

Automatic Service Migration in WebLogic Server

An Oracle White Paper

July 2008



NOTE:

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

INTRODUCTION

Today's mission critical applications require highly available services. WebLogic Server offers messaging, transaction, and other system services to facilitate building enterprise grade applications. These services can operate in a clustered environment to help build scalable and highly available applications. Previous releases of Oracle WebLogic Server provided support for administrators to manually migrate highly available services from a failed cluster server to a healthy server. Oracle WebLogic Server 10.3 adds support to the WLS Clustering infrastructure to automatically migrate the services for high availability without any administrative intervention. The authors present various elements of the automatic service migration framework in general, followed by detailed explanations and best practices for this feature in the WebLogic Server 10.3 release.

TERMS AND DEFINITIONS

Term	Definition
WebLogic Domain	A WebLogic Server domain is a logical grouping of WebLogic Server resources for administration purpose. Each domain has a special WebLogic Server instance called the Administration Server. It is the central point to configure and manage all resources in the domain. Usually, a domain also includes additional WebLogic Server instances called Managed Servers.
WebLogic Cluster	A WebLogic Server Cluster is a set of WebLogic Server instances that work together to provide increased scalability and reliability. All the servers in the cluster belong to the same WebLogic domain. A cluster appears to clients as a single WebLogic Server instance.
Node Manager	Node Manager is a WebLogic Server utility that allows controlling the lifecycle of an Administration Server and Managed Server instances from a remote

	location. A Node Manager process is associated with a specific machine. It can be used to control server instances in any WebLogic Server domain, as long as the server instances reside on the same machine as the Node Manager process.
Target	WebLogic Server's administration entity that can host and act as a container for various J2EE subsystem services. For example, a Server, Cluster or Migratable Target
Server Target	A single WebLogic Server instance as a Target.
Cluster Target	One or more servers clustered together to act as one entity providing cluster services by replicating and sharing the common resources, like JNDI, etc.
Migratable Target	A virtual target that contains one or more candidate servers, with only one active server at any given time.
Candidate Servers	A list of servers that can act as standby for the user preferred server in the migration process.
User Preferred Server	The server instance that the user wants the hosted services to be running at the time of the migratable/system startup.
Source Server	The server instance where the services are migrated "from".
Destination Server	The server instance where the services are migrated "to".
SAN	Storage Area Network

OVERVIEW OF WEBLOGIC SERVER SERVICES

WebLogic Server offers both clustered as well as singleton services. ***“Clustered”*** services are generally available on all nodes in a cluster. For example, a web application deployed to the cluster will have an instance of the application on each server in the cluster. If the application is configured for session replication, the cluster replication service, which runs on all the nodes in the cluster, will replicate sessions to another node in the cluster.

“Singleton” services, are services that exist only on one node of the cluster at any given point of time, so as to offer specific qualities of service (QOS), but most importantly, to preserve data consistency. One such example is the JMS (Java Messaging Service) service, where the service must ensure that only one copy of the application message gets persisted to storage for every persistent send operation from the producer. But this high QOS has a cost. It is obvious that any problems or issues with the node that hosts the singleton service would render the service unusable or make it unavailable until the problem is corrected. This is a concern in an enterprise application environment.

Before we delve into detail about how this concern is addressed in WebLogic Server, we would like to provide a brief summary of various singleton services that are currently offered by WebLogic Server.

The current set of singleton services can be grouped into the following broad categories:

- Persistent Store Service
- JMS-related Services
- JTA-related Services
- User-defined singleton services

Persistent Store Service

WebLogic Server provides a Persistent Store service that can be shared by many subsystems to store various persistent data. A particular persistent store and the subsystems that use it must all run on the same WebLogic Server instance. Every WebLogic Server comes with a “default” file-based persistent store. You can configure additional custom stores and associate them with one or more subsystem services. WebLogic Server supports both file based as well as database table based custom stores.

Custom stores are migratable, while the default store is migratable only implicitly as part of JTA service migration (the default store cannot be used by migratable services other than JTA). For more information on the store see [1].

JMS Related Services

JMS Service

The WebLogic JMS Service offers publish-subscribe and point-to-point messaging functionality in WebLogic Server. This service is managed through the configuration entity called “JMSServer” that acts as a container hosting JMS *physical* destinations (queues and topics) and is typically targeted onto a WLS Server instance. To support the persistent messaging and preserve the data consistency, all

the persistent states (both messages and internal states such as transaction, acknowledgement, and so forth) of a given JMS Server are stored in only one place in the cluster, which is an associated persistent store hosted on the same WebLogic Server instance. For more information on JMS see [2].

SAF Service

The Store-And-Forward (SAF) service was introduced as another messaging service in WebLogic Server 9.0. As the name implies, the SAF service stores messages in a local cluster or domain and forwards them to their final destinations in a remote cluster or domain when the destinations are available. This service is exposed and managed through the configuration entity named “SAF Agent,” which is similar to JMS Server but acts as the container hosting the SAF *imported* destinations. The difference between the JMS Server and the SAF Agent in terms of the targeting is that the SAF Agent can be targeted to a cluster or a list of WebLogic Server instances as opposed to a single WebLogic Server instance for the JMS Server.

As with JMS Servers, SAF Agents also uses an associated persistent store for storing persistent state information and hence face similar problems with unhealthy server and store issues leading to service downtime.

Note that, the SAF agents can be configured to support Web-Service Reliable Messaging (WS-RM). In these scenarios, the system will only choose the non-migratable SAF Agents for hosting the imported destinations to use with WS-RM. For more information on the SAF service, see [3].

Path Service

The Path Service is an optional persistent map singleton service that is used to store the mapping of a named sequence of messages (a JMS Message ‘Unit-of-Order’) to a particular JMS Server or SAF Agent in a cluster. It provides a way to enforce ordering by pinning messages with same sequence name to the same physical location. This service is exposed and managed through the configuration entity named “Path Service”. There can only be one Path Service entity configured per Cluster, and again similar to the JMS Server and SAF Agent, the Path Service uses an associated persistent store to persist its data. For more information, see [4].

Message Driven Beans (MDB)

Message driven beans when configured to listen on a destination which is hosted by a migratable JMS server, would migrate along with the destination whenever the hosting JMS Server gets migrated. For more information, see [5].

JTA Related Services

The JTA Transaction Recovery Service (TRS) is designed to gracefully handle transaction recovery after a crash. Automatic migration of the Transaction Recovery Service leverages the Health Monitoring subsystem to monitor the health of the service hosted by a migratable target. When the primary server fails, the

migratable service framework automatically migrates the Transaction Recovery Service to a backup server. The migration framework selects a backup server from the configured candidate servers. If a backup server fails before completing the migration actions, then you must attempt to manually re-migrate the Transaction Recovery Service to another backup server.

For manual or automatic service migration, you must configure the default persistent store so that it stores records in a shared storage system that is accessible by any potential machine to which a failed migratable server might be migrated. In other words, you must restrict the potential servers to which you can migrate the Transaction Recovery Service to only those that have access to the default store files of the original server.

For more information on TRS, see [6].

User defined Singleton Services

WebLogic Server also allows developers to create custom services that can act as Singleton services in the cluster. To act as a Singleton, the class must implement the “weblogic.cluster.singleton.SingletonService” interface.

The SingletonService interface contains the following methods:

- `public void activate()`

This method is called when the service is activated on any node in the cluster.

- `public void deactivate()`

This method is called when the service gets deactivated on any node in the cluster.

For more information on user defined singleton services, see [7]

MIGRATION PROCESS

As mentioned in the previous section, singleton services offer one of the highest qualities of service but are very susceptible to “single point of failure.” To address this issue, WebLogic server offers a solution called “*migration*.”

Migration in WebLogic Server is the process of moving a clustered WebLogic Server instance or a subsystem component running on a clustered instance elsewhere in the event of failure.

The process of moving the entire server instance from one physical machine to another upon failure is called *Whole Server Migration* (WSM). On the other hand, moving only the affected subsystem services from one server instance to another running server instance is called *Service Migration*.

The *whole server* migration process was introduced in WebLogic Server 9.0 release. In this process, when a migratable server becomes unavailable for any reason, for example, if it hangs, loses network connectivity, or its host machine fails—the

server instance is automatically migrated. Upon server instance failure, a migratable server is automatically restarted on the same machine if possible. If the migratable server cannot be restarted on the machine where it failed, it is migrated to another machine. In addition, an administrator can manually initiate migration of a server instance. Please see section 0 for more details.

Service Migration was added in WebLogic server 7.0. This technique provides the necessary infrastructure to *manually* migrate only the failed subsystem services from an unhealthy server instance to an available healthy server instance.

However, manual migration needs human intervention (to detect the server/service failure and initiate the migration), which increases overall administration and ownership costs for WLS customers, as well as potentially leads to unpredictable service unavailability.

In WebLogic Server 10.3, the entire process of service failure detection and recovery is fully automated with an improved migration framework. The *Automatic Service Migration (ASM)* framework now proactively monitors the health of the Singleton services and automatically migrates failing services to a healthy and available server instance in a cluster, thus greatly reducing the time taken to perform the migration and increasing the overall availability of these services.

In rest of this paper, we talk about various infrastructure, framework, utilities, and tools that are in place to make the singleton services highly available in a clustered environment.

SERVICE MIGRATION FRAMEWORK

The service migration framework includes migratable targets, health monitoring service, leasing service, and node manager processes.

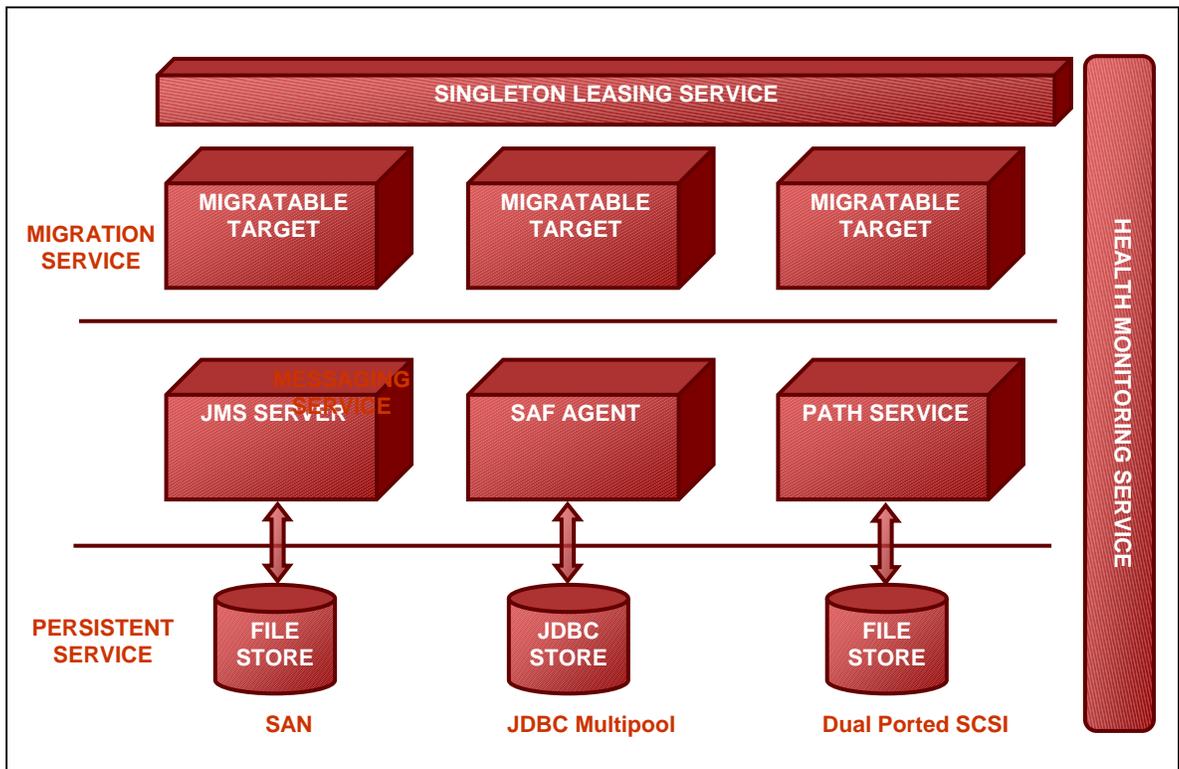


Figure 1. High Level Architecture of JMS ASM

Migratable Target

A Migratable Target is a logical target that can host many migratable services. It serves as a grouping of services that you want to run together and migrate together. A migratable target in turn is associated with the cluster. A Migratable Target is hosted at any point in time by only one node in the cluster called the *current hosting server*. All of the migratable services targeted to the Migratable Target run on the cluster node hosting the Migratable Target. When the Migratable Target is migrated, all services hosted by that target are also migrated.

Subsystem Health Monitoring Services

The Server subsystem health monitoring service detects various adverse conditions which would affect a server's normal operations or performance (such as low memory, and IO errors that prevent data persistence), and then takes appropriate action to reduce service disruptions.

Leasing Services

Leasing services grant leases to server instances in a cluster to host migratable targets. These leases guarantee exclusive ownership of the migratable target. A migratable service is hosted on the server which holds the migratable target's lease. Each server renews their leases periodically so as to continue to host the migratable

services. If a server crashes or is unable to renew its lease due to some other failure like a network partition, then the failed server's migratable services are migrated to a healthy server which acquires the lease.

In the leasing services framework, one managed server instance in a cluster acts as "Singleton Master", which is responsible for monitoring the health of all the Singleton and Migratable Services in that cluster, for migrating these services to a different WebLogic server instance within the cluster in case of failures, and for ensuring that a migratable target set of migratable services never runs on more than one server within the cluster.

There are two options available to a WebLogic Cluster for handling leasing information.

- Database Leasing

Leasing information is stored in a database. The database however must be highly available. Also each cluster node must be able to connect to the database in order to access leasing information. If the database is not available then automatic service migration framework will not work and all the migratable targets and singleton services will be deactivated in the cluster.

- Consensus

Leasing information is stored in-memory of the cluster nodes. It does not require a highly available database, but it does require all cluster servers be started by a Node Manager process. The Node Manager process must always be running for consensus leasing to work properly. The node manager is explained in the next section.

For more information on leasing services, see [8]

Node Manager

The Service Migration Framework depends upon the Node Manager to execute optional pre and post migration scripts, which are explained later in this paper. Also consensus leasing requires that servers be started and monitored by the Node Manager [9].

CONFIGURING THE SERVICE MIGRATION FRAMEWORK

Migration Basis (Leasing options) – on a Cluster

The Migration Basis determines the leasing service type – database or consensus – used by the migration service framework. See the previous section for guidance. The setting applies to the entire cluster.

A Migratable Target provides the following configurable properties Note that the JTA transaction recovery service does not have an explicit MT.

User Preferred Server (UPS) – on MigratableTarget

By default, every WebLogic Server instance comes with the migration counterpart which is named WLS Server instance name (migratable).

Every Migratable Target requires that the administrator select one node in the cluster as its User Preferred Server (UPS). The migration service framework always tries to start the service first on its UPS, if it is available.

Candidate Servers – on MigratableTarget

Candidate Servers define the set of servers where the Migratable Target could be migrated. It is a subset of the nodes in the cluster. If no candidate servers are defined for a Migratable Target, then all the nodes in the cluster become candidate servers and the Migratable Target could be migrated to any node in the cluster.

Migration Policies

A Migratable Target's Migration Policy controls whether migration automatically occurs in the event of a health failure or administrative shutdown. There are two automatic migration policies, "Exactly-Once" and "Failure-Recovery", as well as a "Manual" policy for disabling automatic migration. The default policy is "Manual".

Exactly-Once

The Exactly Once policy indicates that the service must always be active if at least one server in the candidate list is running. The Service is first tried to be started on its User-Preferred Server (UPS). But if the UPS is unavailable, then the service will be activated on some other candidate server in the cluster. If the server hosting the service fails or is shut down (either gracefully or forcibly), then the service will be migrated to another candidate server.

Failure-Recovery

The Failure-Recovery policy indicates that the service will only start if its User-Preferred Server (UPS) is started normally. If an administrator manually shuts down the UPS, either gracefully or forcibly, then a *failure-recovery* service will not migrate. However, if the UPS fails due to an internal error, that is detected by health monitoring service or if it crashes then the service will automatically migrate to another candidate server.

This policy is useful when the migratable service needs to be started on a particular server in the cluster and it shouldn't be migrated if the server is shutdown. But if the server crashes, then the service should be migrated.

Manual

The Manual policy indicates that the service will not automatically migrate when the User-Preferred Server (UPS) for the service fails or is shutdown (either gracefully or forcibly). The administrator needs to manually migrate the migratable target from a failed node to a healthy node.

Pre-Activation/Post-Deactivation Migration Scripts

These optional scripts are executed by the migration framework just after the migratable services are deactivated on the original machine and just before migratable services are activated on the target machine. The migration framework asks the original and target WebLogic Servers' Node Managers to execute these scripts. These scripts provides a mechanism to do any pre or post migration clean up tasks like mounting and un-mounting of shared partitions/file systems that needs to be available to the migratable services

Post-deactivation script failure could be marked as fatal to block migration of migratable services if the script fails. In such a case, the administrator needs to manually migrate the migratable target after taking corrective actions for the script failure.

Restart-In Place

Migratable targets provide a *"restart-in-place"* option that attempts to deactivate and reactivate a failed service on the same cluster node instead of migrating it to another cluster node.

The migration framework only attempts to restart a service if the services' Migratable Target is unhealthy but the server's health is satisfactory (i.e., in a RUNNING state). If the server is not healthy, then the framework immediately proceeds to the migration stage, skipping all in-place restarts. The number of restart attempts and the time interval between attempts is configurable. If the Migratable Target fails to restart after the designated number of restart attempts, the framework will migrate the target to another server.

For more information for configuring Migratable Targets, see [10]

MIGRATION CONFIGURATION RULES AND BEST PRACTICES

When using the Automatic Service Migration Feature, you must configure the "Migration Basis" for the cluster, enable automatic migration, and optionally specify whether you will be using any pre/post migration scripts.

When a messaging subsystem service needs to be migrated then it cannot use the "default" store for its persistence and it must define a custom persistent store that is also targeted to the same Migratable Target as the service provider entity.

A SAFAgent, a JMSServer, and a custom store can all share a migratable target. To preserve SAF message consistency, WebLogic Server prevents you from retargeting an existing SAF agent to a migratable target. Instead, you must delete the existing SAF agent and configure a new one with the same values and target it to a migratable target to preserve the message ordering.

Targeting Rules

JMS Servers

When not using migratable targets, a JMS server can be targeted directly to a specific cluster member and may use either the host server's default file store or a custom store. However, when targeted to a migratable target, a JMS server must use a custom persistent store, and must be targeted to the same migratable target as the custom store. A JMS server, SAF agent, and custom store can share a migratable target.

SAF Agents

When not using migratable targets, a SAF agent can be targeted to an entire cluster or a list of multiple servers in a cluster, with the requirement that the SAF agent and each server in the cluster must use the default persistent store. However, when targeted to a migratable target, a SAF agent can only have one target. It must also use a custom persistent store, and, like a JMS server, must be targeted to the same migratable target used by the custom store. A SAF agent, JMS server, and custom store can share a migratable target.

In addition, consider the following topics when targeting SAF agents to migratable targets.

Re-targeting SAF Agents to Migratable Targets

To preserve SAF message consistency, WebLogic Server prevents you from re-targeting an existing SAF agent to a migratable target. Instead, you must delete the existing SAF agent and configure a new one with the same values and target it to a migratable target.

Targeting Migratable SAF Agents for Increased Message Throughput

When not using migratable targets, a SAF agent can be targeted to an entire cluster or multiple servers in a cluster for increased message throughput. However, When a SAF agent is targeted to a migratable target; it cannot be targeted to any other servers in the cluster, including an entire cluster. Therefore, if you want to increase throughput by importing a JMS destination to multiple SAF agents on separate servers in a cluster, then you should create migratable targets for each server in the cluster, and then create separate SAF agents that are targeted individually to each migratable target.

Targeting SAF Agents for Consistent Quality-of-Service

A WebLogic administrator has the freedom to configure and deploy multiple SAF agents in the same cluster or on the same server. As such, there could be situations where the same server has both migratable SAF agents and non-migratable ones. For such cases, the behavior of a JMS client application may vary depending on which SAF agent handles the messages.

For example, an imported destination can be deployed to multiple SAF agents, and messages sent to the imported destination will be load-balanced among all SAF agents. If the list of the SAF agents contains non-migratable agents, the JMS client application may have a limited sense of HA (high availability). Therefore, a recommended best practice is to deploy an imported destination to one or more SAF agents that provide the same level of HA functionality. In other words, to get consistent forwarding quality and behavior, you should target the imported destination to a set of SAF agents that are all targeted to migratable targets or are all targeted to non-migratable targets.

Path Service

When not using migratable targets, a path service is targeted to single member of a cluster, and can use either the default file or a custom store. However, when targeted to a migratable target, a path service cannot use the default store, so a custom store must be configured and targeted to the same migratable target. As an additional best practice, the path service and its custom store should be the only users of that migratable target, whereas, a JMS server, SAF agent, and custom store can share a migratable target.

Persistent Stores

As mentioned previously, all JMS-related services require a custom persistent store that is also targeted to the same migratable targets as the JMS services.

Migratable custom file stores can be configured on a shared disk that is available to the migratable target servers in the cluster or can be migrated to a backup server target by using pre/post-migration scripts. For highest reliability, use a shared storage solution that is itself highly available—for example, a storage area network (SAN) or a dual-ported disk.

JTA Transaction Recovery Service

For JTA, migratable target configuration is not necessary for automatic or manual migration because a migratable target is automatically defined for JTA at the server level. The default migration policy for JTA is manual, but when configured for automatic migration, the JTA policy is internally set to failure-recovery. This means that Transaction Recovery Service will only start if its user-preferred server (UPS) is started. If an administrator shuts down the UPS either gracefully or forcefully, this service will not be migrated anywhere. However, if the UPS shuts down due to an internal error, then this service will be migrated to another candidate server.

It is recommended that you attempt to restart a crashed server and allow the transaction Recovery Service to handle incomplete transactions, rather than migrate the Transaction Recovery Service to another server. However, if you anticipate that the server will be unavailable for an unacceptable length of time, you can migrate the Transaction Recovery Service to another service so that the backup server can complete transaction work for the failed server.

SUBSYSTEM SERVICE MIGRATION VS WHOLE SERVER MIGRATION

As described in “Migration Process,” in addition to automatic subsystem service migration, WebLogic Server supports the migration of the entire WebLogic Server instance, which is called as “whole server migration” or WSM. Whole server migration enables the transparent failover of the entire server instance from one physical machine to another thus providing the option for “homogeneous” service migration, or migrating everything together. As this needs to boot an entire WLS instance on new hardware, it usually takes a longer time to make the services available. Whereas, the automatic subsystem service migration (ASM) is primarily used for migrating “pinned” services such as JMS and JTA. Unlike, whole server migration, the automatic service migration supports both failover and failure recovery of the subsystem services and offers faster failover or recovery time compare to whole server migration.

Users can choose to use either WSM or ASM or both. The decision to use which type of migration to use depends on a number factors, such as whether the user wants to migrate all the applications along with the JMS/JTA subsystem services or not. In general, the whole server migration is preferred for basic use due to its relative simplicity, but automatic service migration becomes attractive when faster fail-over times and advanced control over the service migration are desirable. When both are configured, the server will first try to first migrate the subsystem services and if that fails, then the whole server migration would take place. For more info on the whole server migration, please see [[11]].

TROUBLESHOOTING

The following WebLogic debug flags would help diagnose any problems that arise in this feature area. The debug messages are logged in the appropriate server log files.

Subsystem Area	Debug flag
Information about the Singleton Monitor's actions	-Dweblogic.debug.DebugSingletonServices=true
If using Consensus Leasing	-Dweblogic.debug.DebugConsensusLeasing=true
To trace the JMS Server deployment/undeployment	-Dweblogic.debug.DebugJMSBackEnd=true
To trace the SAF Agent deployment/undeployment	-Dweblogic.debug.DebugJMSAF=true
To trace the JMS Module deployment/undeployment	-Dweblogic.debug.DebugJMSModule=true
To trace the Persistent Store deployment/undeployment	-Dweblogic.debug.DebugStoreIOPhysical=true -Dweblogic.debug.DebugStoreIOLogical=true -Dweblogic.debug.DebugStoreIOLogicalBoot=true

CONCLUSION

WebLogic Server 10.3 delivers the fully automated service migration feature that eliminates the need for human intervention when making the singleton services highly available. Customers can now take advantage of this feature to reduce the total cost (TCO) associated with server administration. By making their enterprise applications highly available they can increase their overall return on investment (ROI) on the WebLogic Server platform.

APPENDIX A: REFERENCES AND RELATED DOCUMENTS

- [1] [WebLogic Persistent Store](#)
- [2] [WebLogic JMS Service](#)
- [3] [WebLogic Store and Forward Service](#)
- [4] [WebLogic Path Service](#)
- [5] [MDB Migration](#)
- [6] [WebLogic Transaction Recovery Service](#)
- [7] [Automatic Migration of User Defined Services](#)
- [8] [Leasing for Automatic Migration Services](#)
- [9] [Node Manager Administrator's Guide](#)
- [10] [Configuring Migratable Targets](#)
- [11] [Whole Server Migration](#)



Automatic Service Migration in WebLogic Server
July 2008

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Copyright © 2008, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

Other names may be trademarks of their respective owners.