understanding the Store-and-Forward Service

These sections review the different WebLogic Store-and-Forward (or SAF) service concepts and features, and describe how they work with WebLogic Server.

It is assumed the reader is familiar with other WebLogic Server administration concepts.

The SAF Anatomy and Environment

These sections describe the components and participants of the SAF service.

What Is a Store-and-Forward (SAF) Service?

The SAF service enables WebLogic Server to deliver messages reliably between applications that are distributed across WebLogic Server instances. For example, with the SAF service, an application that runs on or connects to a local WebLogic Server instance can reliably send messages to a destination that resides on a remote server. If the destination is not available at the moment the messages are sent, either because of network problems or system failures, then the messages are durably saved on a local server instance, and are forwarded to the remote destination once it becomes available.

WebLogic JMS utilizes the SAF service to enable local JMS message producers to reliably send messages to remote JMS queues or topics, as described in. WebLogic Web Services relies on the the SAF service to support the reliability of Web Services Reliable Messaging (WSRM).

Communication between distributed applications is accomplished through a pair of sending and receiving SAF Agents that are configured on the local and remote server instances, respectively. The working behavior of the store-and-forward service is controlled by a number of configurable parameters.

Capabilities of the SAF Service

The scope of the SAF service for this release is as follows:

- The SAF service can deliver messages within a WebLogic Server cluster, between two clusters, across different WebLogic Server domains, or between two stand-alone server instances.
- An application can only receive directly from a remote server and when the remote server is available.
- Client-side store-and-forward is not supported

What Are SAF Agents?

There are two sides involved in the process of storing and forwarding a message: a sending side and a receiving side. Therefore, the SAF service requires a SAF Agent on each side, called the *sending agent* and the *receiving agent*, respectively. However, a SAF Agent can also be both a receiving agent and a sending agent.

- **Sending agent** — takes care of storing messages to a persistent storage, forwarding messages to the receiving side, and re-transmitting messages when acknowledgements do not come back in time.
- **Receiving agent** — takes care of detecting and eliminating duplicate messages sent by the receiving agent, and delivers messages to the final destination.

What Is JMS SAF?

The JMS Store-and-Forward feature builds on the WebLogic SAF service to provide highly-available JMS message production. For example, a JMS message producer connected to a local server instance can reliably forward messages to a remote JMS destination, even though that remote destination may be temporarily unavailable when the message was sent. JMS Store-and-forward is transparent to JMS applications; therefore, JMS client code still uses the existing JMS APIs to access remote destinations.

For more information, see. Chapter 3, "Configuring Store-and-Forward for JMS Messages."

Main Steps to Configure a SAF Service

These are the main steps for configuring a SAF service with the Administration Console.

1. Configure sending a SAF agent in the sending-side cluster or server instance(s).

   Click the Services →Store-and-Forward Agents node to open the Summary of Store-and-Forward Agents page.

   a. Click the "Lock and Edit" button in the Change Center to activate the configuration pages.

   b. Click the New button to open the Create a New Store-and-Forward Agent page.

   c. Provide a name and, optionally, select a persistent store if one was configured for SAF. Otherwise, leave this property set to None when using the default Persistent Store.

   d. In the Agent Type listbox, select Sending-Only.

      Note: A SAF agent can also be both a Sending and Receiving agent.

   e. Click Next.

   f. On the Select Targets page, select the cluster or server instance(s) where you want to deploy the SAF agent.

   g. Click Finish.

h. Click the Activate Changes button in the Change Center to activate the new SAF Agent.

2. If you want to configure additional Advanced Sending Agent options for the new SAF agent, such as Retry Delay parameters and a Message Buffer Size, then reopen the SAF agent you created and configure those options.

3. To configure receiving a SAF agent(s) in the receiving-side cluster or server instance, repeat Steps 1–2 for the receiving cluster or server instance.

4. Optionally, configure a persistent store if you want a dedicated store for stored-and-forwarded messages. For more information about configuring a persistent store, see Using the WebLogic Persistent Store in *Designing and Configuring WebLogic Server Environments*.

   Click the Services →Persistent Stores node to open the Summary of Persistent Stores page.

5. Optionally, configure a Path Service if the sending-side is a cluster and the JMS producer is associated with a Unit-of-Order, which enables JMS message producers to group ordered messages into a single unit. For more information about JMS Unit-of-Order, see *Using Message Unit-of-Order* in *Programming WebLogic JMS*.

   Click the Services →Path Services node to open the Summary of Path Services page.

   The Path Service is a persistent map that can be used to store the mapping of a group of messages to a messaging resource such as a SA agent. For more information about configuring a Path service, see *Using the WebLogic Path Service* in *Configuring and Managing WebLogic JMS*.

6. Configure JMS SAF for a JMS module, as described in Chapter 3, "Configuring Store-and-Forward for JMS Messages."

   Click the Services →JMS →JMS Modules →Store-and-Forward node to open the summary of JMS Store-and-Forward page.

CHAPTER 3

Configuring Store-and-Forward for JMS Messages

The JMS Store-and-Forward feature builds on the WebLogic Store-and-Forward (SAF) service to provide highly-available JMS message production. For example, a JMS message producer connected to a local server instance can reliably forward messages to a remote JMS destination, even though that remote destination may be temporarily unavailable when the message was sent. JMS Store-and-forward is transparent to JMS applications; therefore, JMS client code still uses the existing JMS APIs to access remote destinations.

The following sections explain how to:

What Are SAF Resources In a JMS Module?

When configuring Store-and-Forward for a JMS module, you need to configure the following SAF resources:

What Are SAF Imported Destinations?

A SAF Destination is a representation of a remote JMS Destination in the local server or Cluster. This SAF Destination is said to be imported in the local cluster or server. A SAF Destination can only be imported if the associated JMS Destination in the remote cluster or server is exported. A collection of SAF Destinations is called JMS Imported Destinations. Each collection of JMS Imported Destinations is associated with a SAF Remote Context. They can also share the same JNDI prefix, Time-To-Live default (message expiration time), and SAF Error Handling name.

When a JMS producer sends messages to an SAF Destination, these messages are added to the SAF Destination. A SAF Forwarder forwards these messages to the actual JMS Destination (which this Imported JMS Destination represents) when the JMS Destination is reachable, using the SAF Remote Context.

What Is a SAF Remote Context?

A SAF Remote Context defines the URL of the remote Cluster or Server where the JMS Destination is exported, and the security credentials to be authenticated and authorized in the remote cluster or server.

What Is SAF Error Handling?

Store-and-Forward (SAF) Error Handling defines the action to be taken when the delivery of a JMS message fails to be sent to a JMS Imported SAF Destination. This includes an Error Handling Policy (Redirect, Log, Discard, or Always Forward), a Log Format, and sending Retry parameters.

Main Steps to Configure SAF In a JMS Module

These are the main steps for configuring the SAF JMS feature for forwarding JMS messages.

1. Before you begin, you must configure a sending and receiving SAF agent, as described in "Main Steps to Configure a SAF Service" on page 2-3.
2. Click the "Lock and Edit" button in the Change Center to activate the configuration pages.
3. Create a new JMS module for your JMS SAF resources.
   a. Click the Services →JMS →JMS Modules node to open the Summary of JMS Modules page.
   b. Click the New button to open the Create a New JMS Module page.
   c. Provide a name, a descriptor name, and optionally, specify where in the domain the system module will reside. Otherwise, it will default to the domain's `config/jms` directory.
   d. Click Next.
   e. On the Select Targets page, target the JMS module to the cluster or server instance where the SAF agent is configured.
   f. Click Finish.

   For more information, see JMS System Modules in *Configuring and Managing WebLogic JMS*.
4. Create a new SAF Remote Context in the JMS module.
   a. Click the Services →JMS →JMS Modules →Store-and-Forward →Remote Context node.
   b. If necessary, select the JMS module you created in the SAF Remote Contexts in module list box.
   c. Click the New button to open the Create a New SAF Remote Context page.
   d. Uniquely name the new SAF Remote Context configuration.
   e. In the URL property, provide the URL of the remote context.
   f. Enter a Username/Password combination for the remote context.
   g. Click Finish.
5. Create a new SAF Error Handling configuration in the module.

a. Click the Services →JMS →JMS Modules →Store-and-Forward →Error Handling node.

b. If necessary, select the JMS module you created in the "Error Handler Configurations in module" list box.

c. Click the New button to open the Create a New Error Handling page.

d. Uniquely name the Error Handler configuration, and then accept the remaining default properties.

e. Click Finish.

6. Create a new SAF Imported Destination in the module and associate it with the SAF Remote Context you created for the module.

a. Click the Services →JMS →JMS Modules →Store-and-Forward →Imported Destinations node.

b. If necessary, select the JMS module you created in the "SAF Imported Destinations in module" list box.

c. Click the New button to open the Create a New SAF Imported Destinations page.

d. Uniquely name the SAF Imported Destination.

e. In the Remote SAF Context list box, select the SAF Remote Context that you created for the module.

f. Click Next.

g. On the Select Targets page, target the imported destination to the appropriate cluster, server instance, or JMS server.

h. Click Finish.

7. Reopen the SAF Imported Destination you created, and configure the following resources:

a. Click the Services →JMS →JMS Modules →Store-and-Forward →Imported Destinations node.

b. If necessary, select the JMS module you created in the "SAF Imported Destinations in module" list box so the SAF Imported Destination you created appears in the table.

c. Click your SAF Imported Destination name in the table to edit the configuration.

d. On the General page, use the Error Handling listbox to select the configured Error Handling configuration, and then click Save.

e. Click the Queues tab, to open the SAF Queues page.

f. Click the New button to create a new SAF Queue.

g. Reopen the newly created SAF Queue and specify a Remote JNDI Name (required).

h. In, the Error Handling listbox, select the configured Error Handling configuration (required).

i. Optionally, you can configure other SAF Queue parameters or accept the defaults.

j. Save the new SAF Queue.

8. Click the Activate Changes button in the Change Center to activate your new JMS SAF configuration.

Frequently Asked Questions About JMS SAF

This section answers commonly asked questions about how JMS SAF operates in a domain.

Q. Which sending agent or receiving agent is picked by SAF for a JMS producer?

A. WebLogic Sever's cluster load balancing is used to pick a sending agent or receiving agent for a given JMS producer. Once these SAF agents are picked, they are used for the life of the JMS producer

Q. How do JMS clients find a SAF destination?

A. A SAF destination can be found the same way a non-SAF JMS destination is found.

- JMS clients can look up a SAF Destination in JNDI
- `createDestination` API

  JMS clients can also use the `createDestination` API to find JMS Destination. JMS clients have to use the SAF destination name in a JMS module. The `Name` must be a fully-qualified name delimited by exclamation points (!). For example:

  ```
  <EAR Name>!<JMS Module Name>!<ImportedDestinationsName>!<SAFQueue or
  SAFTopic Name>
  ```

Q. Can a JMS producer sending messages to a JMS SAF imported destination be associated with a JMS Unit-of-Order?

A. Yes. For information about the Unit-of-Order feature, see For more information about configuring a Path service, see *Using Message Unit-of-Order* in *Programming WebLogic JMS*.

Q. Why does my JMS producer associated with a JMS Unit-of-Order fail to send messages if the sending-side is a cluster?

A. In order to use JMS Unit-of-Order with SAF, you must configure the Path Service for the sending-side cluster. For information about the Path Service feature, see *Using the WebLogic Path Service* in *Configuring and Managing WebLogic JMS*.

Q. Do different JMS producers in the same Unit-of-Order pick up the same Sending Agent?

A. Yes. JMS SAF uses the Path Service to route to the same Sending Agent.

Q. Can a consumer be attached to a JMS SAF Imported Destination?

A. No. JMS consumers can only be attached to actual JMS destinations.

Q. Can a distributed destination be imported?

A. Yes, it can be imported using its JNDI name.

Q. Where do I configure Server Affinity, Load Balancing Enabled, and Forward Delay for a distributed destination that is imported in a sending cluster?

A. Server Affinity and Load Balancing Enabled are configured in the JMS connection factory on which the JMS producer was created. (A JMS connection factory creates a JMS connection; a JMS connection creates a JMS session; a JMS session creates a JMS producer). Forward Delay is configured on the JMS distributed destination.

Q. Are the Server Affinity and Load Balancing parameters configured on a JMS connection factory in the sending cluster or server honored on the receiving cluster or server where the JMS Destination resides?

A. Yes. These attributes on the sending cluster or server are honored on the receiving cluster or server. For information about Server Affinity and Load Balancing for distributed destinations, see *JMS Distributed Destination Tasks* in *Configuring and Managing WebLogic JMS*

Q. Do XA transactions on the sending-side of a cluster ever cross the JMS SAF boundary? In other words, can the receiving-side participate with an XA transaction from the sending- side?

A. No,.messages are not forwarded until the XA transaction is committed.

Q. Does JMS SAF preserve the order of messages sent in a JMS session from a sending-side to a JMS destination?

A. Yes.

Q. In the SAF Remote Context, should I configure a Principal Name or a Username/Password?

A. You can configure the Remote SAF Context anyway you want. Username and Passwords are stored in the JMS module, and the principal name is stored in a Credential Mapper configured in the sending-side domain.