BEA WebLogic Server®

Configuring and Managing WebLogic Store-and-Forward

Version 9.0
Revised: July 22, 2005

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Contents

1. Introduction and Roadmap

2. Understanding the Store-and-Forward Service

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 SAF 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 SAF for JMS Messages," describes how to configure the Store-and-Forward resources for JMS messages.
- Chapter 4, "Monitoring and Managing SAF Agents," describes how to describes how to monitor and manage the run-time statistics for your Store-and-Forward service.
- Chapter 5, "Troubleshooting WebLogic SAF," provides a list of frequently asked questions concerning SAF configuration issues.

Related Documentation

This document contains SAF-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, such as JMS servers, JMS modules, and the Path Service.
- Using Reliable Web Service Messaging in *Programming WebLogic Web Services* contains instruction for using SAF with Web Services Reliable Messaging (WSRM).
- Using the WebLogic Persistent Store in *Configuring WebLogic Server Environments* provides information about the benefits and usage of the system-wide Persistent Store.

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.

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.

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.

CHAPTER 2

Understanding the Store-and-Forward Service

These sections review the different WebLogic Store-and-Forward (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 WebLogic SAF Environment

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

The 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 an endpoint 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 saved on a local server instance, and are forwarded to the remote endpoint 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 "Using SAF with WebLogic JMS" on page 2-4.

WebLogic Web Services relies on the SAF service to support the reliability of Web Services Reliable Messaging (WSRM), as described in "Using SAF with Web Services Reliable Messaging" on page 2-4.

SAF Service Agents

There are two sides involved in the process of storing and forwarding messages: a local sending side and a remote receiving endpoint. SAF agents are responsible for store-and-forwarding messages between these local sending and remote receiving endpoints. A SAF agent can be configured to have only sending capabilities, receiving capabilities, or both.

JMS SAF only requires a sending agent on the sending side for JMS messages. Whereas, WSRM SAF requires both a sending agent and a receiving agent.

- **Sending agent** — Used for JMS messages and WSRM. If message persistence is required, a sending agent stores messages to a persistent storage, forwards messages to the receiving side, and re-transmits messages when acknowledgements do not come back in time.
- **Receiving agent** — Used only for WSRM. Receiving agents detect and eliminate duplicate messages sent by a sending agent, and delivers messages to the final destination.

  Note: In the case of JMS messages, the JMS server associated with a remote exported JMS destination on the receiving side manages duplicate elimination.

SAF Agent Configuration Parameters

A SAF agent is a configurable object that is similar to a JMS server in that it manages message persistence, paging parameters, and thresholds and quotas. The working behavior of the SAF service is controlled by a number of configurable parameters on SAF agents:

- General configuration parameters, including:
  - select persistent storage
  - setting message paging defaults
  - enable message logging
  - specify delivery retry settings
  - determine window size
  - specify message acknowledgement intervals (WSRM only)

  For more information about delivery retry settings, logging, and paging defaults, see "Designing SAF Agents" on page 2-8.

  For more information about all general configuration parameters for SAF agents, see "Store-and-Forward Agents: Configuration: General" in the *Administration Console Online Help*.

- Threshold and quota parameters for controlling the message throughput of SAF agents.

  For more information about threshold and quota parameters, see "Designing SAF Agents" on page 2-8.

- Monitoring capabilities for SAF agents, remote endpoints, and conversations. You also have the capability to manage incoming, forwarding, and receiving message operations on SAF agents and to remote endpoints.

  For more information about monitoring options for SAF agents, see Chapter 4, "Monitoring and Managing SAF Agents."

Persistent Store Rules When Targeting SAF Agents to a Cluster

When targeting a SAF agent to a standalone server, you can either use the server's default persistent store or select an explicitly configured store. However, the following persistent store selection rules apply in order to properly target SAF agents in a cluster.

- Each server instance in the cluster must use the default persistent store, and the SAF agent(s) must also use the default store.

- For JMS messages, the sending SAF agent and the imported destination must reside on the same server instance. When a SAF agent is targeted to a cluster, an agent is automatically configured on each server instance in the cluster.

Using SAF with WebLogic JMS

The JMS store-and-forward uses a single sending SAF agent 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 SAF 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 SAF for JMS Messages."

Using SAF with Web Services Reliable Messaging

WSRM uses a pair of sending and receiving SAF agents that are configured on the local and remote server instances, so that two Web Services running on different server instances can communicate reliably in the presence of failures in software components, systems, or networks. In particular, a message sent from a source endpoint to a destination endpoint is guaranteed either to be delivered, according to one or more delivery assurances, or to raise an error.

For more information, see "Using Reliable Web Service Messaging" in *Programming Web Services for WebLogic Server*.

When to Use the SAF Service

The SAF Service should be used when forwarding JMS or WSRM messages between WebLogic Server 9.0 or later domains.

The SAF service can deliver messages:

- Between two stand-alone server instances.
- Between server instances in a cluster.
- Across two clusters in a domain.
- Across separate domains.

When not to use the SAF service:

- Forwarding messages to prior releases of WebLogic Server.
- Interoperating with third-party JMS products (for example, MQSeries).

  For these tasks, you should use the WebLogic Messaging Bridge. See "Understanding the Messaging Bridge" in *Configuring and Managing WebLogic Messaging Bridge*.
- When using temporary destinations with the `JMSReplyTo` field to return a response to a request.

There are also some restrictions for in this release when using JMS SAF:

- An application can only receive messages directly from a remote server, and only when the remote server is available.
- Client-side store-and-forward is not supported in this release.

Configuring a Basic SAF Service

These are the main tasks that must be completed to implement the SAF service in a domain.

1. Configure a SAF agent in the sending-side cluster or server instance(s). There are a number of ways to create SAF agents:
   - The WebLogic Server Administration Console enables you to configure, modify, target, monitor, and delete SAF agents in your environment. For step-by-step instructions of the SAF agent configuration tasks, see "Configure Store-and-Forward agents" in the *Administration Console Online Help*.

- WebLogic Java Management Extensions (JMX) enables you to access the `SAFAgentMBean` and `SAFAgentRuntimeMBean` MBeans to create and manage SAF agents. For more information see "Overview of WebLogic Server Subsystem MBeans" in *Programming WebLogic Management Services with JMX*.
- The WebLogic Scripting Tool (WLST) enables you to create and manage JMS servers and JMS system resources. For more information, see "Using WLST to Manage JMS Servers and JMS System Resources" in *Configuring and Managing WebLogic JMS*.

2. When using WSRM, even if you configure a SAF agent with sending *and* receiving capability on the sending side, you *must* also configure a SAF agent with receiving capability on the receiving-side cluster or server instance.
3. When configuring a SAF agent, you can accept the server's default store or configure a store if you want a dedicated store for SAF messages. However, when targeting a SAF agent to a cluster, you must accept the server's default store, as described in "Persistent Store Rules When Targeting SAF Agents to a Cluster" on page 2-3.

   For more information about persistent stores, see "Using the WebLogic Persistent Store" in *Configuring WebLogic Server Environments*.
4. For WebLogic JMS, configure SAF Imported Destination, SAF Context Handling, and SAF Error Handling (optional) objects in a JMS module, as described in Chapter 3, "Configuring SAF for JMS Messages."
5. Configure a Path Service when:
   - For JMS messages when 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*.
   - Using WSRM in a cluster.

   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 SAF agent. For more information about configuring a Path service, see "Using the WebLogic Path Service" in *Configuring and Managing WebLogic JMS*.

Note: When using the message unit-of-order with a Path service to deliver ordered non-persistent JMS messages within a cluster, but you are using a quality-of-service (QOS) of At-Most-Once, then if the JMS producer is recycled (due to a brief network outage) there is a *possibility* that messages can sent out of order to the remote endpoint. Persistent messages are always sent with a QOS of Exactly-Once, and therefore, are

always guaranteed to be delivered in order. For more information on selecting a QOS for JMS messages, see "Selecting a Quality-of-Service (QOS) Level" on page 3-5.

Designing SAF Agents

Use the following information to help you design and configure SAF agents for forwarding messages.

Setting a Message Time-To-Live Duration and Message Delivery Failure Policy

The reliability of SAF is time based. You can configure the duration that a message needs to be delivered by a SAF agent reliably using the Time-To-Live parameter. When the configured Time-To-Live duration expires, the sending agent removes the message from its storage and discontinues attempts to retransmit the message to the receiving side. It is up to the application to decide what to do with the messages that fail to be delivered when its Time-To-Live expires.

For information on how JMS handles failed message deliveries, see "SAF Error Handling" on page 3-3.

Messages can fail to be delivered for the following reasons:

- Network outage
- The endpoint does not exist (not configured)
- The endpoint is down
- Endpoint quota failure
- Security denial
- Required QOS is not supported
- The WSRM conversation times out (the conversation is idle for too long).

Logging Failed Message Deliveries

If the Logging parameter is enabled for the sending agent, it logs a message in the server log for every failed message. This is an alternative for applications that do not have their own failure handling or their failure handling cannot complete. For more information on WebLogic logging, see "Understanding WebLogic Logging Services" in *Configuring Log Files and Filtering Log Messages*.

Setting Delivery Retry Attempts

The sending agent needs to connect to the receiving side in order to forward messages over, yet there are times that the connection may not be available. When an attempt to send a message fails, the sending agent must retry until it succeeds. Similarly, if the desired QOS is Exactly-Once or At-Least-Once, the sending agent must keep sending a message until it receives an acknowledgement for that message.

To control the frequency of attempts and the interval between two subsequent attempts, you can configure the default values for Retry Delay Base, Retry Delay Multiplier, and Retry Delay Maximum parameters. The Retry Delay Multiplier must be greater or equal to 1, and the Retry Delay Maximum value must greater or equal to the Retry Delay Base value. By default, the Retry Delay Multiplier value is set to 1, which means there's a fixed interval, defined by the Retry Delay Base value, between two successive attempts and that Retry Delay Maximum will be ignored. If the Retry Delay Multiplier is greater than 1, an exponential back-off algorithm will be used to adjust the retry intervals.

The delays will be exponentially increased, starting from the Retry Delay Base value, and will be multiplied by the Retry Delay Multiplier value each time. The amount of delay will not be increased any more once the Retry Delay Maximum value is reached. Once there is a successful attempt, the Retry Delay Multiplier value is reset to the Retry Delay Base value.

For more information about the delivery retry parameters for SAF agents, see "Store-and-Forward Agents: Configuration: General" in the *Administration Console Online Help*.

Using Message Quotas, Thresholds, and Paging

Persistent messages are saved in the persistent store on the sending side until they are successfully forwarded to and acknowledged by the receiving side. However, non-persistent messages pending for delivery exist in-memory on the sending side, and all of the history records also exist in-memory on the receiving side. If the remote side is not available for a long time, the pending

non-persistent messages could use up the sending side server's memory and even take the server down. By configuring quotas for each SAF agent, you can protect the server from running out of memory. Once the quota is about to be exceeded, the SAF agent will reject any new requests.

You can also configure SAF agents to page out messages or history records to a paging directory before the agent reaches the quotas. Paging will be triggered by certain conditions specified as thresholds in the SAF agent's configuration. The persistent store for messages or history records is also used for paging purposes.

The SAF agent threshold and quota parameters and relationship are the same as for JMS destinations and JMS servers.

For more information about threshold and quota parameters for SAF agents, see "Store-and-Forward Agents: Configuration: Thresholds & Quota" in the *Administration Console Online Help*.

Boot-Time Recovery

When a WebLogic Server instance reboots, the messages that were not sent before the server instance went down are recovered from the server's persistent store. The sending agent then attempts to send those messages to the remote side if they have not expired. Similarly, on the receiving side, the history records are recovered during reboot.

SAF Service Migration

WebLogic Server supports migration at the server level—a complete server instance, and all of the services it hosts can be migrated to another machine, either automatically, or manually. For more information on WebLogic service migration, see "Server Migration" in *Using WebLogic Server Clusters*.

CHAPTER 3

Configuring SAF 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:

SAF Resources In a JMS Module

In this release, JMS configurations, such as queue, topics, and connections factories) are stored outside of the WebLogic domain as module descriptor files, which are defined by XML documents that conform to the `weblogic-jmsmd.xsd` schema. JMS modules also provide the configuration of SAF resources that allow WebLogic JMS to store-and-forward JMS messages. For more information on JMS modules, see "Understanding JMS Resource Configuration" in *Configuring and Managing WebLogic JMS*.

Once your JMS SAF resources are configured, a configured SAF sending agent forwards messages to the receiving side, re-transmits messages when acknowledgements do not come back in time, and, if message persistence is required, a stores messages to a persistent storage.

JMS store-and-forward is transparent to JMS applications. Existing JMS applications can take advantage of this feature without any code changes. In fact, you only need to configure imported JMS destinations within JMS modules, which then associate remote JMS destinations to local JNDI names. JMS client code still uses the existing JMS APIs to access the imported destinations. JMS store-and-forward is only for message production; therefore, JMS clients still need to consume messages directly from imported destinations.

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

- "SAF Destinations" on page 3-2
- "SAF Remote Context" on page 3-3
- "SAF Error Handling" on page 3-3

SAF Destinations

A SAF *destination* is a representation of a JMS queue or topic in a remote server instance or cluster that is imported into the local cluster or server instance, so that the local server instance or cluster can send messages to the remote server instance or cluster. All JMS destinations are automatically exported by default, unless the Export SAF Destination parameter on a destination is explicitly disabled.

A collection of imported SAF destinations is called *SAF imported destinations*. Each collection of 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 object.

When a JMS producer sends messages to a SAF destination, these messages are stored on the SAF destination for future delivery. A SAF agent forwards the messages to the remote JMS

destination (that the imported destination represents) when the destination is reachable, using the SAF Remote Context.

SAF Remote Context

A remote SAF context defines the URL of the remote server instance or cluster where the JMS destination is exported from. It also contains the security credentials to be authenticated and authorized in the remote cluster or server. A SAF remote context configuration is required to use imported destinations. A remote SAF context can be re-used by multiple imported destination configurations.

SAF Error Handling

SAF error handling resources define the action to be taken when the SAF service fails to forward messages to a remote destination. Configuration options include an Error Handling Policy (Redirect, Log, Discard, or Always-Forward), a Log Format, and sending Retry parameters. SAF error handling resources are not required for imported destinations, but are recommended as a best practice.

Creating JMS SAF Resources

There are a number of ways to create SAF resources in a JMS module.

- The WebLogic Server Administration Console enables you to configure, modify, target, monitor, and delete JMS system modules and JMS resources in your environment. For a road map of the JMS SAF resource configuration tasks, see "Create Store-and-Forward Agents" in the *Administration Console Online Help*.
- The WebLogic Scripting Tool (WLST) enables you to create and manage JMS servers and JMS system resources. For more information, see "Using WLST to Manage JMS Servers and JMS System Resources" in *Configuring and Managing WebLogic JMS*.
- WebLogic Java Management Extensions (JMX) enables you to access the `JMSSystemResourceMBean` and `JMSRuntimeMBean` MBeans to create and manage JMS destinations and connections. For more information see "Overview of WebLogic Server Subsystem MBeans" in *Developing Custom Management Utilities with JMX*.
- JMS Module Helper Extension APIs enable you to locate JMS runtime MBeans, as well as methods to manage (locate/create/delete) JMS system module configuration resources in a given module. For more information, see "Using the JMS Module Helper to Manage Applications" in *Programming WebLogic JMS* or the `JMSModuleHelper` Javadoc.

Main Steps to Configure SAF In a JMS Module

These are the main steps when using the Administration Console to configure the SAF resources for forwarding JMS messages to remote destinations.

1. On the sending side, configure a SAF sending agent, as described in "Create Store-and-Forward agents" in the *Administration Console Online Help*.
2. If necessary, create JMS system modules on both the sending and receiving side to contain your JMS destination resources, as described in "Configure JMS system modules" in the *Administration Console Online Help*.

   For more information, see "JMS System Resource Modules" in *Configuring and Managing WebLogic JMS*.
3. In the sending side JMS module, configure a new remote SAF context resource to define the URL of the receiving side where the remote queue or topic is exported from. For step-by-step instructions, see "Create SAF remote contexts" in the *Administration Console Online Help*.
4. In the sending side JMS module, optionally configure a SAF error handling configuration to define the action to be taken when the SAF service fails to forward messages to a remote destination. For step-by-step instructions, see "Create SAF error handling" in the *Administration Console Online Help*.
5. In the sending-side JMS module, configure a SAF imported destination and associate it with the remote SAF context and SAF error handling resources you created in the module. For step-by-step instructions, see "Create SAF imported destinations" in the *Administration Console Online Help*.
6. Reopen the SAF imported destination you created, and configure a SAF queue and/or SAF topic to represent the remote queue and/or topic on the receiving side. The SAF queue or topic uses the same JNDI name as the remote queue or topic. For step-by-step instructions, see:
   - "Create SAF Queues" in the *Administration Console Online Help*
   - "Create SAF Topics" in the *Administration Console Online Help*
7. By default, all JMS destinations are available for access by SAF imported destinations. However, you can selectively specify not to export a destination by changing the destination's SAF Export Policy value to None. This way, remote users cannot send messages to a destination using store-and-forward.

Designing SAF for JMS Messages

Use the following information to help you design and configure a WebLogic SAF for forwarding JMS messages.

Selecting a Quality-of-Service (QOS) Level

Persistent JMS messages are always forwarded with Exactly-Once QOS provided by the SAF service. For non-persistent messages, three different QOS levels can be defined on imported SAF queues and topics:

- Exactly-once—The highest QOS guarantees that a message is forwarded to the remote endpoint once and only once. With Exactly-once, messages survive server crashes and network down time, while guaranteeing one occurrence of each message at the endpoint.
- At-least-once—Guarantees that a message is forwarded to the remote endpoint at least once, but with the possibility of duplicates. With At-least-once, multiple copies of a message might show up on the remote endpoint because of network failures or server crashes that occur when the message is in transit.
- At-most-once—The lowest QOS guarantees that each message is forwarded to the remote endpoint only once, if at all. It does not guarantee that a message is forwarded to the endpoint. With At-most-once, messages may get lost because of network failures or server crashes. No duplicate messages will reach the endpoint.

How SAF Handles Delivery Modes

A SAF application can also specify a delivery mode for each message, as follows:

- Persistent messages are saved in the persistent store on the sending side until they are successfully forwarded to and acknowledged by the receiving side.
- Non-persistent messages are kept in memory on the sending side until the receiving side acknowledges them. This means that non-persistent messages can be lost if the sending side crashes.

Using Message Unit-of-Order

Within a cluster, a JMS producer can be associated with a *message unit-of-order*, which enables a stand-alone message producer, or a group of producers acting as one, to group messages into a single unit with respect to the processing order. For more information about JMS Unit-of-Order, see "Using Message Unit-of-Order" in *Programming WebLogic JMS*.

Imported SAF destinations can use either a Hash Map or a Path Service to group ordered messages in a cluster. However, as a best practice, you should configure a Path Service. 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 SAF agent. For more information about configuring a Path service, see "Using the WebLogic Path Service" in *Configuring and Managing WebLogic JMS*.

Transactional Messages

If an application message is in a transaction, saving the message in the persistent storage becomes part of the user transaction to preserve Exactly-Once semantics.

In particular, the message will be removed from the persistent storage as part of the transaction rollback if the application decides to rollback the transaction. However, forwarding is *not* part of the application transaction. The sending agent will not forward a transactional message until the transaction commits. Within a transaction, message ordering is preserved based on when the messages are sent.

Message Compression Across SAF Boundaries

JMS store-and-forward can compress messages when they are forwarded between different clusters. A message compression threshold can be set programmatically using a JMS API extension to the `WLMessageProducer` interface, or administratively by either specifying a Default Compression Threshold value on a connection factory or on a JMS remote SAF context.

For more information, on using message compression for JMS messages, see "Tuning WebLogic JMS" in *Configuring and Managing WebLogic JMS*.

When an uncompressed message that exceeds the remote SAF context's compression threshold value is about to be forwarded across the SAF boundary, it is compressed. The message stays compressed until it is received by the remote endpoint. If the message has already been compressed when it crosses the SAF boundary because the compression is turned on the connection factory, the message will stay compressed across SAF boundary no matter if the SAF compression is triggered or not.

SAF to a Distributed Destination

A remote endpoint can be a distributed destination. Messages to a remote distributed destination are stored and forwarded in the same way as messages that are forward to remote standalone destinations. The SAF sending agent routes the messages to a member of the distributed destination the same way as we do currently. For more information on configuring distributed

destinations, see "Configuring Distributed Destinations" in *Configuring and Managing WebLogic JMS*.

Using the JMSReplyTo Field with SAF

Generally, JMS applications can use the `JMSReplyTo` header field to advertise its temporary destination name to other applications. However, the use of temporary destinations with a `JMSReplyTo` field is not supported for SAF imported destinations.

For more information on using JMS temporary destinations, see "Using Temporary Destinations" in *Programming WebLogic JMS*.

Securing JMS Messages

The following security measures apply to SAF imported destinations.

- Secure roles and policies can be set on imported SAF queues and topics. For more information on configuring roles and policies, see the following sections in the *Administration Console Online Help*:
  - "Configure SAF Queue Security Roles"
  - "Configure SAF Queue Security Policies"
  - "Configure SAF Topic Security Roles"
  - "Configure SAF Topic Security Policies"
- The SAF service does not preserve a message's `JMSXUserID` across SAF boundaries. A JMSXUserID is a system generated property that identifies the user sending the message. JMSXUserID is defined in the JMS Specification.

CHAPTER 4

Monitoring and Managing SAF Agents

The following sections explain how to monitor the run-time statistics and manage message operations for your SAF agents from the Administration Console.

Monitoring SAF Agents

The Administration Console provides monitoring capabilities for SAF agents, remote endpoints, and connections. The following information can be viewed on each SAF agent:

- For WSRM only, all of the receiving agents that an agent has talked to
- For WSRM and JMS, all of the endpoints that an agent has sent messages to
- For WSRM only, all of the conversations that have occurred

The following statistics information will be recorded on each sending agent:

- Current, total, and high count of messages (per agent and per remote endpoint)
- Total number of messages that has failed (per conversation, agent and remote endpoint)
- Current, total, and high count of conversations (per agent and per remote endpoint)
- Total uptime, downtime, and last time connected or disconnected for a remote agent.

Managing Message Operations on SAF Agents

The Administration Console also provides run-time message management capabilities for SAF agents, including:

- Temporarily pausing any incoming, forwarding, and receiving message operations on a SAF agent, as described in "Monitor Store-and-Forward statistics" in the *Administration Console Online Help*.
- Temporarily pause incoming and forwarding message operations to remote endpoints. You can also expire all messages or purge all messages for remote endpoints, as described in "Monitor Store-and-Forward remote endpoints" in the *Administration Console Online Help*.
- Destroy WSRM conversations and purge all messages associated with the conversation, as described in "Monitor Store-and-Forward conversations" in the *Administration Console Online Help*.

CHAPTER 5

Troubleshooting WebLogic SAF

The following sections explain how to troubleshoot WebLogic SAF.

Frequently Asked Questions About JMS SAF

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

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

A. WebLogic Sever's cluster load balancing is used to pick a sending agent for a given JMS producer. Once a SAF agent is picked, it is 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 "Configuring Distributed Destinations" 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 a transaction from the sending- side?

A. No. Messages are not forwarded until the 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.